Numerical simulations of lattice models of many-body localization

Fabien Alet[1]

1 Laboratoire de Physique Théorique, CNRS and Université de Toulouse, France
* fabien.alet@cnrs.fr

July 24, 2023

Abstract

These notes provide a gentle introduction to numerical simulations for lattice models of many-body localization. They definitively do not cover details of all possible simulations techniques, and several of them are simply briefly mentioned. We nevertheless hope that they will be useful to beginners not only to have an overview of the field (including the essential references) but also to make them no longer be afraid of starting their own simulations. A repository associated to these lecture notes containing starting codes is precisely meant for this: https://github.com/fabienalet/Cargese-MBL. This is a β-version of these notes, only meant for participants in the school. Please do not diffuse this document for the moment as it is far from being final (there is probably a lot of errors and typos left). We encourage all comments/criticisms/ideas (even minor or typographical) to improve them. Contributions to the repository are also welcome. These notes were originally written for a school in Les Houches in 2019, slightly refreshed for this Cargese school and will be further reshaped and augmented in late 2023.

Contents

1 Introduction

The many-body localization (MBL) problem has attracted a lot of interest over the last 10 years. We do not aim to provide an introduction to MBL, and refer to the lectures by D. Abanin in this school, as well as to excellent reviews [1–6]. The discussion will be focused here on numerical simulations of MBL lattice models. To the best of our knowledge, there is no review on this specific topics (even though Ref. [2] and especially Ref. [4] have some informative sections). The goal of these notes is thus to gather some specific knowledge on this topic. This introductory section provides the general context and motivations for such simulations, and sets the roadmap for the other sections.

1.1 Context of numerical approaches to the many-body localization problem

1.1.1 Specificities of the MBL problem

Numerical simulations have been instrumental in our understanding of MBL as many physical effects have been first found in numerics. However, the MBL problem possesses specific features making its numerical treatments different from other cond-mat or AMO problems. Before discussing these peculiarities, let us first make a wish-list. Ideally, numerical simulations of lattice models of MBL should:

- access eigenstates *and* dynamics.
- address all regimes in a typical phase diagram: the MBL phase, the ergodic/ETH phase (including possible subdiffusive / 'bad metal' / non-ergodic regimes), the ETH/MBL transition (including the possible coexistence of MBL and ETH states in the same spectrum).
- make no approximation.
- work in any dimension, for any model.
- eventually allow to study coupling to a bath, study (periodically) driven systems.

- scale well, or at least reach large enough system sizes to be able to comment on the thermodynamic limit and/or experiments

Obviously, no method or family of methods offers this (the second and last points are probably the most tricky), but it is good to have these higher goals in mind to identify the benefits/caveats of each numerical technique. Let us now try to see which specific issues are associated to simulations of MBL:

- MBL is a 'high-energy' problem: contrary to most condensed matter studies, we are not specifically interested in low-energy physics. Obtaining eigenstates in the middle of the spectrum is challenging for traditional cond-mat numerical methods.

- Absence of thermalization in the MBL phase. Thus methods should *not* couple the system to a heat-bath. We need to address eigenstates *individually*.

- Slow dynamics in the MBL / critical regime: we need to probe long time-scales.

- The Hilbert space size $|\mathcal{H}|$ grows exponentially: this is standard for the quantum many-body problem (but is different from Anderson localization). MBL specific issues are: (i) the sizes $|\mathcal{H}|$ are in practice smaller than on other comparable cond-mat problems, due to other issues in this list, (ii) finite-size effects appear to be strong for the 1d MBL transition, on top of which there is not yet an accepted finite-size scaling theory to understand them.

- Disorder issues: (i) Disorder reduces symmetries of the problem (we can no longer take advantage of them to reduce complexity / increase system size). Furthermore, some high symmetries are not compatible with MBL [7]. (ii) Disorder averaging is required. While this is standard in other disordered cond-mat problems (*e.g.* Anderson localization), the required numerical effort cannot at the moment be alleviated (see a possible way out in the discussion session). Furthermore, Griffiths effects (which require a lot of averaging to be probed), are very important in 1d where most of the MBL simulations are performed.

Standard numerical methods in condensed matter do not work for one (or several) of these reasons. A non-exhaustive list includes: (i) methods that target ground-state and low-energies properties: iterative linear algebra methods* (*e.g.* Lanczos algorithm), Density Matrix Renormalization Group* (DMRG), projector quantum Monte Carlo (QMC) techniques$^{*(?)}$, Variational Monte Carlo, $T = 0$ series expansion* etc, (ii) methods that need a thermal bath: finite-temperature (e.g. path-integral) QMC, high-temperature series expansions ... Methods marked with an asterix $^{(*)}$ can be and have been extended to address the MBL problem, as will be discussed in these notes.

On the other hand, some MBL features are to our advantage:

- MBL states have low entanglement: their area law entanglement is very favorable to Matrix Product States (MPS) methods (see Section 3). A tentative list is: DMRG for excited states under all its nicknames (DMRG-X, SIMPS, ES-DMRG etc), the unitary diagonalizing the Hamiltonian described by a single Matrix-Product Operator (MPO), Time-Evolving Block Decimation, MPO methods for open systems etc.

- Analytical understanding through the existence of local integral of motions (LIOMs) in the MBL phase (see lecture by D. Abanin): several methods taking explicit advantage of the LIOM construction/existence (including unitary flow techniques, quantum Monte Carlo based on LIOMs etc.) are very briefly reviewed in Section 4.

- Decades of experience of simulations of the related Anderson localization models as well as in exact simulations of quantum many-body physics in general (the latter being the focus of Section 2).

1.1.2 Lattice model(s)

Which lattice models are available to apprehend MBL physics? Many of the MBL simulations crystallized on the random-field spin 1/2 XXZ spin chain:

$$H = \sum_{i=1}^{N} S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + \Delta S_i^z S_{i+1}^z - h_i S_i^z \tag{1}$$

where $S_i^\alpha = 1/2\sigma_i^\alpha$ are half Pauli matrices, and the random fields h_i are often taken uniformly from a box distribution $h_i \in [-h, h]$. The spin anisotropy Δ quantifies the interactions in this model (the model can be mapped to a model of 1d spinless fermions which interact precisely with a strength Δ with their neighbors). We do not review the physics of this model here, but simply note that (i) this model has a U(1) symmetry due to total magnetization $S^z = \sum_i S_i^z$ conservation, (ii) it is reasonably well accepted that this model exhibits both a ergodic/ETH phase for low values of h and a MBL phase for large h.

There are many more lattice models and/or forms of disorder that have been explored, some of which are in relation to experiments: quantum Ising models, spinless fermionic models, fermionic or bosonic Hubbard models, more recently constrained models, Aubry-André / quasiperiodic potentials, binary disorder, random interactions etc (see a brief review in Ref. [2]).

For several reasons, it might be useful to get rid not only of the magnetization conservation (this can be done easily by adding a different energy scale for the $S^y S^y$ term), but also of energy conservation. Then one considers (periodically) driven Floquet systems which do not conserve energy. A typical example are "bang-bang" models where two different evolutions U_1 (duration T_1) and U_2 (duration T_2) are performed consecutively to create a total evolution operator $U = U_1 U_2$ which is repeated periodically in time (period $T = T_1 + T_2$). We choose evolution terms that do not commute with each other, a typical example is $U_1 = \exp(-\mathrm{i} H_1 T_1)$ with $H_1 = \sum_i S_i^z S_{i+1}^z + h S_i^z$ with and $U_2 = \exp(-\mathrm{i} H_2 T_2)$ with $H_2 = g \sum_i S_i^x$. Typical favorable ingredients for simulations of such lattice models are the low dimensionality of the physical space (1d or 2d) and, if it exists, of the local Hilbert space (*e.g.* spin 1/2 models have 2 degrees of freedom per site). Of course, to really probe MBL specific features, one has to make sure that there is no mapping to an Anderson single-particle problem (*e.g.* avoid free-fermions point in the phase diagram, such as $\Delta = 0$ in the XXZ model above) in the lattice model under study.

1.2 Plan of the notes

We try in these notes to provide an introduction to most of the numerical methods that have been explored to address MBL physics. However, due to our lack of space and mostly lack of expertise on several of these methods on the author' side, we do not provide a detailed account of all of them.

Section 2 is an exception to this, as it comprehensively describes exact methods (methods that use a machine-precision representation of wave-functions) efficient for MBL. It presents the construction of the Hilbert space, the shift-invert technique, the Krylov time-expansion method and the computation of observables. The next two sections do not go into so much details and rather provide a bird's eye-view on other numerical methods that are valuable for the MBL problem. Section 3 is dedicated to methods based on MPS and MPO. Finally, Section 4 is essentially a reference list (with a few comments) on other numerical methods for MBL. It also contains a conclusion on open challenges.

A repository (https://github.com/fabienalet/Cargese-MBL) associated to these lecture notes contains codes in C++ and Python notebooks which implement most (some) of the methods presented in Section 2 (3). The notebooks are particularly useful as playgrounds to test and check various models, observables and ideas.

2 Methods based on exact representation of many-body wave-functions

Methods which explicitly work on wave-functions on the full Hilbert space $\mathcal{H}$ of a many-body problem are very useful in the MBL context, even if its size $|\mathcal{H}|$ grows exponentially. The main reason for this usefulness is the failure of many traditional condensed matter numerical methods (see Section 1), as well as the fact that other methods (such as reviewed in Sections 3 and 4) which can sometimes handle larger system sizes work in specific physical regimes (*e.g.* only in a full MBL phase). The exact methods described below provide reference data for any parameter range in a MBL phase diagram and for any lattice model. Besides their physical interest, such data also are particularly useful to benchmark other numerical techniques.

We begin with a discussion on how to efficiently represent a basis for $\mathcal{H}$, including non-trivial constructions for models with conserved quantities or constraints. We then briefly discuss available strategies to implement the Hamiltonian or the evolution operator. After a quick reminder on the Lanczos algorithm (useful to get extremal eigenpairs of a sparse operator), we discuss spectral transform methods (in particular the shift-invert technique) which are optimal to get eigenpairs in the middle of a many-body spectrum. We then make a small pause and provide a practical guide on on how to compute specific observables as well as on usual strategies and pitfalls of MBL simulations. The next section then moves on to discuss how to perform simulations of dynamics in a lattice model using Krylov expansion methods, followed by a discussion for specific dynamical correlation functions. This section is concluded by a brief discussion on simulations of quantum circuits.

2.1 Efficient representation of the basis

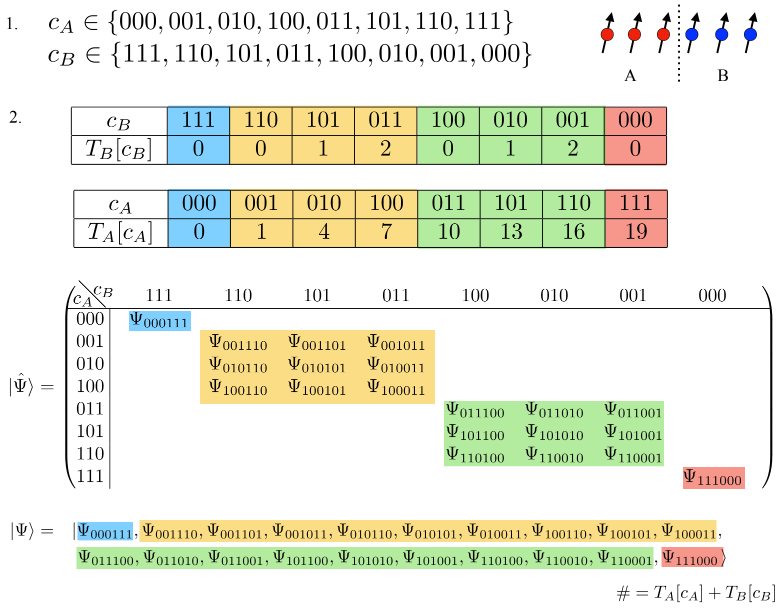
As the Hilbert space size $|\mathcal{H}|$ grows exponentially with the system size, one should be careful to ensure an optimal use of CPU time and memory in generating *and* working with its basis. The important tasks are (i) to generate efficiently and provide an index for all configurations with this property, (ii) to easily recover this index given a configuration (this is needed for computing matrix elements).

Consider a spin system with N spin $1/2$ degrees of freedom. Start without specifying any symmetry for the moment, meaning that there are 2^N configurations in the total basis. Since there is a one to one mapping (bit representation) between these configurations and all integers $\in [0, 2^N - 1]$, one can easily navigate from the bit representation to the the integer number indexing the configurations: tasks (i) and (ii) are trivially solved.

Now consider only configurations with as many up than down spins in the standard $\{S^z\}$ basis (this happens for instance, when working in the zero magnetization sector $S^z = 0$ of the XXZ model Eq. 1). There are now $\binom{N}{N/2}$ configurations. The issue (i) is not very problematic in general (a simple loop over all 2^N states which selects only the valid configurations works well on moderate to fairly large systems). It may become more cumbersome for $N \simeq 40$ or for other (large) magnetizations sectors. In those situations, one could use some dedicated algorithm that starts with a valid configuration and moves bits one by one in a Gray-code like manner (in any case, we discuss a simpler way later). The issue (ii) can rapidly become more serious. A simple large array T, containing the mapping $T[c] = i$ where c is a (bit) representation of configurations and i its index, is in principle sufficient. However, the memory requirement for this table (of size $|\mathcal{H}|$) can become important (*e.g.* ≥ 16 Gb for $N \geq 32$). This becomes an issue for parallel distributed memory methods (*e.g.* using MPI), which for efficiency would need to store a copy of this table on each process.

