15 Dynamical Mean-Field Theory with Quantum Computing

Thomas Ayral
Eviden Quantum Laboratory
Les Clayes-sous-Bois, France

Contents

E. Pavarini, E. Koch, A. Lichtenstein, and D. Vollhardt (eds.)
Understanding Correlated Materials with DMFT
Modeling and Simulation Vol. 15
Forschungszentrum Jülich, 2025, ISBN 978-3-95806-813-1
http://www.cond-mat.de/events/correl25

1 Introduction: why quantum computers for dynamical mean-field theory?

Quantum computers were originally proposed to solve the challenges met by classical computers to tackle the quantum many-body problem [1]. One famous representative of these problems is the Fermi-Hubbard model [2]

$$H = \sum_{ij,\sigma} t_{ij}\, c^\dagger_{i\sigma} c_{j\sigma} + U \sum_i n_{i\uparrow} n_{i\downarrow} - \mu \sum_i n_i, \tag{1}$$

which describes the competition between a kinetic term with hopping amplitudes t_{ij} that favors delocalized states, and an interaction term with interaction strength U that favors localized states, with extra complexity coming from the possibility to tune the average density using a chemical potential μ. Here, creation (resp. annihilation) operators $c^\dagger_{i\sigma}$ (resp. $c_{i\sigma}$) create electrons of spin $\sigma = \uparrow, \downarrow$ on lattice site i, and $n_{i\sigma} = c^\dagger_{i\sigma} c_{i\sigma}$, $n_i = n_{i\uparrow} + n_{i\downarrow}$. Among other reasons, the purported relation between this model and the physics of high-temperature cuprate superconductors spurred early and prolonged interest into the phase diagram of this model, with phases as diverse as Fermi liquids, Mott insulators, superconductors or charge density insulators.
In particular, one central object to elucidate these phase diagrams is the so-called spectral function $A(\boldsymbol{k}, \omega)$, a quantity that is accessible via, e.g., angle-resolved photoemission experiments: its momentum $\boldsymbol{k}$ and energy ω dependence contains distinctive features of the aforementioned phases. From a computational point of view, the spectral function can be computed from the imaginary part of the retarded Green function $G^{\mathrm{R}}(\boldsymbol{k}, \omega)$ ($A(\boldsymbol{k}, \omega) = -\frac{1}{\pi} \operatorname{Im} G^{\mathrm{R}}(\boldsymbol{k}, \omega)$), itself defined as the space and time Fourier transform of the following sum of correlation functions

$$G^{\mathrm{R}}(i, j; t) = \Theta(t) \left(-i \langle c_i(t) c^\dagger_j \rangle - i \langle c^\dagger_j c_i(t) \rangle \right). \tag{2}$$

Here, the average denotes $\langle \cdots \rangle = \frac{1}{Z} \operatorname{Tr} \left(e^{-\beta H} \cdots \right)$, with the partition function $Z = \operatorname{Tr}(e^{-\beta H})$ and the inverse temperature $\beta = 1/T$. Time-dependence is to be understood in the Heisenberg picture, $c_i(t) = e^{iHt} c_i\, e^{-iHt}$. In the zero-temperature limit, the average becomes $\langle \cdots \rangle = \langle \Psi_0 | \cdots | \Psi_0 \rangle$, with $|\Psi_0\rangle$ the ground state of H. The retarded Green function describes how an electron created at site j ($c^\dagger_j$) in the system (described by a Gibbs state at finite temperature or the ground state at zero temperature) propagates to site i for a time t, where it is annihilated (and likewise for a hole, described by the second term). From this definition, one sees that computing $A(\boldsymbol{k}, \omega)$ implies the ability to describe the time evolution of an electron or a hole in a quantum system prepared in its Gibbs or ground state.