2.1.1 Lin Tables

There is a simple trick, known as Lin tables [8], which allows to overcome both issues (i) and (ii) at the same time, and can as a bonus ease computation of reduced density matrices and entanglement entropy (see Sec. 2.4.3). The idea is to split the system in two subsystems A and B (which we take of the same size for simplicity, assuming N to be even, even though this is not needed). One then exactly enumerates the basis on each subsystem (this is fast) in a given order, and forms the two Lin tables T_A and T_B in the following way: starting with a configuration on the A side (say c_A), assign $T_A[c_A] = n$ with n the number of configurations of the total system found so far ($n = 0$ at the beginning), list all (say k) configurations of B (say c_B^i with $i = 0 \ldots k-1$) compatible with c_A (*i.e.* which would result in valid $S^z = 0$ configuration) and assign $T_B[c_B^i] = i$, increase n by k, and repeat with the next c_A. An example of Lin tables for $N = 6$ is provided in Fig. 1. You can check that the table $T_B[c_B]$ is uniquely defined. Solving issue (ii) for any configuration c is done by splitting it in two parts c_A and c_B: the index of this configuration is $T_A[c_A] + T_B[c_B]$. The memory requirement is very light as the two tables T_A and T_B are of size $2^{N/2}$. They can be stored at a low cost on every parallel process.



1. $c_A \in \{000, 001, 010, 100, 011, 101, 110, 111\}$
   $c_B \in \{111, 110, 101, 011, 100, 010, 001, 000\}$

2.

c_B	111	110	101	011	100	010	001	000
$T_B[c_B]$	0	0	1	2	0	1	2	0

c_A	000	001	010	100	011	101	110	111
$T_A[c_A]$	0	1	4	7	10	13	16	19

$$|\hat{\Psi}\rangle = \left(\begin{array}{c|cccccccc} c_A \backslash c_B & 111 & 110 & 101 & 011 & 100 & 010 & 001 & 000 \\ \hline 000 & \Psi_{000111} & & & & & & & \\ 001 & & \Psi_{001110} & \Psi_{001101} & \Psi_{001011} & & & & \\ 010 & & \Psi_{010110} & \Psi_{010101} & \Psi_{010011} & & & & \\ 100 & & \Psi_{100110} & \Psi_{100101} & \Psi_{100011} & & & & \\ 011 & & & & & \Psi_{011100} & \Psi_{011010} & \Psi_{011001} & \\ 101 & & & & & \Psi_{101100} & \Psi_{101010} & \Psi_{101001} & \\ 110 & & & & & \Psi_{110100} & \Psi_{110010} & \Psi_{110001} & \\ 111 & & & & & & & & \Psi_{111000} \end{array}\right)$$

$$|\Psi\rangle = |\Psi_{000111}, \Psi_{001110}, \Psi_{001101}, \Psi_{001011}, \Psi_{010110}, \Psi_{010101}, \Psi_{010011}, \Psi_{100110}, \Psi_{100101}, \Psi_{100011}, \Psi_{011100}, \Psi_{011010}, \Psi_{011001}, \Psi_{101100}, \Psi_{101010}, \Psi_{101001}, \Psi_{110100}, \Psi_{110010}, \Psi_{110001}, \Psi_{111000}\rangle$$

$\# = T_A[c_A] + T_B[c_B]$

Figure 1: Illustration of the Lin method for enumerating configurations accessible to a spin system (here the 20 configurations with $S^z = 0$ for $N = 6$ spins 1/2). After cutting the system in two and brute-force enumerating the configurations in the subsystems (1), the Lin tables are constructed (2, see text). The basis enumerated in this way has a block structure (see colors in the matrix or vector forms of a wave-function $|\Psi\rangle$), where in this case the blocks can be labelled by (S_A^z, S_B^z). The total configuration index (in the order of the full basis) is easily retrieved as $T_A[c_A]+T_B[c_B]$ when a configuration is split into its two components c_A and c_B.

You will remark that a given configuration of A is not necessarily compatible with all configurations of B. This is due to the fact that the total magnetization is constrained to be $S^z = 0$, thus we must have $S^z_A + S^z_B = 0$. This results in a block structure (where blocks of configurations can be labelled by $b = S^z_A$ – strictly speaking the pair (S^z_A, S^z_B) – in this example) such that configurations of A in block b are compatible only with those of a single block in B, and vice-versa. This *diagonal* block structure immediately entails a block structure for the reduced density matrix $\rho_A = \mathrm{Tr}_B \left|\Psi\right\rangle \left\langle\Psi\right| = \hat{\Psi}\hat{\Psi}^\dagger$, and is useful in many respects, notably to reduce the complexity of observables and entanglement computations. It is quite useful to order the configurations of A and B using this block structure such that issue (i) is solved by doing a triple loop over all block indices, all configurations of A in this block and all configurations of B in the corresponding block: this is actually a very efficient way of generating all required configurations. Lin tables can be used for larger spins, and are compatible with translation invariance (although they become a bit more complicated). As the systems considered for MBL usually lack translation invariance, we refrain from elaborating on this and refer interested readers to Ref. [8].

2.1.2 Basis construction for constrained systems

The half-system trick works fine in the spin chain example as the Hilbert space has a tensor product structure: $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$ (and $\mathcal{H}_A, \mathcal{H}_B$ are themselves tensor products of local Hilbert spaces). What happens for constrained models where this property is no longer fulfilled? The same trick can still apply, but with possible differences in the properties of the blocks. Let us take two illustrative examples.

Systems with 'diagonal constraints' : quantum dimer models — We first consider quantum dimer models, where the Hilbert space is defined by dimers (hardcore objects living on a lattice bond) coverings of a given lattice (see Fig. 2). The degrees of freedom now live on the lattice bonds, and the constraint is that every lattice site is touched by one and only one dimer. We assume no further symmetry for the moment. Enumerating all configurations that fulfill this constraint (task (i)) is not easy in general: even though there are published algorithms to generate all these dimer coverings in the computer science community (see *e.g.* Ref. [9] for bipartite lattices), they do not turn efficient in practice for the system sizes considered. We nevertheless assume for now that we have a basic covering heuristic algorithm, which can be slow, that can do this. We will speed up the process by separating the lattice bonds into two sets (red and blue in Fig. 2). In each set, there are sites which only belong to this set, and N_m sites which also belong to the other (ideally we want a bipartition such that N_m is minimal). The method works as follows. Enumerate all 2^{N_m} ways of randomly assigning these N_m sites to one set or the other: this will create 2^{N_m} pairs of mini-lattices A_i, B_i, for which we now search coverings using the basic covering algorithm (we find n_{A_i}, n_{B_i} coverings for these mini-lattices). In doing this, we have found (at most) 2^{N_m} blocks such that all configurations of A_i are compatible with those of B_i to form a covering of the full lattice. This is again a block-diagonal decomposition of the basis. The rest of the method is then exactly similar to the previous discussion on spin basis (Lin tables, triple loop over sectors and configurations), such that the total number of configurations is $\sum_i n_{A_i}.n_{B_i}$. The only difference is that there is now an exponential (with the surface of the partition between red and blue bonds) number of sectors, in contrast with the linear number of magnetization sectors for the spin chain. This of course slows down things, but the method stays manageable and fast up to large system sizes. In practice there are many blocks with no configurations either in A, B or both (such as for A_2 and B_2 in Fig. 2 which have an even number of sites), that are not used in the rest of the computation.

Let us mention that on some lattices, there are further conserved symmetries (geometrically conserved quantities known as winding numbers) for dimer coverings. Depending on the set bipartition choice, one of these conserved numbers can also be conserved on the mini-lattices A_i, B_i such that there exist mini-blocks inside the 2^{N_m} blocks, with each mini-block in

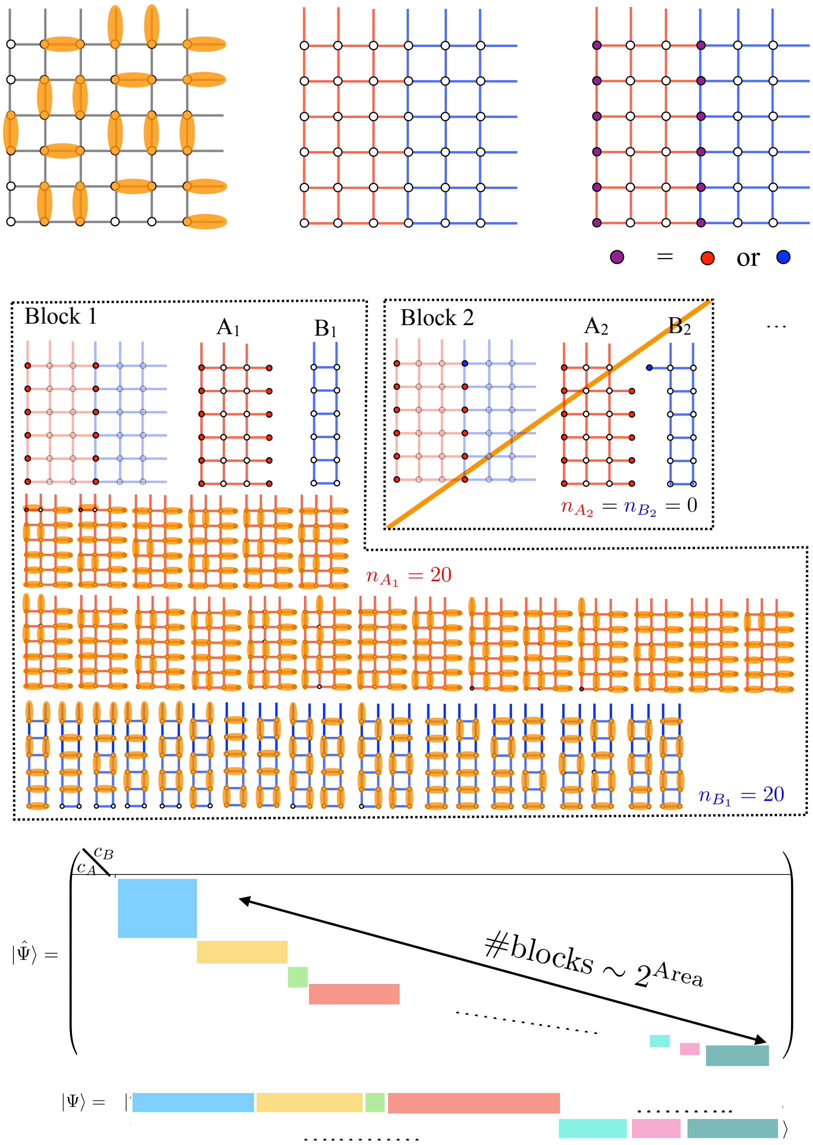


Figure 2: Enumeration of dimers coverings (such as given on the top left) of a square lattice (here with 6 × 6 sites and periodic boundary conditions). We first bipartition the lattice into two sets A and B of bonds and identify boundary sites (●) which touch both sets (top figures). The Lin blocks now correspond to all possible states of these boundary sites (● = ● or ●). For each block i, sub-lattices A_i, B_i are identified on which dimer coverings are enumerated (middle figure). For the first block, the same number $n_{A_1} = n_{B_1} = 20$ of coverings is obtained but this need not be the case in general. For the second block, no coverings is possible as A_2 and B_2 both contain an odd number of sites. All coverings of A_i are compatible with all of B_i to form a valid dimer configuration. This gives again a block-diagonal structure to the basis (bottom figure). The number of blocks grows exponentially with the area between A and B.

A_i having only one mini-block partner in B_i. This is again a diagonal constraint which can be implemented straightforwardly.

Off-diagonal constraints: PXP model — The PXP model [10] is a spin-1/2 chain model with a local constraint: the spin configurations with two consecutive up spins $\cancel{|\uparrow\uparrow\rangle}$ are not allowed in the Hilbert space. This leads to Hilbert space size scaling as $F(N+2)$ for a spin chain with N sites and open boundary conditions, with $F(N)$ the N-th Fibonnaci number (with $F(1) = F(2) = 1$). Constructing these configurations (task (i)) is easily performed with a recursive scheme, however it is useful to consider how the two-subsystems trick works for task (ii). For open boundary conditions, there are two relevant blocks of configurations for each sub-system: block A_0 (A_1) with configurations ending with a down spin $|\ldots\downarrow\rangle$ (with a up spin $|\ldots\uparrow\rangle$), and blocks B_0 (B_1) starting with a down (up) spin on the right subsystem $|\downarrow\ldots\rangle$ ($|\uparrow\ldots\rangle$). Obviously block A_0 is only compatible with block B_1 while A_1 is compatible with both B_0 and B_1. This *off-diagonal* block structure leads to a slightly different way of generating configuration, as the first loop now is over pairs of compatible blocks (instead of over the block label of one subsystem). This off-diagonal structure readily generalizes to periodic boundary conditions where now four blocks of configurations are present for each subsystem $(|\downarrow\ldots\downarrow\rangle, |\downarrow\ldots\uparrow\rangle, |\uparrow\ldots\downarrow\rangle, |\uparrow\ldots\uparrow\rangle)$.

2.2 Defining the Hamiltonian

Once the computational basis is constructed, we need to compute elements of the Hamiltonian (or of another evolution operator). Many-body Hamiltonians are usually sparse matrices, and we need to take advantage of this. In addition, one should of course use all symmetries (S^z symmetry in the random field XXZ chain Eq. 1), and in principle this has been done in the construction of the basis (for disordered systems, there is no translation or point-group symmetries to be used in general).

Unless when dealing with a very small system, one never saves the full Hamiltonian in memory, but rather only its non-zero elements. Most libraries offer such a sparse representation of matrices. Fast sparse matrix-vector multiplication are then possible. It may not always be possible to save (non-zeroes of) the Hamiltonian in memory on a single computer due to memory limitations, in which case one can use parallel resources to save the Hamiltonian (e.g. each node of a supercomputer gets some lines of the matrix). There are good available libraries which do that (*e.g.* PETSc) which are reasonably easy to use and usually installed on supercomputers.

An important alternative strategy is to not save the Hamiltonian, but rather recompute its elements when needed. For extremely large systems ($N \sim 50$ spins 1/2), this is the only way, and fortunately there are many matrix-free algorithms which are efficient (see Lanczos algorithm in Sec. 2.3.1). What is quite important here is to have an optimized routine which applies the Hamiltonian to a given vector (in other words, fast matrix-vector multiplication). This approach is recommended for (Floquet) evolution operators which are usually dense matrices (see discussion in Sec. 2.5.3), as well as for operators which are neither dense nor very sparse (e.g. models with long-range hopping terms or SYK-type Hamiltonians). Note that some algorithms (*e.g.* LU decomposition, see Sec. 2.3.2) are not compatible with matrix-free representations.

Floquet systems — In a driven system, energy is not conserved and one will more naturally deal with the (unitary) evolution operator $U(t)$ rather than with the (Hermitian) Hamiltonian operator that has been the focus of the discussion so far. Dealing with a unitary rather than an Hermitian operator calls for a different approach. When the drive is periodic, one can associate to the evolution operator over one drive period $U(T)$ the *Floquet Hamiltonian* defined as $U(T) = e^{-\mathrm{i}H_F T}$. The eigenvalues of U, which can be written as $\exp(i\varphi_n)$ (all of unit modulus) and its eigenstates contain precious information.

2.3 Obtaining exact eigenstates

2.3.1 Diagonalization routines

Full diagonalization — To obtain all eigenpairs of a Hamiltonian or a periodic Floquet evolution operator, one resorts to standard diagonalization routines (Lapack, Eigen) which typically use shared-memory parallelism (openMP). They scale as $|\mathcal{H}|^3$ ($|\mathcal{H}|^2$) for CPU time (memory). One can typically perform full diagonalization up to Hilbert space size $|\mathcal{H}| \simeq 30.000$ (larger can be obtained if only asking for eigenvalues); *i.e.* $L = 16$ for the XXZ chain Eq. 1 in the $S^z = 0$ sector. For even larger matrices, one can use Scalapack (perhaps up to size 200.000, probably quite slow!), a parallel implementation useful for distributed memory machines. For disordered systems, one also needs to average over many disorder realizations. For a given computational budget, a balance has therefore to be made, limiting in practice the larger $|\mathcal{H}|$ that can be reached.

Getting (a few) extremal eigenvalues — This is an old-problem in numerical linear algebra, and the most interesting algorithms are iterative. Among them, the Lanczos algorithm, a Krylov space projection technique (see Fig. 3), is often considered optimal for sparse Hermitian matrices. Many stable implementations can be found in several libraries, however it is very instructive to write a small code to learn it (see notebook in the repository).

For the MBL Hamiltonian problem, the eigenstates of interest are located at finite energy density and not at the extrema of the spectrum. The Lanczos algorithm thus does not appear directly useful – however it will become so after a spectral transform.

Diagonalization techniques for Floquet systems — Here, the driven nature of the system is helping us: because energy is not conserved, there is no notion of low or high energy states. In particular, any Floquet eigenstate is as good as any other at probing the thermal/ETH or MBL nature of the system. Applying the Lanczos algorithm to $U(T)$ itself yields poor convergence results, as its eigenvalues are all of the same modulus. First, one is much better off working with an Hermitian operator, such as $\tilde{H} = U(T) + U^\dagger(T)$. Second, a clever use of spectral transforms is often necessary in the Floquet case (see below). Let us remark that the power of iterative methods is conditioned on one's capability to efficiently compute the action of the operator on a given vector. The Floquet operators being generically *dense* in any local computational basis (contrary to Hamiltonians), these methods are *a priori* seriously impaired. However, when the Floquet operator can be written as a shallow quantum circuit or in specific convenient cases, this issue is alleviated, as discussed in Sec. 2.5.3.

2.3.2 Spectral transform techniques

One often desires eigenpairs close to some target energy λ which is assumed to be deep in the middle of the spectrum, such that iterative algorithms will fail (if the eigenstate is less than, say, 1000 states away from the spectrum extrema, iterative techniques can work). This problem which is central in MBL (as often one considers states at 'infinite temperature') is quite difficult: indeed the level spacing scales there as $\exp(-\alpha N)$. A good idea is to perform a *spectral transform* by considering a different operator G with the same eigenstates as H, but for which the eigenvalues close to λ for H are now extremal for G. Besides the fact that one can obtain exact eigenstates for larger systems as described below, another important aspect of spectral transforms is that they are energy-resolved: varying λ allows to probe physics at different energies, which is useful to discuss the existence of many-body mobility edges. The first operator that comes to mind is $G = (H-\lambda)^2$. It works, but fails for large matrices (typically $L > 18$ for the XXZ spin chain). The main reason is that the spectrum, which is already quite dense for H, will even get denser for $(H-\lambda)^2$ close to the desired eigenvalue. The next natural choice is $G = (H-\Lambda)^{-1}$: this is the shift-invert transform method that we discuss in detail below. Another slightly different possibility can be identified: filter transformations,

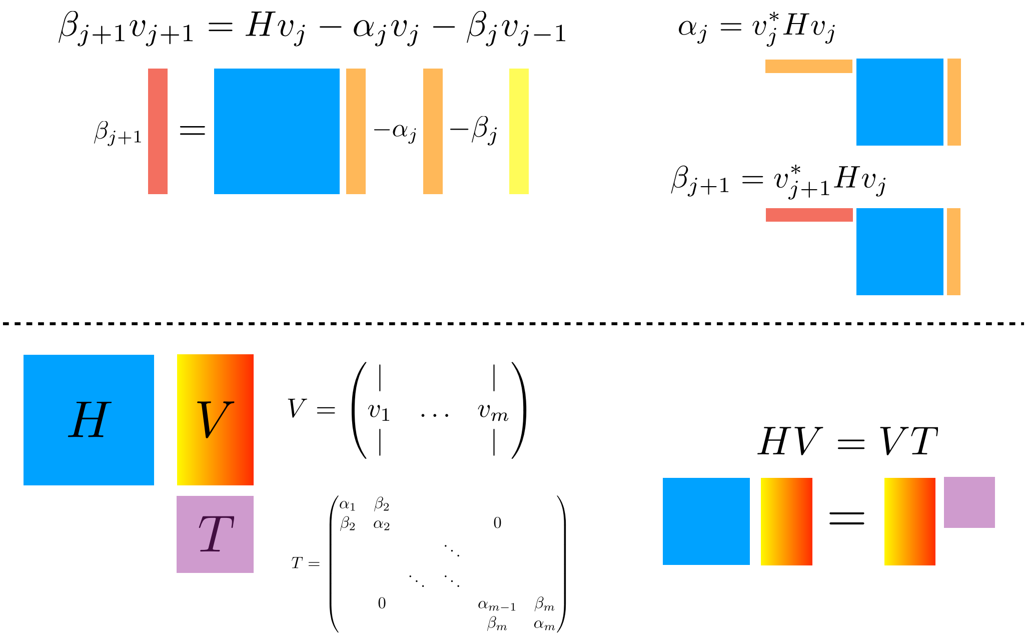

$\beta_{j+1}v_{j+1} = Hv_j - \alpha_j v_j - \beta_j v_{j-1}$
β_{j+1}
$=$
$-\alpha_j$
$-\beta_j$
$\alpha_j = v_j^* H v_j$
$\beta_{j+1} = v_{j+1}^* H v_j$
H
V
T
$V = \begin{pmatrix} | & & | \\ v_1 & \dots & v_m \\ | & & | \end{pmatrix}$
$T = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & & 0 & \\ & & \ddots & & \\ & & \ddots & \ddots & \\ & 0 & & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix}$
$HV = VT$
$=$

Figure 3: The Lanczos algorithm starts from a random vector v_1 and constructs in m iterations a basis of the Krylov space $\mathcal{K}_m(v_1) = \text{Span}\{v_1, Hv_1, H^2 v_1, ..., H^{m-1}v_1\}$ for an Hermitian matrix H. The Lanczos recurrence (top panel) along with the definitions of α_j, β_j ($\beta_1 \equiv 0$) construct the Lanczos vectors v_j (which are orthonormal to each other). Bottom panel: Regroup the Lanczos vectors into a matrix $V = [v_1, ...v_m]$ and form the tridiagonal matrix T. Observe that $HV = VT$: T is similar orthogonal to H, and has thus the same spectrum. The final step is to diagonalize T, which is fast using standard routines (it is small and tridiagonal). In practice, m can be quite small $(100, 200)$ to converge a few low-lying eigenvalues of very large matrices (we deliberately omitted the error terms in the formulas above, assuming that m is large enough). The eigenvectors of H are obtained by applying V to the eigenvectors of T, which is usually done in a second pass of the Lanczos recurrence. The crucial points of the Lanczos algorithm is that it only requires: (i) 3 vectors to be stored at the same time, (ii) a matrix-vector multiplication routine, which can be quite fast using matrix-free methods, and/or parallel implementations (with shared or distributed memory systems). The Lanczos algorithm is the specialization of the Arnoldi algorithm to Hermitian matrices.

such as with Chebyshev polynomials, can target an energy range within the original spectrum. Finally, eigenproblems with structures, such as the unitary operator arising in Floquet driving, can also benefit from a spectral transform.

Shift-invert transform — The *Shift-Invert* (SI) transformation is, as of today, probably the most efficient way of getting exact (machine precision) eigenpairs for lattice MBL Hamiltonians in the full parameter regime (record is $L = 26$ for the XXZ spin chain [11]). When the Hamiltonian is slightly less sparse than the XXZ Hamiltonian, filter transformations (as described below) become as competitive. Eigenstates for larger systems can be obtained with MPS methods (see Section III), but only quite deep in the MBL phase, and precision has to be discussed. As already mentioned, the SI method deals with the operator $G = (H - \Lambda)^{-1}$. It is not specific to MBL and appears in various physical contexts (*e.g.* for the Anderson problem). The eigenstates of H close to λ will be relegated to the ends of the spectrum of G, as readily checked. Moreover taking the inverse will ensure large gaps in the extrema of G as the original problem has small level differences. To make progress, we remark that one does not need to explicitly compute matrix elements of G (which is a dense operator) to obtain its extremal eigenvalues: it is sufficient to know how to apply G to a vector, and use iterative methods (Lanczos-like). We thus need to be able to solve the linear problem $Gx = y$ or in other words $(H-\lambda)y = x$(where y is the unknown). The solution of this linear system can be found using iterative methods (*e.g.* with the GMRES or Jacobi algorithms) or using a direct method that computes and exact decomposition of $H-\lambda$, such as the LU decomposition. For several many-body problems, we find the latter to be more efficient and stable (albeit more demanding in terms of memory), but for specific problems (such as strongly diagonal) iterative methods may be preferable.

LU decomposition — We decompose $H - \lambda$ in a lower triangular matrix L with 1's on diagonal and U an upper triangular matrix. This is done using techniques similar to Gauss elimination. A LU decomposition cannot always be performed (*e.g.* for non-invertible matrices), but a PLU decomposition can, where P is a permutation matrix that swaps some rows and columns (numerical implementations of LU decomposition always perform permutations in order to speed up later operations). For symmetric matrices, one can use a simpler form LDL^t with D a diagonal matrix. Once the LU decomposition is done, the job is done as we simply need to first solve the problem $Lz = x$ (z is the unknown), which is quite easy as L is lower triangular. We then solve $Uy = z$ (y is our unknown), which is again easy.

Shift-invert in practice — The LU decomposition and the full shift-invert procedure has been implemented in several libraries (*e.g.* in python). For large systems, we recommend using the SLEPc and PETSc parallel libraries. They are nicely interfaced to efficient parallel solvers (Strumpack, MUMPS) that perform the LU decomposition in parallel using the sparsity of H. They may however require a lot of memory (this is currently the bottleneck of the full method). The figure of merit is not so much the original number of non-zeroes of H, but rather the fill-in ratio (how many zeroes in the factors L, U are created per zero in H). This fill-in ratio is quite unpredictable for a given matrix H, but for MBL lattice models one often finds that it is unfortunately quite high. The associated repository contains an example of a SI computation for the XXZ spin chain Eq. 1. A more complete and pedagogical introduction to the method for MBL can be found in Ref. [11].

Polynomial filter methods — The idea is there to use a specific polynomial $G_{\sigma,M}(H) = \sum_{k=0}^{M} a_k H^k$ which is chosen such that its eigenvalues are large within a band $[\lambda-\sigma, \lambda+\sigma]$ around the target ("band-pass filter"). It is well-know that Chebyshev polynomials are optimal (in a mathematical sense) to construct band-pass filters, and they are often chosen to perform this filter transform. A nice illustration of the idea can be found in Figure 1 of Ref. [12]. Instead of specifying σ, one often asks for a certain number of eigenvalues around λ. In this way, the method requires an estimate of the density of states as well as the extreme values of H to be well-calibrated (e.g. to pre-compute the appropriate values of the coefficient a_k as well as the maximal order M). The Lanczos (or equivalent) algorithm is then applied to G. The polynomial filter method was first introduced in applied mathematics (to the best of our knowledge) and in the MBL context through the POLFED algorithm [12], which P. Sierant presented during his lecture at this school.