1.1 The difficulties of classical methods... and of quantum computers

Due to the many-body nature of the problem, traditional mean-field theories fail to properly describe these states and their subsequent evolution: for instance, the Hartree-Fock method does not capture Mott insulators because they cannot be described with single Slater determinants. In other words, Hubbard physics is generally not described well by single-particle physics:

correlations (essentially entanglement beyond the trivial entanglement required by the Pauli principle) play an important role. This warrants the use of more sophisticated classical methods. These are either exact, but with a cost exponential in some parameter (like the system size in exact diagonalization or some quantum Monte Carlo methods)—or approximate, and therefore limited to certain regimes (think of tensor networks, which are limited to weakly entangled states).

These limitations (some of which were known when quantum computers were first proposed) make processors with quantum properties—commonly called quantum computers or quantum simulators—ideal candidates for computing the spectral function: if these many-body systems can be engineered or programmed to follow similar dynamics to Hubbard dynamics, the time evolution of the processor will require resources than scale, at least at face value, only linearly with the size of the system and evolution time: to reach, say, larger lattice sizes, one just has to add more "particles" (or quantum bits, as we shall call them), and perform longer time evolutions. . . raising hope for exponential speedups to perform time evolutions. Several quantum algorithms, which we will explain in this lecture, were proposed to exploit this fact.

However, as we shall see, quantum algorithms are subject to strong constraints inherent to their quantum nature. A major constraint, which became obvious with the advent of physical realizations of quantum computers in the last decade, is decoherence, namely unwanted entanglement with the outside environment. Decoherence places hard limitations on the duration (number of operations) of quantum algorithms, which rules out many textbook quantum algorithms if no countermeasures are taken. It can possibly be suppressed with quantum error correction techniques, but these in turn require formidable resources that will remain out of the reach of quantum processors for many years. Worse still, even in the absence of decoherence, the local nature of available operators and measurements, and the projective nature of the latter, also need to be taken into consideration when designing a quantum algorithm, and when comparing it to classical counterparts. Finally, even on a complexity-theoretic level, preparing ground states or low-temperature states of many-body systems is likely hard (that is, exponential) even for perfect quantum computers.

Quantum processors are therefore not to be considered a silver bullet to solve strongly-correlated models like the Hubbard model, but rather as powerful heuristics that could outperform classical heuristics in some difficult regimes whose precise delineations yet need to be determined. . . One should perhaps even consider quantum processors as coprocessors to be used in combination with classical heuristics to reach regimes hitherto inaccessible to either classical or quantum algorithms: in the same way as graphics processing units (GPUs) are now used routinely to speed up some linear algebra operations, quantum processing units (QPUs) could be used to accelerate some well-defined subroutines of an otherwise classical program.

1.2 A method to reduce the complexity of the problem

One very successful classical heuristic for tackling the Hubbard model is dynamical mean-field theory (DMFT, [3]). It maps the Hubbard model to a simpler, yet still many-body problem