Key aspects of the method are that: (i) one only needs to be able to *apply powers* of H to (several) vector(s) in the Lanczos algorithm. Taking advantage of recurrence properties of the Chebyshev polynomials, the Clenshaw algorithm allows to reduce the number of applications of H. This provides an efficient method whenever H can be applied fast. (ii) Memory requirements are typically much smaller than with direct LU solvers for the shift-invert method. However, the maximal order M increases fast with the size of matrices, and we are thus trading memory to CPU computations: time to solutions can be long with filter methods. (iii) Most operations in the algorithm can be straightforwardly parallelized, with openMP using block matrices for Lanczos vectors, or through MPI for the matrix-vector computations.

In practice, benchmarks in Ref. [12] indicate that it is slightly less efficient than shift-invert and LU factorization for the XXZ model, but can reach larger sizes for models which have longer-range interactions (such as the $J_1 - J_2$ model), at the price of longer runs (overall there is a memory/CPU time tradeoff that is model-depedent). We finally note that there is room for improvement in the practical implementation of the method presented in Ref. [12], and that the recent filter algorithms of the SLEPc library should allow to quite efficiently code them. Writing a generic open source parallel code for the Polynomial filter method would be a great contribution to the field of MBL simulations.

Spectral transforms for Floquet problems — Ref. [13] introduces and exploits another spectral transform very useful for Floquet problems described by an evolution operator U. It takes the form of the geometric sum $g_k(U) = \sum_{m=0}^{k} \exp(-im\phi_\lambda) U^m$ where ϕ_λ is the targeted phase of the eigenvalue of U. In this way, eigenvalues around the target angle acquire a much larger amplitude than unity (and eigenvalues far from it have a much lower amplitude) such that they can now be easily reached with iterative methods such as the Arnoldi algorithm. The key issue is again to have a fast application of U to be able to apply $g_k(U)$ rapidly.

2.4 Extracting physical information from eigenpairs

Here we provide a collection of remarks and good practices for MBL simulations, once eigenpairs have been obtained. The notebooks in the associated repository are also useful sources of information.

2.4.1 Eigenstatistics

For historical and simplicity reasons, the first quantities considered for a MBL problem are eigenvalues and eigenstates statistics. First let us assume an ordered energy spectrum

$$E_0 < \cdots < E_n < E_{n+1} < \cdots < E_{\max}$$

and start with an analysis of energy gaps $s_n = E_n - E_{n-1}$. The study of gap statistics and distribution probability $P(s)$ has a long history in physics and it should allow in principle to distinguish for instance an ETH phase with level repulsion $P(s) \propto s \exp\left(-cs^2\right)$ (so-called Wigner surmise, $\beta = 1$ for the gaussian orthogonal ensemble (GOE) of random matrix theory, valid for real Hamiltonians such as Eq. 1), from a MBL phase with uncorrelated levels $P(s) = \exp(-s)$ (Poisson distribution). However these formula assume that the average density of states is flat, which is not the case for many-body problems. There are unfolding procedures for rendering a flat density of states from a numerically obtained spectrum. However there are possible biases in doing the unfolding (see *e.g.* Ref. [14]).

It is customary to consider instead gap ratios $r_n = \frac{s_n}{s_{n+1}}$ which do not suffer from this issue (the density of states cancel in the ratio in some sense), and one often introduces the reduced gap ratio $\tilde{r}_n = \min(r_n, 1/r_n)$ such that $0 \leq \tilde{r}_n \leq 1$ and $P(\tilde{r}) = 2P(r)$. There are equivalent known forms for $P_{\text{Poisson}}(\tilde{r}) = 2/(1+\tilde{r})^2$ for the MBL phase and $P_{\text{GOE}}(\tilde{r}) = \frac{27}{8}\frac{\tilde{r}+\tilde{r}^2}{(1+\tilde{r}+\tilde{r}^2)^{5/2}}$, which is a good approximate for the GOE ensemble [15]. Average values are often considered $\langle \tilde{r} \rangle_{\text{Poisson}} \simeq 0.38629$ and $\langle \tilde{r} \rangle_{\text{GOE}} \simeq 0.5307$ in numerical simulations as limiting cases (the latter is a numerical estimate and does not derive from the formula above).

While often the first quantity that is computed for a new MBL model, $\langle \tilde{r} \rangle$ and gap statistics are not a panacea. First, obviously $\tilde{r}_n$ and $\tilde{r}_{n+1}$ are correlated when coming from the same spectrum. Second, one needs to be very cautious at strong disorder where degeneracies can occur: imagine a very weak bond in a random bond model, this will effectively break the chain and results in almost degenerate states with $r_n \simeq 0$ and $\langle \tilde{r} \rangle < \langle \tilde{r} \rangle_{\text{Poisson}}$. Also, sometimes (discrete) symmetries are present in a model, but they are unknown to or untreated (that is the Hamiltonian is not block-diagonalized) when computing spectral statistics. This results in different distributions for $P(r)$ and estimates for $\langle r\rangle$ than the ones above, which could lead to mistakes in identifications of the correct physics. Fortunately, a recent work [16] provides predictions for such compound spectra. Another issue is machine precision, which for large enough systems, may not be sufficient to compute correctly gaps (this typically occurs around $N \simeq 25$ for the XXZ spin chain when using single precision). Finally, the average gap ratio is often found phenomenologically to exhibit strong finite-size effects close to the MBL transition.

Another important issue which is worth mentioning while one has the many-body density of states in hand is to be careful when large energy scales are involved. If for instance the disorder (or Δ) is too strong, the many-body density of states can exhibit large bands (and several mini-bands inside these bands) on finite systems, which are *not* representative of the thermodynamic limit. This can result in false-positive detection of MBL phases (see a discussion in Ref. [17]).

Another interesting, less explored, route is to consider statistics of eigenstates $|\Psi_n\rangle$, when expended in a (computational) basis $|\Psi_n\rangle = \sum_i \Psi_n^i |i\rangle$. A useful indicator, well studied in the Anderson localization context, is the inverse participation ratio (IPR) $\mathrm{IPR}(|\Psi_n\rangle) = \frac{\sum_i |\Psi_n^i|^4}{\sum_i |\Psi_n^i|^2}$ and its generalization $\mathrm{IPR}^q(|\Psi_n\rangle) = \frac{\sum_i |\Psi_i^n|^{2q}}{\sum_i |\Psi_i^n|^2}$ (one often omits the denominator for normalized eigenstates). It provides insights on how $|\Psi_n\rangle$ is delocalized in the basis $|i\rangle$ (the interpretation of localization in many-body bases is somewhat less clear than for single-particle problems).

Another useful indicator of localization is the comparison between two eigenstates $|\Psi_n\rangle$ and $|\Psi_{n'}\rangle$ through their Kullback-Leibler divergences $\mathrm{KL} = \sum_i |\Psi_i^n|^2 \log\left(\frac{|\Psi_n^i|^2}{|\Psi_{n'}^i|^2}\right)$ which is know to average to a finite value $\langle \mathrm{KL}\rangle_{\mathrm{GOE}} = 2$ in the GOE ensemble (where eigenstates are ‘similar’) and to diverge in the MBL phase (where eigenstates are qualitatively different).

Note that eigenstate statistics as defined above are *basis-dependent* quantities, and it is natural to ask whether they actually make sense. From experience (and without rigorous claim), it appears that the IPR or KL divergence (and their scaling with system size) expressed in preferential bases where the Hamiltonian is sparse (such as the $\{S^z\}$ basis for the XXZ spin chain) contain a lot of physical information.

2.4.2 Observables

With the eigenstates expressed as $|\Psi_n\rangle = \sum_i \Psi_n^i |i\rangle$, it is straightforward to compute observables $\mathcal{D}$ which are diagonal in the computational basis (for instance the spin magnetization $\mathcal{D} = S_j^z$ in the $\{S^z\}$ basis for the XXZ spin chain). The expectation value is simply given by $\langle \Psi_n|\mathcal{D}|\Psi_n\rangle = \sum_i |\Psi_n^i|^2 (\mathcal{D}|i\rangle)$, where the term in parenthesis can be pre-computed (and stored as a vector of size $|\mathcal{H}|$) if this observable is to be measured often.

Off-diagonal observables do not enjoy this property, and are either computed on-the-fly (if they are simple enough) or pre-computed and stored as matrices if they are too costly to be computed on the fly. Typically observables are sparse matrices, as the Hamiltonian, and they thus should be stored in the same way (*e.g.* with the same parallel decomposition).

2.4.3 Computation of entanglement spectrum and entropy

There is one specific quantity for which numerical computation is worth spelling out: the entanglement entropy and the related entanglement spectrum. As seen in Sec. 2.1, bipartitioning the basis into two parts A, B is beneficial for speeding up computations as well as decreasing the memory load, but it turns also extremely useful for computation of the entanglement spectrum. The reduced density matrix in subsystem A of a state $|\Psi\rangle$ is obtained by tracing over all degrees of freedom B: $\rho_A = \mathrm{Tr}_B \left|\Psi\right\rangle \left\langle\Psi\right|$. It is useful to think/reshape $|\Psi\rangle$ as a matrix $|\Psi\rangle = \sum_{ij} \hat{\Psi}_{ij} \left|c_i\right\rangle_A \left|c_j\right\rangle_B$ where $\{|c_i\rangle_A\}$ ($\{\left|c_j\right\rangle_B\}$) form orthogonal basis of the A (B) subsystem: we have just created these two basis already, so let's use them. This matrix representation of the wave-function can be found in Figures 1,2. The reduced density matrix elements can be written as $\rho_{ij} = \sum_k \hat{\Psi}^*_{ik} \hat{\Psi}_{kj}$, but it is actually not needed to form ρ. Indeed, its eigenvalues λ_i are nothing but the square of the singular values ($\lambda_i = \sigma_i^2$) in the singular-value-decomposition (SVD) of $|\Psi\rangle = \sum_i \sigma_i \left|d_i\right\rangle_A \left|d_i\right\rangle_B$. $\{\lambda_i\}$ is often referred to as the entanglement spectrum, from which the von Neumann entanglement entropy $S = -\sum_i \lambda_i \log(\lambda_i)$ can be deduced.

We can thus directly perform the SVD of $|\Psi\rangle$ as a matrix in the already-computed basis. Note that the reduced density matrix has blocks, either inherited from symmetries of the system (*e.g.* conservation of magnetization) or from a constrained Hilbert space, which are nothing but the blocks encountered in the construction of the basis. Thus to obtain all singular values, we can perform the SVD of each block independently, resulting in a large computational gain. Another good point is that the singular values σ_i are obtained with machine precision, thus one obtains a much higher precision for the entanglement spectrum $\{\lambda_i\}$ than with the diagonalization of ρ_A. This can be useful if one interested in the very small eigenvalues of ρ_A.

Note that the entanglement cut is fixed when choosing the A/B bipartition in constructing the basis. When several entanglement cuts are desired (and not only the one used for the construction of the basis), one can pre-compute the corresponding basis bipartition for each entanglement cut and store a single table that stores the index shuffle between the computational basis $\mathcal{B}_1$ and this different entanglement-cut basis $\mathcal{B}_2$ (i.e. $R[c_i^{\mathcal{B}_1}] = c_j^{\mathcal{B}_2}$). As the entanglement computation is usually performed on a single node after all the massively parallel computations have been performed, this amounts to storing there one extra vector of the size of the Hilbert space.

2.4.4 Dealing with disorder

There is nothing much different for simulations of MBL than from other disordered systems. Quenched disorder implies repeating the same type of simulations for a larger number of disorder realizations, which clearly impacts the computational time. The number of disorder realizations needed can be very large close to the MBL transition, as the probability distribution of observables typically gets broader [18] (and can speculatively grow without limit, in analogy with infinite-disorder quantum critical points).

A first basic advice is to save the random disorder realization (e.g. $\{h_i\}$ for the XXZ spin chain) or at least the random generator seed in the output of all simulations, for matters of reproducibility and comparison with other methods. This is also useful to identify samples which are not representative of the distribution, and can lead to Griffiths effects.

Another issue, perhaps more specific to MBL, was put forward in Ref. [19] in distinguishing the different procedures in performing statistics and computing expectation values of observables and their variances. One can average over (i) disorder realizations, (ii) eigenstates in the same disorder realization, and (iii) different spatial positions (*e.g.* position of the entanglement cut, see Sec. 2.4.3). While one expects that in the thermodynamic limit, the order in which these averages are performed does not matter, it clearly does for finite systems where the data variance differ for different procedures and models [19]. Therefore it is worth saving all individual eigenstates and spatial position data and not to average before outputting.

2.5 Time evolution

In MBL simulations, the dynamical evolution after a quench is an important setup to make contact with experiments. For an isolated Hamiltonian system, the time evolution $|\Psi(t)\rangle = U(t)|\Psi(0)\rangle$

of an initial state $|\Psi(0)\rangle$ is provided by the evolution operator $U(t) = \exp(-\mathrm{i}Ht)$. In this situation, iterative time-expansion techniques such as the Krylov method described below are efficient for large sparse systems. On the other hand, time evolution generated by a dense, non-Hamiltonian, evolution operator (such as encountered in some quantum circuits or Floquet systems), require different approaches, which we briefly discuss in Sec. 2.5.3.

2.5.1 Krylov method for time evolution after a quench

Starting from an initial state $|\Psi(0)\rangle$, the Krylov approach approximates the wave-function after a small time step $|\Psi(t+dt)\rangle = \exp(-\mathrm{i}Hdt)|\Psi(t)\rangle$ by its projection into the Krylov space $\mathcal{K}_m(|\Psi(t)\rangle) = \text{Span}\{|\Psi(t)\rangle, H|\Psi(t)\rangle, H^2|\Psi(t)\rangle, ..., H^{m-1}|\Psi(t)\rangle\}$ generated from $|\Psi(t)\rangle$. $\mathcal{K}_m(|\Psi(t)\rangle)$ can be generated as in the Lanczos/Arnoldi algorithms, to obtain the matrix V of Lanczos/Arnoldi vectors and the small $m \times m$ matrix T which is the orthogonal projection of H into $\mathcal{K}_m(|\Psi(t)\rangle)$. The final approximation becomes:

$$|\Psi(t+dt)\rangle \approx V \exp(-\mathrm{i}Tdt)e_1 \tag{2}$$

which can be applied iteratively ($e_1 = (1, 0, 0 \ldots, 0)$ is the first canonical basis vector of $\mathbb{R}^m$). The small matrix T is easy to exponentiate. The Krylov method requires to store m vectors of size $|\mathcal{H}|$ (instead of 3 in the Lanczos algorithm), which can be prohibitive. However, the method naturally parallelizes well (based on sparse matrix vector multiplication), and moreover there exist restart schemes which require less storage. Note that there are other schemes which allow the computation of the action of the exponential of the matrix to a given vector, such as using Padé approximation with a scaling and squaring method [20], or an approximation of the exponential function with Chebyshev polynomials (see Ref. [21, 22], this requires less storage but in general more iterations).

There are rigorous bounds (depending on dt, m and spectral properties of H) of the error made on $\exp(-\mathrm{i}dtH)|\Psi_0\rangle$ by using its Krylov approximation [23]. In practice, for dt reasonably small (typically ~ 1 in the inverse typical energy scale of H), m can be small too and storing V in parallel is manageable. There are simple checks of the validity and numerical stability of the full computation that can be performed. First, as the time evolution is unitary, the norm of $|\Psi(t)\rangle$ should be conserved. Second, one can do two calculations with two fixed values of $m, m_2 > m$: if results are identical within numerical accuracy, they are correct. Finally, in the same vein, one can work with a fixed m, and simply reduce the time step dt (say by half): identical results again ensure convergence. A starting code for the Krylov time evolution of the random-field XXZ chain is provided in the repository, based on the SLEPc library.

Once the state $|\Psi(t)\rangle$ is obtained, one can probe its physical properties with various observables or through the computation of its entanglement entropy, using the same techniques as described in Sec. 2.4. Time-dependent correlators require more book-keeping, as described in the next section. Note that on top of disorder averaging, one can also average over the initial states $|\Psi(0)\rangle$.

2.5.2 Computation of time-dependent correlators, of OTOC

Time-dependent correlators after a quench, e.g. density-density correlations $C_{ij}(t) = \langle\Psi(0)|S_i^\alpha(t)S_j^\alpha(0)||\Psi(0)\rangle$ with $S^\alpha(t) = \exp(\mathrm{i}Ht)S^\alpha\exp(-\mathrm{i}Ht)$, can be obtained by performing two Krylov evolutions: $\widetilde{|\Psi(t)}\rangle = S_j^\alpha(0)|\Psi(0)\rangle$, and the usual $|\Psi(t)\rangle = \exp(-\mathrm{i}Ht)|\Psi(0)\rangle$ (on which we apply S_i^α to form $\widehat{|\Psi_(t)}\rangle = S_i^\alpha|\Psi(t)\rangle$), to finally obtain the correlator as an overlap $C_{ij}(t) = \widehat{|\Psi(t)}|\widetilde{\Psi}(t)$. The propagation has to be performed simultaneously for both states in order to form this overlap for all required values of t. In case of a disordered system and for which individual C_{ij} is wanted, this can be costly. Note that depending on the value of $\alpha = x, y, z$, the symmetry of the model (e.g. $U(1)$ symmetry with conservation of S^z in spin chains) and the basis chosen, one may have to deal with propagations in different symmetry sectors of the Hilbert space, which can also be different than the one of the initial state.

Another interesting quantity is the Out-of-Time-Ordered-Correlator (OTOC) defined as $\mathcal{C}_{AB}(t) = \langle\Psi|A^\dagger(t)B^\dagger(0)A(t)B(0)|\Psi\rangle$. Usually the expectation value denotes the average over the thermal ensemble, but one can also study the OTOC over a single state $|\Psi\rangle$ which can be a basis state, or a random vector for instance to emulate the thermal ensemble at infinite temperature. The OTOC is usually computed using full-diagonalization, but it is also possible to perform a double Krylov expansion twice in the following way. For each time t, perform a Krylov propagation of (1) $\exp(-\mathrm{i}Ht)B(0)|\Psi\rangle$ and (2) $\exp(-\mathrm{i}Ht)|\Psi\rangle$, apply the operator A to obtain the intermediates state $|\phi(t)\rangle = A\exp(-\mathrm{i}Ht)B(0)|\Psi\rangle$ and $|\varphi(t)\rangle = A\exp(-iHt)|\Psi\rangle$. Then perform a second propagation $\exp(\mathrm{i}Ht)|\phi(t)\rangle$ and $\exp(\mathrm{i}Ht)|\varphi(t)\rangle$ (note the sign change) to obtain $|\overparen{\phi(t)}\rangle = \exp(\mathrm{i}Ht)|\phi(t)\rangle$ and $|\overparen{\varphi(t)}\rangle = \exp(\mathrm{i}Ht)|\varphi(t)\rangle$. The final value for the OTOC is then obtained as the overlap between $|\overparen{\phi(t)}\rangle$ and $B|\overparen{\varphi(t)}\rangle$. Again care has to be taken on which Hilbert space sectors to work with depending on the nature of A and B operators. This method works for large systems (larger than those available with full diagonalization), but clearly its t^2 scaling is unfavorable for large times.

2.5.3 Time evolution in quantum circuits and Floquet systems

A quantum circuit describes the time evolution of N local degrees of freedom; in the case of qbits (to which we restrict ourselves here for the sake of simplicity), the Hilbert space size therefore scales as $|\mathcal{H}| = 2^N$. The evolution operator from time t to time $t+1$ writes as a tensor product of (usually) local operators $u^{(j,l)}_{t+1|t}$:

$$U_{t+1|t} = \prod_{l=1}^{D} \bigotimes_{j=1}^{N} u^{(j,l)}_{t+1|t}, \tag{3}$$

where j is a position index, indicating the site at which the (finite) support of $u^{(j,l)}_{t+1|t}$ is centered, and l is a layer index (the total number of layers, D, is the *depth* of the circuit).

Typically the $u^{(j,l)}_{t+1|t}$ are $1-$ or $2-$qbits gate, making the time-evolution computationally efficient, provided that the depth D is not too large (typically, one considers $D \simeq 2$ for most applications, see *e.g.* Sec. 3.1.1). However contracting the $\{u^{(j,l)}_{t+1|t}\}$ explicitly to form $U_{t+1|t}$ yields a *dense* operator, whose storage quickly becomes too expensive. Some better strategies are needed to compute the time evolution of a quantum state.

The strategy of this section (storing entirely the wave-functions) limits the computation to 28-34 qbits on a single node. On a parallel architecture, one can reach larger number of qbits (up to $45-48$, see Ref. [24–26]). We should also mention alternative, more involved approaches:

- *Clifford stabilizer codes* — This method efficiently simulates circuits mostly made out of Clifford gates (for qbits, those gates are products of Pauli operators); the cost is exponential in the number of non-Clifford gates [27].

- *Tensor network representations* – Any n qbits gate can be viewed as a $2n$ legged tensor. Algorithms exploiting such representations are exponentially costly in the amount of entanglement of the quantum state [28–30], and are useful for simulating noisy intermediate-scale quantum computers [31]. We refer to lecture notes [32] or review [33] on tensor network techniques for time evolution, as well as to Section III for a more modest description of their application in the MBL context.

While explicit models for shallow quantum circuits can be provided *ad hoc*, it is important to note that (i) low depth quantum circuits are a framework for the time evolution of isolated quantum systems (see Sec. 3.1.1) and (ii) some (but not all) Floquet systems, whose Hamiltonian is periodic in time $H(t+T)=H(t)$, can be also cast as low depth quantum circuits. This can be the case for instance for binary drives: $U(T)=\exp(-\mathrm{i}H_1T/2)\exp(-\mathrm{i}H_2T/2)$, where one switches between two simple Hamiltonians H_1, H_2 every half-period.

Time-evolution in practice — We now describe in detail the implementation of time-evolution using the strategy of storing the entire wave-function. While we chose for simplicity to consider the above binary drive, the method applies to any circuit. Typically H_1, H_2 are local and diagonal in some simple basis, but they do not commute with each other: for instance $H_1=\sum_i S_i^z S_{i+1}^z + hS_i^z$ and $H_2 = g\sum_i S_i^x$. The evolution operator is dense in, say, the $\{S^z\}$ basis, which forbids its storage. In this case, one can resort to matrix-free methods. The application of $\exp(-\mathrm{i}H_1T/2)$ is easy as it is diagonal. One still needs to apply $\exp(-\mathrm{i}H_2T/2)=\prod_{j=1}^N \exp\left(-\mathrm{i}gT/2S_j^x\right)$, which can be done efficiently by acting sequentially with each local operator $u_j=\exp\left(-\mathrm{i}gT/2S_j^x\right)$. Importantly, while $\exp(-\mathrm{i}H_2T/2)$ is a dense matrix in the computational basis (whose action on a vector necessitates $|\mathcal{H}|^2$ operations), each u_j is a sparse matrix, leading to an efficient method requiring only $|\mathcal{H}|\log|\mathcal{H}|$ operations. On a distributed architecture, each one of the N matrix-vector products can easily be parallelized over a large number of CPUs/GPU units, using *e.g.* the MPI or CUDA frameworks. In the very specific example above where one needs to switch rapidly between the $\{S^z\}$ and $\{S^x\}$ basis, an interesting alternative is to use fast Hadamard transform (which complexity also scales as $|\mathcal{H}|\log|\mathcal{H}|$), as first proposed in Ref. [34].

Finally, note that the evolution operator of a generic Floquet drive, $\exp(-\mathrm{i}H_\alpha T/2)$, cannot in general be written as a shallow circuit of local gates, and knowing its action on a given vector via the above method is no longer efficient. In that case, one can turn to a Krylov expansion of the operator using the technique described in Sec. 2.5.1 (see *e.g.* Ref. [35]).

3 Applications of Matrix Product State methods to the MBL problem

There are excellent reviews [33,36], lectures notes [32] and websites (https://tensornetwork.org/) that explain in great details the basis and formalism of matrix product states (MPS) and operators (MPO) representation of quantum many-body systems, as well as efficient algorithm to obtain their dynamics and ground-states within this formalism. In this section, we provide a superficial description of the algorithms and their range of applications and specificities in the MBL context. We distinguish them from other non-exact methods (briefly presented in Section 4) as (i) their precision can be almost as good as with the exact methods introduced in Section 2 (albeit in a certain parameter range), (ii) they are naturally singled out by the recognition of MBL eigenstates as low entanglement states.

3.1 Dynamics with MPS / MPO methods

3.1.1 Dynamics after a quench with the Time-Evolution Block Decimation

The MPS formalism is well set-up to perform time evolutions of one-dimensional lattice models. Let us start with the simpler case of nearest-neighbour interactions for which the efficient Time-Evolution Block Decimation (TEBD) technique [28, 37] is available. In this situation, the Hamiltonian can be decomposed into two sets of mutually commuting bond terms $H=\sum_{i\text{ even}} H_{i,i+1} + \sum_{i\text{ odd}} H_{i,i+1}$. The idea is first to approximate the infinitesimal evolution operator $U(dt) = \exp(\mathrm{i}Hdt)$ by the Trotter formula $U(dt) = \exp(\mathrm{i}dtH_{\mathrm{even}})\exp(\mathrm{i}dtH_{\mathrm{odd}}) + O(dt)$. Higher-order formulas (such as $U(dt) = \exp(\mathrm{i}dtH_{\mathrm{even}}/2)\exp(\mathrm{i}dtH_{\mathrm{odd}})\exp(\mathrm{i}dtH_{\mathrm{even}}/2) + O(dt^2)$) are not more complex to use in practice. As obvious, the Trotter formulas for nearest-neighbor interactions readily leads to a quantum circuit of the form Eq. 3 with only 2-qbits gates.

The TEBD algorithm simply applies such circuit to a state in a MPS form. Since all terms in $H_{\mathrm{even/odd}}$ commute with each other, it is sufficient to consider the application of $u_{i,i+1} = \exp\big(\mathrm{i}dtH_{i,i+1}\big)$. Applying the unitary 2-qbits gate on the bond $(i, i+1)$ to a MPS in a canonical form results in a state which can also be cast as a MPS in a canonical form (through a SVD), albeit with an increased dimension for this bond. In order to keep the computational effort manageable, the SVD is truncated to keep only the larger χ_m singular values. Due to the independent All the procedure is detailed in e.g. Ref. [32]. An introductory code, based on the Itensor library, for the random-field XXZ model can be found in the repository.