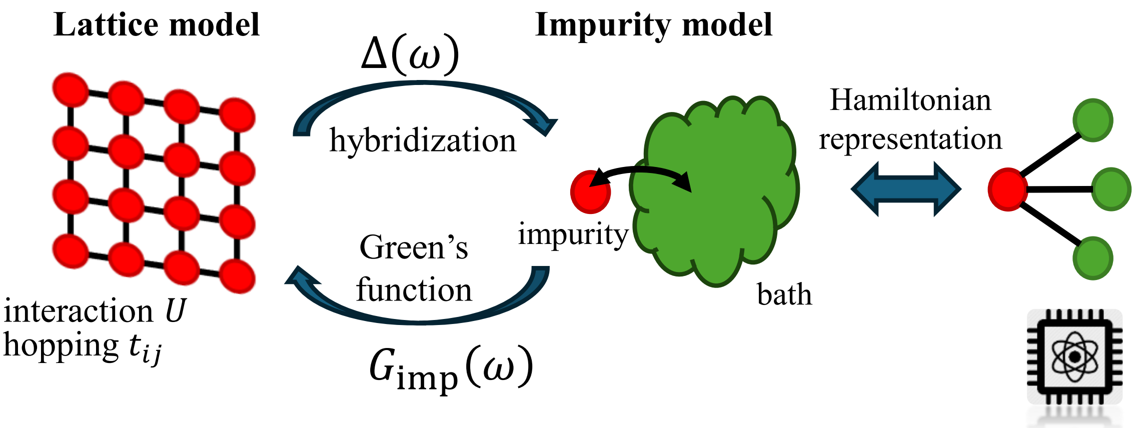

Fig. 1: *DMFT self-consistency cycle: a lattice model (here the Hubbard model) is self-consistently mapped to an impurity model, defined by its hybridization function $\Delta(\omega)$. This impurity model can be represented with an Anderson impurity Hamiltonian describing a correlated impurity site coupled to noninteracting bath sites (right). Quantum processors can be used to solve the impurity model, namely compute its Green function $G_{\text{imp}}(\omega)$.*

called an impurity model. This impurity model describes one (or a few) interacting fermionic sites embedded in a noninteracting environment. The properties of this environment are adjusted to have the Green function of the impurity $G^{\text{R}}_{\text{imp}}(\omega)$ match the local component of the lattice Green function $G^{\text{R}}_{\text{loc}}(\omega) = \sum_{\boldsymbol{k}} G^{\text{R}}(\boldsymbol{k}, \omega)$. This mapping is exact only in the limit of lattices of infinite dimensions thanks to the local nature of the Hubbard interaction, but DMFT is typically used in lower dimensions. Yet, it is able, among other successes, to capture the Fermi-liquid to Mott-insulator transition. Perhaps more importantly, DMFT comes with a control parameter, the number N_c of correlated sites (also called the size of the impurity "cluster") of the impurity model: it becomes exact in the limit of infinite N_c; gradually increasing N_c until convergence of quantitites of interest provides a well-defined way of quantifying errors.

Over the years, very sophisticated methods to solve this correlated, yet simpler impurity model, called "impurity solvers", have been developed. While some of these impurity solvers (think of the numerical renormalization group or segment-picture continuous-time quantum Monte Carlo solvers) yield essentially exact solutions for the $N_c = 1$ case at equilibrium, they face a number of difficulties beyond this regime. For instance, as the number N_c of impurities (or orbitals when going beyond the single-orbital Hubbard model) increases, the exponential wall plaguing classical methods to tackle the Hubbard model makes its comeback. Besides, many of these solvers (like quantum Monte-Carlo methods) work on the imaginary time axis, and thus require uncontrolled analytical continuation techniques to obtain data on the real axis. Conversely, most real-axis techniques (like exact diagonalization) must deal with the large number of bath sites used to represent the noninteracting environment, and the associated exponential complexity. As for tensor network techniques, the time evolution required to obtain Green functions may lead to large entanglement levels and thus make these methods either unreliable or unpractical. All these issues are made worse when turning to out-of-equilibrium regimes, which are beyond the scope of this lecture.

Given these limitations of impurity solvers, quantum computers, now that the problem has been "distilled" down from a large lattice problem to its "quantum quintessence", an impurity model, appear as interesting candidates to reach regimes that are out of the reach of classical impurity solvers: for instance, the regime of large cluster size N_c, or the long-time limit needed to resolve low-energy components of the spectral function. Even though the inherent difficulties of quantum computers are still there, they are made less severe by the reduction to an impurity model.

The goal of this lecture is to explain how the impurity problem of DMFT could be tackled by quantum computers in theory (that is, with perfect or error-corrected quantum computers), but also in practice (taking into account the limitations of current and near-term quantum hardware). To this aim, we will first very briefly recall the DMFT formalism (illustrated in Fig. 1), and then introduce the basic tools of quantum computing. Then, we will describe the textbook quantum algorithms that can be used to solve impurity models, before turning to the practical issues and how other types of algorithms can be used to try and overcome those issues.