The computational time of the method scales as $L\chi_m^3 t$. In practice, the accuracy of the method is limited at large t by the amount of entanglement in the evolved state $|\Psi(t)\rangle$. To probe high-energy physics in the MBL context, one usually starts from a low-entangled state (such as a product state) with an energy density corresponding to infinite temperature. In an ergodic phase, the von Neumann entanglement entropy S grows very fast (linearly) in an ergodic phase towards a saturation value which is extensive with the subsystem size: the TEBD method will thus fail rapidly for large system sizes. Fortunately, this is not the case in the MBL phase as the growth of entanglement is slow (logarithmic) there, such that the TEBD works excellently (of course reaching times larger than $\sim 10^4$ is difficult though). Actually, it was shown that the TEBD (and other MPS methods such as TDVP, see below) also work well in some cases in the ergodic phase near the phase transition (see *e.g.* Ref. [38, 39]), which can be rationalized as the entanglement growth is slower there too (with a sub-linear power law) [40].

The TEBD algorithm is formulated above assuming nearest-neighbor interactions, due to the use of Trotter formula. It parallelizes nicely as any gate (in the even and odd sums) can be applied independently. The TEBD can be extended to slightly longer-range interactions (next nearest neighbours) through the inclusion of swap gates within the Trotter formulation [41]. An alternative for longer-range interactions, which does not rely on Trotter formulae, is to find directly an approximate MPO form $W^I(dt)$ for $\exp(\mathrm{i}Hdt)$, which is valid up to order I in dt. Explicit algorithms for deriving $W^1(dt)$ and $W^2(dt)$ have been derived in Ref. [42] and are implemented in ITensor. Furthermore, these approximates can be combined to have an even better approximation (e.g. $W^2(\tau_+)W^2(\tau_-) = \exp(\mathrm{i}dtH) + O(dt^3)$ with $\tau_\pm = (1 \pm \mathrm{i})d\tau/2$, higher-order can be found in Ref. [43]).

An alternative to time-evolution with MPS is through the time-dependent variational principle (TDVP) [44, 45], which is also very efficient, but won't be covered in the current version of these lecture notes. The Tenpy library [32] has an implementation of TDVP. We refer to the review Ref. [33] for an overview of time-evolution MPS algorithms.

3.1.2 Dynamics in open systems

So far, our discussion was limited to close systems and dynamics following an unitary time evolution. Implementations of lattice MBL in experiments on the other hand always imply coupling to a bath, even if possibly small: it is thus useful to be able to simulate such situations. Also, adding coupling to a weak boundary bath gives access to transport properties in an efficient fashion for large systems in the ergodic phase. An excellent account of MPS / MPO (and other) simulation techniques for open systems can be found in Ref. [46], we just provide here basic statements in relation to MBL simulations.

The discussion will be limited to the simulation of the Lindblad equation which describes the time evolution of the system's density matrix ρ in a Markovian approximation :

$$\frac{d\rho}{dt} = \mathrm{i}[\rho, H] + \sum_i \gamma_i [L_i \rho, L_i^\dagger] + [L_i, \rho L_i^\dagger]. \tag{4}$$

L_i are Linblad operators that provide an explicit form for the coupling to the bath, and γ_i the associated coupling rate. For instance for an atomic system, one can consider Lindblad operators as annihilation operators $L_i = b_i^\dagger$ to simulate particle loss, or as density operators $L_i = n_i$ to simulate dephasing, or a local density measurement. In the context of spin systems, $L_i \propto S_i^\pm$ allows to probe spin transport when i is restricted to be an edge spin.

The solution to the time-evolution equation is $\rho(t) = \exp(\mathcal{L}t)\rho(0)$ where the action of the Liouvillian $\mathcal{L}[\rho]$ on the density matrix is the right hand side of Eq. 4. Assuming $\rho(t)$ is computed, there are two interesting situations that are helpful in the MBL context. First is the change of behavior in the short to intermediate-time behavior of observables after a quench, which is induced by the coupling to a bath. For instance, how does the imbalance decay $\mathcal{I}(t) = \mathrm{Tr}(\rho(t)\mathcal{I})$ decays from its long-time limit in the MBL when a bath is added? This has been addressed in several works using MPS / MPO methods (see *e.g.* Refs. [47,48]), using different forms for the bath, and in relation with experiments where the coupling to the bath can be explicitly tuned.

The other situation corresponds to the steady-state limit where $\rho(t)$ converges at long times to a non-equilibrium steady state ρ_∞ which, assuming a small coupling to the bath, can then be probed to obtain valuable information on the response properties of the closed system. A typical situation in the MBL context is to probe infinite-temperature spin transport in a 1d spin chain (conserving spin magnetization) by imposing two small reservoirs which only weakly couple to a single-spin at each end of the chain. The Lindblad operators take in this case the form $L_\pm^{\text{left}} = \sqrt{1 \pm \mu} S_1^\pm$ and $L_\pm^{\text{right}} = \sqrt{1 \mp \mu} S_L^\pm$. They will force the system to take magnetization values $S_1^z = \mu/2$ and $S_N^z = -\mu/2$ in the long-time limit, and will thus induce a non-trivial magnetization profile $\langle S_i^z \rangle_{\rho_\infty}$ and a non-zero spin-current $j_i = \langle S_i^x S_{i+1}^y - S_i^y S_{i+1}^x \rangle_{\rho_\infty}$ in the steady-state. Typically, one considers the current in the middle of a long-chain to avoid boundary effects. The type of transport can be then probed by considering how the current scale with the system size: diffusive transport is signaled by a $j \propto 1/N$ decay, whereas $j \propto 1/N^a$ (with $a > 1$) signals subdiffusive transport. This approach has been undertaken in several MBL works [49,49–51].

How can we compute the evolved state $\rho(t)$? We can again think of approximating the infinitesimal time evolution $\exp(\mathcal{L}dt)$ as done in the previous section. This time now ρ is a density matrix, and $\mathcal{L}$ a superoperator. We can write the state ρ as a MPO or, as often done in practical implementations, as a MPS $|\rho\rangle$ in a doubled-Hilbert space (*e.g.* reshape a 2×2 matrix as a 4-states vector). We can then perform a time evolution using a Trotter approximation (as in TEBD) if the interactions in H are on nearest-neighbors only and if the coupling to the bath is also local [52, 53]. Alternatively, we can use the $W^I(dt)$ approximations mentioned above to $\exp(\mathcal{L}dt)$. The efficiency of the MPO approach will be dictated by the value of the entanglement entropy of the super-state $|\rho(t)\rangle$, or in other words the operator entanglement entropy of $\rho(t)$ (given by its SVD). In the first usage (imposing bulk particle loss and/or dephasing dynamics, and following dynamics of observable after a quench), the method will be mostly efficient in the MBL phase, while the second usage (weak edge bath and monitoring transport) will on the other hand be efficient in a thermal phase, such that the density matrix departs not too much from identity. Let us also mention that in the case of probing $\rho(t \to \infty)$, one needs to make sure that the state is well converged, which may require long-time scales for large systems (even in a diffusive phase). Also, as for any MPS/MPO method, the finite-χ_m truncation effects have also to be carefully monitored.

Let us finally mention that an alternative to using a full-MPO method as above is to perform time-evolution with an MPS technique (such as TEBD), and implement the effect of the baths using the quantum trajectories technique [54], where Lindblad operators are applied to the evolved wave-function at certain times ('quantum jumps')).

3.2 Obtaining MBL eigenstates and eigentransformation

3.2.1 DMRG for excited states

The Density Matrix Renormalization Group (DMRG) is an extremely powerful method, now formulated in the MPS language, to obtain the ground-state of 1d many-body systems which sustain a low entanglement. It has a variational nature, and its crucial point is to optimize a local tensor (on one or two sites) in the MPS formulation of the variational state $|\Psi\rangle$ in order to minimize the energy $\langle\Psi|H|\Psi\rangle$, while the other tensors are left unchanged. This step is applied recursively along the chain ("sweep"). Due to the local nature of the optimization step, the effective matrix to diagonalize is of much lower dimension $\chi_m.d^2$ (χ_m maximal bond dimension, d size of the local Hilbert space – $d = 2$ for spin-1/2 chains) than the full Hilbert space. The low-entanglement requirement for the method to work is precisely given by the fact that an MPS state with maximal bond dimension χ_m has at most a von Neumann entanglement entropy $S = \log(\chi_m)$. We again refer to Ref. [32, 36] for a detailed exposition.

As mentioned in the introduction, we are usually not interested in ground-states for MBL problems, but rather to highly-excited eigenstates. Crucially, the MBL states in the MBL phase do fulfill an area law and hence should be captured by MPS in 1d. Conversely, any MPS based method to capture eigenstates is expected to break down in the ergodic phase as the entanglement entropy of eigenstates is extensive there.

Of course, one has to construct a different variational scheme to obtain MBL highly-excited eigenstates. The first type of methods just changes the operator over which to minimize: the choices of Sec. 2.3.2 appear natural. Minimizing the expectation value of $G = (H-\lambda)^2$ during a DMRG sweep has been performed in several works (En-DMRG [55], Ref. [56]), but as already mentioned the spectrum of this operator is very dense at low energies, and the DMRG procedure can fail in reaching the ground-state. The other possibility $G = (H - \lambda)^{-1}$ does not suffer for this problem, and allows to obtain very good eigenstates in the MBL phase, even reasonably close to the MBL transition. The price to pay is to find a MPS that approximates the best G applied to the current MPS. Two slightly different techniques (Shift-Invert DMRG [57], Ref. [58]) have been proposed, which are both based on solving the linear system equivalent to the inversion (similarly to what is discussed in Sec. 2.3.2).

An alternative method called 'DMRG-X' [59] is based on the local integral motions of picture valid in the MBL phase. Its idea is to search for a variational state that is the closest to a given product state (an input of the method). Instead of optimizing the MPS by minimizing the energy $\langle\Psi|H|\Psi\rangle$ (in the projected reduced basis where the tensors on the rest of the chain are contracted), one seeks the eigenstate of H (again in the projected basis) which has maximum overlap with the product state (which is morally a LIOM). The rest of the DMRG algorithm is unchanged. The DMRG-X avoids the small level spacing problem as nearby eigenstates in the MBL phase are totally different (they have LIOMs at totally different positions in general). An improvement on this DMRG-X algorithm has been proposed [60], which instead picks up the linear combination of few eigenstates (with a large overlap with the product state) which mimimizes the variance of H.

For all these methods, we need to know variational state $|\Psi\rangle$ is indeed a good eigenstate of the Hamiltonian. The figure of merit is the energy variance $\sigma^2 = \langle\Psi|H^2|\Psi\rangle - \langle\Psi|H|\Psi\rangle^2$. Ideally one would like σ^2 to be at machine precision, but this is rarely reached, except deep in the MBL phase. We refer to Refs [55–60] for presentation of the precision and results obtained with these extensions of the DMRG method.

3.3 Full diagonalization unitary as a MPO

MBL eigenstates can thus be well described by MPS, but one can go even further. It has been argued that the full unitary U which diagonalizes the Hamiltonian can be well represented by a MPO (a single tensor) in the MBL phase [61,62]. Note that this is a stronger statement (than just collecting the MPS eigenstates in a matrix) as it means that a single tensor (which can be of low dimension quite deep in the MBL phase) faithfully encodes the full diagonalization of H. With all the eigenstates being well approximated in a single shot, dynamics can be obtained too (contrary to MPS methods concentrating on a single eigenstate). A similar result has been obtained that allows the encoding of the Floquet evolution operator in a Floquet MBL phase as a MPO [63].

How can these results be put in practice? Again a variational scheme (as in DMRG) is possible to search for the optimal MPO that minimizes the Hamiltonian variance, except that this time we have to make sure that the MPO keeps unitary [64]. For this, Ref. [64] proposes to find U with a low-depth quantum-circuit ansatz (similar to Eq. 3) with 2-qbits gates and a few layers (the more layers, the more accurate the ansatz). Ref. [65] suggests an improvement by increasing the size of the local gates. This last method has been used to study MBL in one dimension on large spin chains [65] as well as in 2d [66]. For the MPO representation of the Floquet evolution operator, a TEBD-like algorithm was used, after which a DMRG-X computation is performed on this MPO to obtain its eigenstates [63].

Note that it is paramount for these methods to be efficient that the system is located in a full-MBL phase, where *all* eigenstates are many-body localized.

4 Discussion

4.1 Other methods not discussed in these notes

This section could be improved. Think of other methods? Your comments are welcome ...

Due to the new physics involved and the associated technical challenges, several other numerical methods have been proposed and used for the MBL problem. In this section, we just provide references to several of them along with a very brief description.

Real Space Renormalization Group for excited states (RSRG-X) — Great insights were obtained on the MBL problem thanks to real space renormalization group (RSRG) methods, adapted to excited states [67, 68]. They allow to capture eigenstates deep in the MBL phase and can also lead to phenomenological classical models (see e.g. Refs [69–71]) used to study the transition to the ergodic phase. The decimation in the RSRG and the study of the effective classical models often go through a numerical treatment, but the numerical simulations are perhaps less demanding than for the previously discussed methods. See [72] for a review of RSRG developments applied to the MBL field.

Series or cluster expansions have been also extended to study MBL physics [73] (the workhorse still remains the full or subset-diagonalization of the cluster).

Phenomenological truncations — Phenomenological renormalizations based on LIOM wavefunctions [74] or on basis truncation [75, 76] have also been proposed to extend quantum simulations to larger systems. A recent study [77] also showed that going to the Fock basis of Anderson localized states (for fermions models) and neglecting off-diagonal terms can lead to a quite accurate description of dynamics, even on long-time scales, when interactions between fermions are not too strong.

MPS/MPO methods — In this family, let us highlight two developments (one based on the the time-dependent variational principle [78], one on a density matrix truncation [79]) which hold promises for simulating efficiently dynamics in the ergodic phase, up to very long times.

In another direction, a MPO/MPS approximation has been proposed in the framework of a newly developed method based on the *influence matrix* [80].

Can stochastic methods capture MBL ? — In an original contribution, Ref. [81] suggests how to possibly use efficient quantum Monte Carlo (QMC) methods to probe MBL physics, again in relation with the existence of LIOMs. The idea is to first brute-force construct the best integral of motions L^p on a small patch p (containing few sites of the system) assuming it takes a simple form (with only two-body interactions). Then the original Hamiltonian is shifted with these LIOMs: $H_{\text{Shifted}} = H - \sum_p \lambda_p L^p$. The ground-state of H_{Shifted} can be sampled with quantum Monte Carlo (it is designed to have no sign problem). Ref. [81] interprets then the existence of large plateaus when the expectation value $\langle L^p \rangle$ ("the LIOM density"), in the ground-state of H_{Shifted}, is measured as a function of λ_p ("the LIOM chemical potential"), as a proof that this specific LIOM is a good quantum number (and thus the indication of a MBL phase). In our view, this method is interesting but has not yet been thoroughly tested, and in particular compared to exact results. An important question remains on the signs of the coefficients of the MBL wave-functions (the possibility of doing QMC implies that these coefficients have all the same sign in the chosen basis).

In another strand, Ref. [82] exploits the *Eigenstate-to-Hamiltonian* [83, 84] construction in a QMC framework. The idea is to start with a parent Hamiltonian H for which the ground-state is non-ergodic / localized, but is amenable to stochastic simulations. Specific correlation functions in the ground-state can be sampled with QMC, and are used to reconstruct the coefficients of a target Hamiltonian $\tilde{H}$ for which this state is now (approximately) an excited eigenstate. In this way, high-energy "MBL" eigenstates can be constructed, even in two dimensions, at the price of having a different disorder distribution (usually with strong correlations)

Unitary flow methods — Last but not least, let us finish with a family of elegant methods, which can all be cast as continuous unitary flow techniques. The lecture by S. Thomson in this school gave a complete presentation of the method, but let us recap the main concepts. The idea of this method is to iteratively perform unitary transforms (usually on a local scale) that starts (at ficticious time $\tau = 0$) from the original microscopic Hamiltonian, and aims at flowing towards a basis where ultimately at $\tau = \infty$ it takes a diagonal form (hence the problem is solved). This idea has a long history in quantum physics under the name of continuous unitary transform, Wegner of Wegner-Wilson flows (see [85] for an introduction and Ref. [86] for a review in light of the MBL problem). The generator of the unitary transform can be continuous or discrete and there is a large variety of possible choices for this flow. A key insight is to parametrize the flow precisely in terms of LIOMs operators which we know are well defined in the deep MBL limit. The numerical treatment of such flow methods imply solving a set of coupled differential equations (with respect to τ), which turns to be the method bottleneck. We refer to Refs. [87–91] for a detailed account of these techniques and results on MBL physics, including an estimate of the ETH/MBL transition in 1d [89]. We highlight in particular Ref. [90] which furthermore shows how to capture time evolution within the flow method. Even though it is likely that this family of methods is efficient only in a MBL phase, one underexploited advantage is that there are clear prospects in extending this approach to higher dimensions (e.g. probing MBL in 2d or 3d) or to other problems (driven, open systems).

4.2 Conclusion : Open problems, challenges

This section will be improved too. Your comments are welcome ...

Non-exhaustive list of challenges and open problems for MBL numerics

- Localization Landscape : Is there a many-body localization landscape ? If yes, is it as useful for numerics of the MBL problem as in the Anderson localization case? Given the success of the localization landscape for Anderson problems (see Refs. [92, 93]), it is quite strange that these questions have been seldom addressed in the many-body case (see however Ref. [94] for recent numerical work and Ref. [95] for analytical arguments).

- Dimension larger than 1 : Some first numerical attempts have been performed (a complete reference list soon), but there is certainly more to do.
- A new computational paradigm geared towards the critical ETH-MBL point: there are good paradigms for thermalizing systems (typicality) or full-MBL systems (LIOMs), but we are missing one for the transition point (if any)
- Quasiperiodic versus random potential: different universality classes ?
- Getting closer to the experiments: mixture of two species, bath ...

Acknowledgements

I would like to gratefully acknowledge Nicolas Macé for writing Sec. 2.5.3 of these lectures notes on simulations of quantum circuits. Many thanks to David Luitz and Nicolas Laflorencie for many discussions and collaborations on these topics.

Funding information This work benefited from the support of the project THERMOLOC ANR-16-CE30-0023-02 of the French National Research Agency (ANR) and the French Programme Investissements d'Avenir under the program ANR-11-IDEX-0002-02, reference ANR-10- LABX-0037-NEXT.

References

[1] D. A. Abanin, E. Altman, I. Bloch and M. Serbyn, *Colloquium: Many-body localization, thermalization, and entanglement*, Rev. Mod. Phys. **91**, 021001 (2019), doi:10.1103/RevModPhys.91.021001.

[2] F. Alet and N. Laflorencie, *Many-body localization: An introduction and selected topics*, Comptes Rendus Physique **19**(6), 498 (2018), doi:https://doi.org/10.1016/j.crhy.2018.03.003.

[3] D. A. Abanin and Z. Papić, *Recent progress in many-body localization*, Annalen der Physik **529**(7), 1700169 (2017), doi:10.1002/andp.201700169.

[4] D. J. Luitz and Y. B. Lev, *The ergodic side of the many-body localization transition*, Annalen der Physik **529**(7), 1600350 (2017), doi:10.1002/andp.201600350, https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.201600350.

[5] E. Altman and R. Vosk, *Universal Dynamics and Renormalization in Many-Body-Localized Systems*, Ann. Rev. Cond. Matt. **6**, 383 (2015), doi:10.1146/annurev-conmatphys-031214-014701.

[6] R. Nandkishore and D. A. Huse, *Many-Body Localization and Thermalization in Quantum Statistical Mechanics*, Ann. Rev. Cond. Matt. **6**, 15 (2015), doi:10.1146/annurev-conmatphys-031214-014726.

[7] A. C. Potter and R. Vasseur, *Symmetry constraints on many-body localization*, Phys. Rev. B **94**, 224206 (2016), doi:10.1103/PhysRevB.94.224206.

[8] H. Q. Lin, *Exact diagonalization of quantum-spin models*, Phys. Rev. B **42**, 6561 (1990), doi:10.1103/PhysRevB.42.6561.

[9] T. Uno, *Algorithms for enumerating all perfect, maximum and maximal matchings in bipartite graphs*, Algorithms and Computation pp. 92–101 (1997).

[10] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. S. Zibrov, M. Endres, M. Greiner, V. Vuletić and M. D. Lukin, *Probing many-body dynamics on a 51-atom quantum simulator*, Nature **551**, 579 (2017).

[11] F. Pietracaprina, N. Macé, D. J. Luitz and F. Alet, *Shift-invert diagonalization of large many-body localizing spin chains*, SciPost Physics **5**(5), 045 (2018), doi:10.21468/SciPostPhys.5.5.045.

[12] P. Sierant, M. Lewenstein and J. Zakrzewski, *Polynomially filtered exact diagonalization approach to many-body localization*, Phys. Rev. Lett. **125**, 156601 (2020), doi:10.1103/PhysRevLett.125.156601.

[13] D. J. Luitz, *Polynomial filter diagonalization of large Floquet unitary operators*, SciPost Phys. **11**, 021 (2021), doi:10.21468/SciPostPhys.11.2.021.

[14] J. M. G. Gómez, R. A. Molina, A. Relaño and J. Retamosa, *Misleading signatures of quantum chaos*, Phys. Rev. E **66**, 036209 (2002), doi:10.1103/PhysRevE.66.036209.

[15] Y. Y. Atas, E. Bogomolny, O. Giraud and G. Roux, *Distribution of the ratio of consecutive level spacings in random matrix ensembles*, Phys. Rev. Lett. **110**, 084101 (2013), doi:10.1103/PhysRevLett.110.084101.

[16] O. Giraud, N. Macé, E. Vernier and F. Alet, *Probing symmetries of quantum many-body systems through gap ratio statistics*, Phys. Rev. X **12**, 011006 (2022), doi:10.1103/PhysRevX.12.011006.

[17] Z. Papić, E. M. Stoudenmire and D. A. Abanin, *Many-body localization in disorder-free systems: The importance of finite-size constraints*, Annals of Physics **362**, 714 (2015), doi:10.1016/j.aop.2015.08.024.

[18] D. J. Luitz, *Long tail distributions near the many body localization transition*, Physical Review B **93**(13) (2016), doi:10.1103/PhysRevB.93.134201.

[19] V. Khemani, D. N. Sheng and D. A. Huse, *Two universality classes for the many-body localization transition*, Phys. Rev. Lett. **119**, 075702 (2017), doi:10.1103/PhysRevLett.119.075702.

[20] A. Al-Mohy and N. Higham, *Computing the action of the matrix exponential, with an application to exponential integrators*, SIAM Journal on Scientific Computing **33**(2), 488 (2011), doi:10.1137/100788860, https://doi.org/10.1137/100788860.

[21] H. Tal-Ezer and R. Kosloff, *An accurate and efficient scheme for propagating the time dependent schrödinger equation*, The Journal of Chemical Physics **81**(9), 3967 (1984), doi:10.1063/1.448136, https://doi.org/10.1063/1.448136.

[22] V. V. Dobrovitski and H. A. De Raedt, *Efficient scheme for numerical simulations of the spin-bath decoherence*, Phys. Rev. E **67**, 056702 (2003), doi:10.1103/PhysRevE.67.056702.

[23] Y. Saad, *Analysis of some krylov subspace approximations to the matrix exponential operator*, SIAM Journal on Numerical Analysis **29**(1), 209 (1992), doi:10.1137/0729014, https://doi.org/10.1137/0729014.

[24] M. Smelyanskiy, N. P. Sawaya and A. Aspuru-Guzik, *qhipster: The quantum high performance software testing environment*, arXiv:1601.07195 (2016).

[25] T. Häner and D. S. Steiger, *0.5 petabyte simulation of a 45-qubit quantum circuit*, Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis pp. 33:1–33:10 (2017), doi:10.1145/3126908.3126947.

[26] H. D. Raedt, F. Jin, D. Willsch, M. Willsch, N. Yoshioka, N. Ito, S. Yuan and K. Michielsen, *Massively parallel quantum computer simulator, eleven years later*, Computer Physics Communications **237**, 47 (2019), doi:https://doi.org/10.1016/j.cpc.2018.11.005.

[27] R. S. Bennink, E. M. Ferragut, T. S. Humble, J. A. Laska, J. J. Nutaro, M. G. Pleszkoch and R. C. Pooser, *Unbiased simulation of near-clifford quantum circuits*, Phys. Rev. A **95**, 062337 (2017), doi:10.1103/PhysRevA.95.062337.

[28] G. Vidal, *Efficient classical simulation of slightly entangled quantum computations*, Phys. Rev. Lett. **91**, 147902 (2003), doi:10.1103/PhysRevLett.91.147902.

[29] I. Markov and Y. Shi, *Simulating quantum computation by contracting tensor networks*, SIAM Journal on Computing **38**(3), 963 (2008), doi:10.1137/050644756, https://doi.org/10.1137/050644756.

[30] S. Boixo, S. V. Isakov, V. N. Smelyanskiy and H. Neven, *Simulation of low-depth quantum circuits as complex undirected graphical models*, arXiv:1712.05384 (2017).

[31] B. Villalonga, D. Lyakh, S. Boixo, H. Neven, T. S. Humble, R. Biswas, E. G. Rieffel, A. Ho and S. Mandrà, *Establishing the quantum supremacy frontier with a 281 pflop/s simulation*, arXiv:1905.00444 (2019).

[32] J. Hauschild and F. Pollmann, *Efficient numerical simulations with Tensor Networks: Tensor Network Python (TeNPy)*, SciPost Phys. Lect. Notes p. 5 (2018), doi:10.21468/SciPostPhysLectNotes.5.

[33] S. Paeckel, T. Köhler, A. Swoboda, S. R. Manmana, U. Schollwöck and C. Hubig, *Time-evolution methods for matrix-product states*, Annals of Physics **411**, 167998 (2019), doi:https://doi.org/10.1016/j.aop.2019.167998.

[34] T. L. M. Lezama, S. Bera and J. H. Bardarson, *Apparent slow dynamics in the ergodic phase of a driven many-body localized system without extensive conserved quantities*, Phys. Rev. B **99**, 161106 (2019), doi:10.1103/PhysRevB.99.161106.

[35] D. J. Luitz, Y. B. Lev and A. Lazarides, *Absence of dynamical localization in interacting driven systems*, SciPost Phys. **3**, 029 (2017), doi:10.21468/SciPostPhys.3.4.029.

[36] U. Schollwöck, *The density-matrix renormalization group in the age of matrix product states*, Annals of Physics **326**(1), 96 (2011), doi:https://doi.org/10.1016/j.aop.2010.09.012, January 2011 Special Issue.

[37] G. Vidal, *Efficient simulation of one-dimensional quantum many-body systems*, Phys. Rev. Lett. **93**, 040502 (2004), doi:10.1103/PhysRevLett.93.040502.

[38] P. Sierant and J. Zakrzewski, *Many-body localization of bosons in optical lattices*, New Journal of Physics **20**(4), 043032 (2018), doi:10.1088/1367-2630/aabb17.

[39] T. Chanda, P. Sierant and J. Zakrzewski, *Many-body localization transition in large quantum spin chains: The mobility edge*, Phys. Rev. Research **2**, 032045 (2020), doi:10.1103/PhysRevResearch.2.032045.

[40] D. J. Luitz, N. Laflorencie and F. Alet, *Extended slow dynamical regime close to the many-body localization transition*, Phys. Rev. B **93**, 060201 (2016), doi:10.1103/PhysRevB.93.060201.

[41] E. M. Stoudenmire and S. R. White, *Minimally entangled typical thermal state algorithms*, New Journal of Physics **12**(5), 055026 (2010), doi:10.1088/1367-2630/12/5/055026.

[42] M. P. Zaletel, R. S. K. Mong, C. Karrasch, J. E. Moore and F. Pollmann, *Time-evolving a matrix product state with long-ranged interactions*, Phys. Rev. B **91**(16), 165112 (2015), doi:10.1103/PhysRevB.91.165112.

[43] K. Bidzhiev and G. Misguich, *Out-of-equilibrium dynamics in a quantum impurity model: numerics for particle transport and entanglement entropy*, Physical Review B **96**(19) (2017), doi:10.1103/PhysRevB.96.195117.

[44] J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pižorn, H. Verschelde and F. Verstraete, *Time-dependent variational principle for quantum lattices*, Phys. Rev. Lett. **107**, 070601 (2011), doi:10.1103/PhysRevLett.107.070601.

[45] J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken and F. Verstraete, *Unifying time evolution and optimization with matrix product states*, Phys. Rev. B **94**, 165116 (2016), doi:10.1103/PhysRevB.94.165116.

[46] H. Weimer, A. Kshetrimayum and R. Orús, *Simulation methods for open quantum many-body systems*, arXiv:1907.07079 (2019).

[47] H. P. Lüschen, P. Bordia, S. S. Hodgman, M. Schreiber, S. Sarkar, A. J. Daley, M. H. Fischer, E. Altman, I. Bloch and U. Schneider, *Signatures of many-body localization in a controlled open quantum system*, Phys. Rev. X **7**, 011034 (2017), doi:10.1103/PhysRevX.7.011034.

[48] E. van Nieuwenburg, J. Y. Malo, A. Daley and M. Fischer, *Dynamics of many-body localization in the presence of particle loss*, Quantum Science and Technology **3**(1), 01LT02 (2017), doi:10.1088/2058-9565/aa9a02.

[49] M. Žnidarič, A. Scardicchio and V. K. Varma, *Diffusive and subdiffusive spin transport in the ergodic phase of a many-body localizable system*, Phys. Rev. Lett. **117**, 040601 (2016), doi:10.1103/PhysRevLett.117.040601.

[50] M. Žnidarič, J. J. Mendoza-Arenas, S. R. Clark and J. Goold, *Dephasing enhanced spin transport in the ergodic phase of a many-body localizable system*, Annalen der Physik **529**(7), 1600298 (2017), doi:10.1002/andp.201600298, https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.201600298.

[51] V. K. Varma and M. Žnidarič, *Diffusive transport in a quasiperiodic fibonacci chain: Absence of many-body localization at weak interactions*, Phys. Rev. B **100**, 085105 (2019), doi:10.1103/PhysRevB.100.085105.

[52] T. Prosen and M. Žnidarič, *Matrix product simulations of non-equilibrium steady states of quantum spin chains*, Journal of Statistical Mechanics: Theory and Experiment **2009**(02), P02035 (2009), doi:10.1088/1742-5468/2009/02/p02035.

[53] M. Žnidarič, *Dephasing-induced diffusive transport in the anisotropic heisenberg model*, New Journal of Physics **12**(4), 043001 (2010), doi:10.1088/1367-2630/12/4/043001.

[54] A. J. Daley, *Quantum trajectories and open many-body quantum systems*, Advances in Physics **63**(2), 77 (2014), doi:10.1080/00018732.2014.933502, https://doi.org/10.1080/00018732.2014.933502.

[55] S. P. Lim and D. N. Sheng, *Many-body localization and transition by density matrix renormalization group and exact diagonalization studies*, Phys. Rev. B **94**, 045111 (2016), doi:10.1103/PhysRevB.94.045111.

[56] D. M. Kennes and C. Karrasch, *Entanglement scaling of excited states in large one-dimensional many-body localized systems*, Phys. Rev. B **93**, 245129 (2016), doi:10.1103/PhysRevB.93.245129.

[57] X. Yu, D. Pekker and B. K. Clark, *Finding matrix product state representations of highly excited eigenstates of many-body localized hamiltonians*, Phys. Rev. Lett. **118**, 017201 (2017), doi:10.1103/PhysRevLett.118.017201.

[58] M. Serbyn, A. A. Michailidis, D. A. Abanin and Z. Papić, *Power-law entanglement spectrum in many-body localized phases*, Phys. Rev. Lett. **117**, 160601 (2016), doi:10.1103/PhysRevLett.117.160601.

[59] V. Khemani, F. Pollmann and S. L. Sondhi, *Obtaining highly excited eigenstates of many-body localized hamiltonians by the density matrix renormalization group approach*, Phys. Rev. Lett. **116**, 247204 (2016), doi:10.1103/PhysRevLett.116.247204.

[60] D. Trithep, K. Vedika, P. Frank, H. D. A. and S. S. L., *Obtaining highly excited eigenstates of the localized xx chain via dmrg-x*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **375**(2108), 20160431 (2017), doi:10.1098/rsta.2016.0431.

[61] D. Pekker and B. K. Clark, *Encoding the structure of many-body localization with matrix product operators*, Phys. Rev. B **95**, 035116 (2017), doi:10.1103/PhysRevB.95.035116.

[62] A. Chandran, J. Carrasquilla, I. H. Kim, D. A. Abanin and G. Vidal, *Spectral tensor networks for many-body localization*, Phys. Rev. B **92**, 024201 (2015), doi:10.1103/PhysRevB.92.024201.

[63] C. Zhang, F. Pollmann, S. L. Sondhi and R. Moessner, *Density-matrix renormalization group study of many-body localization in floquet eigenstates*, Annalen der Physik **529**(7), 1600294 (2017), doi:10.1002/andp.201600294, https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.201600294.

[64] F. Pollmann, V. Khemani, J. I. Cirac and S. L. Sondhi, *Efficient variational diagonalization of fully many-body localized hamiltonians*, Phys. Rev. B **94**, 041116 (2016), doi:10.1103/PhysRevB.94.041116.

[65] T. B. Wahl, A. Pal and S. H. Simon, *Efficient representation of fully many-body localized systems using tensor networks*, Phys. Rev. X **7**, 021018 (2017), doi:10.1103/PhysRevX.7.021018.

[66] T. B. Wahl, A. Pal and S. H. Simon, *Signatures of the many-body localized regime in two dimensions*, Nature Physics **15**(2), 164 (2019), doi:10.1038/s41567-018-0339-x.

[67] D. Pekker, G. Refael, E. Altman, E. Demler and V. Oganesyan, *Hilbert-glass transition: New universality of temperature-tuned many-body dynamical quantum criticality*, Phys. Rev. X **4**, 011052 (2014), doi:10.1103/PhysRevX.4.011052.

[68] R. Vasseur, A. C. Potter and S. A. Parameswaran, *Quantum criticality of hot random spin chains*, Phys. Rev. Lett. **114**, 217201 (2015), doi:10.1103/PhysRevLett.114.217201.

[69] R. Vosk, D. A. Huse and E. Altman, *Theory of the many-body localization transition in one-dimensional systems*, Phys. Rev. X **5**, 031032 (2015), doi:10.1103/PhysRevX.5.031032.

[70] A. C. Potter, R. Vasseur and S. A. Parameswaran, *Universal properties of many-body delocalization transitions*, Phys. Rev. X **5**, 031033 (2015), doi:10.1103/PhysRevX.5.031033.

[71] S.-X. Zhang and H. Yao, *Universal properties of many-body localization transitions in quasiperiodic systems*, Phys. Rev. Lett. **121**, 206601 (2018), doi:10.1103/PhysRevLett.121.206601.

[72] F. Iglói and C. Monthus, *Strong disorder RG approach – a short review of recent developments*, Eur. Phys. J. B **91**(11), 290 (2018), doi:10.1140/epjb/e2018-90434-8.

[73] T. Devakul and R. R. P. Singh, *Early breakdown of area-law entanglement at the many-body delocalization transition*, Phys. Rev. Lett. **115**, 187201 (2015), doi:10.1103/PhysRevLett.115.187201.

[74] A. K. Kulshreshtha, A. Pal, T. B. Wahl and S. H. Simon, *Approximating observables on eigenstates of large many-body localized systems*, Phys. Rev. B **99**, 104201 (2019), doi:10.1103/PhysRevB.99.104201.

[75] P. Prelovšek, O. S. Barišić and M. Mierzejewski, *Reduced-basis approach to many-body localization*, Phys. Rev. B **97**, 035104 (2018), doi:10.1103/PhysRevB.97.035104.

[76] F. Pietracaprina and N. Laflorencie, *Hilbert Space Fragmentation and Many-Body Localization*, arXiv:1906.05709 (2019).

[77] G. De Tomasi, F. Pollmann and M. Heyl, *Efficiently solving the dynamics of many-body localized systems at strong disorder*, Phys. Rev. B **99**, 241114 (2019), doi:10.1103/PhysRevB.99.241114.

[78] E. Leviatan, F. Pollmann, J. H. Bardarson, D. A. Huse and E. Altman, *Quantum thermalization dynamics with Matrix-Product States*, arXiv:1702.08894 (2017).

[79] C. D. White, M. Zaletel, R. S. K. Mong and G. Refael, *Quantum dynamics of thermalizing systems*, Phys. Rev. B **97**, 035127 (2018), doi:10.1103/PhysRevB.97.035127.

[80] A. Lerose, M. Sonner and D. A. Abanin, *Influence matrix approach to many-body floquet dynamics*, Phys. Rev. X **11**, 021040 (2021), doi:10.1103/PhysRevX.11.021040.

[81] S. Inglis and L. Pollet, *Accessing many-body localized states through the generalized gibbs ensemble*, Phys. Rev. Lett. **117**, 120402 (2016), doi:10.1103/PhysRevLett.117.120402.

[82] H.-K. Tang, N. Swain, D. C. W. Foo, B. J. J. Khor, G. Lemarié, F. F. Assaad, S. Adam and P. Sengupta, *Evidence of many-body localization in 2D from quantum Monte Carlo simulation*, arXiv e-prints arXiv:2106.08587 (2021), 2106.08587.

[83] E. Chertkov and B. K. Clark, *Computational inverse method for constructing spaces of quantum models from wave functions*, Phys. Rev. X **8**, 031029 (2018), doi:10.1103/PhysRevX.8.031029.

[84] M. Dupont and N. Laflorencie, *Many-body localization as a large family of localized ground states*, Phys. Rev. B **99**, 020202 (2019), doi:10.1103/PhysRevB.99.020202.

[85] S. Kehrein, *The Flow Equation Approach to Many-Particle Systems*, vol. 217, doi:10.1007/3-540-34068-8 (2006).

[86] C. Monthus, *Flow towards diagonalization for many-body-localization models: adaptation of the toda matrix differential flow to random quantum spin chains*, Journal of Physics A: Mathematical and Theoretical **49**(30), 305002 (2016), doi:10.1088/1751-8113/49/30/305002.

[87] L. Rademaker and M. Ortuño, *Explicit local integrals of motion for the many-body localized state*, Phys. Rev. Lett. **116**, 010404 (2016), doi:10.1103/PhysRevLett.116.010404.

[88] L. Rademaker, M. Ortuño and A. M. Somoza, *Many-body localization from the perspective of integrals of motion*, Annalen der Physik **529**(7), 1600322 (2017), doi:10.1002/andp.201600322, https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.201600322.

[89] D. Pekker, B. K. Clark, V. Oganesyan and G. Refael, *Fixed points of wegner-wilson flows and many-body localization*, Phys. Rev. Lett. **119**, 075701 (2017), doi:10.1103/PhysRevLett.119.075701.

[90] S. J. Thomson and M. Schiró, *Time evolution of many-body localized systems with the flow equation approach*, Phys. Rev. B **97**, 060201 (2018), doi:10.1103/PhysRevB.97.060201.

[91] S. Savitz and G. Refael, *Stable unitary integrators for the numerical implementation of continuous unitary transformations*, Phys. Rev. B **96**, 115129 (2017), doi:10.1103/PhysRevB.96.115129.

[92] M. Filoche and S. Mayboroda, *Universal mechanism for Anderson and weak localization*, Proceedings of the National Academy of Science **109**(37), 14761 (2012), doi:10.1073/pnas.1120432109.

[93] D. N. Arnold, G. David, M. Filoche, D. Jerison and S. Mayboroda, *Computing spectra without solving eigenvalue problems*, SIAM Journal on Scientific Computing **41**(1), B69 (2019), doi:10.1137/17M1156721.

[94] G. A. Hamilton and B. K. Clark, *Analysis of Many-body Localization Landscapes and Fock Space Morphology via Persistent Homology*, arXiv e-prints arXiv:2302.09361 (2023), 2302.09361.

[95] S. Balasubramanian, Y. Liao and V. Galitski, *Many-body localization landscape*, Phys. Rev. B **101**, 014201 (2020), doi:10.1103/PhysRevB.101.014201.