2 Dynamical mean-field theory and impurity solvers

In this section, we briefly introduce the main DMFT concepts, and in particular the main computational bottleneck of DMFT, the impurity model. We also explain the main challenges of classical methods.

2.1 Self-consistency equations

The main target of equilibrium DMFT is the lattice retarded Green function, $G^{\mathrm{R}}(\boldsymbol{k},\omega)$ (Eq. (2)). A perturbation expansion of the Hubbard Hamiltonian (Eq. (1)) in powers of the interaction U generates a series expansion around the unperturbed Green function $G_0^{\mathrm{R}}(\boldsymbol{k},\omega)$. This series can in turn be cleverly resummed to yield the following equation, called the Dyson equation

$$G^{\mathrm{R}}(\boldsymbol{k},\omega) = G_0^{\mathrm{R}}(\boldsymbol{k},\omega) + G_0^{\mathrm{R}}(\boldsymbol{k},\omega)\varSigma(\boldsymbol{k},\omega)G^{\mathrm{R}}(\boldsymbol{k},\omega), \tag{3}$$

with a new object, $\varSigma(\boldsymbol{k},\omega)$, called the self-energy, that vanishes when U vanishes. This self-energy captures the effect of interactions on the propagation of electrons. In the limit when the Hubbard model is defined on a lattice with infinite dimensions, the self-energy becomes local, namely independent of $\boldsymbol{k}$. DMFT consists in (i) making the approximation that the self-energy remains local also in finite (and even low) dimensions, and (ii) using a surrogate model, which we will later call impurity model, from which a local self-energy $\varSigma_{\mathrm{imp}}(\omega)$ can be computed as an approximation to the lattice self-energy. This surrogate model also originates from the infinite-dimensional limit of the Hubbard model. It is defined by its action

$$S_{\mathrm{imp}} = \iint_0^\beta \mathrm{d}\tau\mathrm{d}\tau' \sum_\sigma c_\sigma^*(\tau)\big(-\mathcal{G}_0^{-1}(\tau-\tau')\big)c_\sigma(\tau') + \int_0^\beta \mathrm{d}\tau\, U n_\uparrow(\tau)n_\downarrow(\tau)\,. \tag{4}$$

Here, $c_{i\sigma}^*(\tau)$ and $c_{i\sigma}(\tau)$ denote two Grassmann fields. S_{imp} depends entirely on U and on the noninteracting Green function $\mathcal{G}_0$ (here given in imaginary time τ for simplicity; in general,

$f^{\mathrm{R}}(\omega)$ can be recovered, at least formally, from $f(\tau)$ by first Fourier transforming $f(\tau)$ to $\tilde{f}(i\omega_n) = \int_0^\beta \mathrm{d}\tau\, e^{i\tau\omega_n} f(\tau)$ and then performing an analytical continuation $f^{\mathrm{R}}(\omega) = \tilde{f}(z = \omega + i\eta))$. $\mathcal{G}_0^{\mathrm{R}}$, $\Sigma_{\mathrm{imp}}^{\mathrm{R}}$ and the impurity Green function $G_{\mathrm{imp}}^{\mathrm{R}}$ are related by the Dyson equation: $\Sigma_{\mathrm{imp}}^{\mathrm{R}}(\omega) = \mathcal{G}_0^{-1}(\omega) - \left(G_{\mathrm{imp}}^{\mathrm{R}}\right)^{-1}(\omega)$. One often defines the hybridization function $\Delta^{\mathrm{R}}(\omega)$ as $\Delta^{\mathrm{R}}(\omega) = \omega + \mu - \mathcal{G}_0^{-1}(\omega)$. With this, the impurity action reads

$$S_{\mathrm{imp}} = S_{\mathrm{loc}} + S_{\mathrm{hyb}}. \tag{5}$$

Here, $S_{\mathrm{loc}} = \int_0^\beta \mathrm{d}\tau \sum_\sigma c_\sigma^*(\tau)\,(\partial_\tau - \mu)\,c_\sigma(\tau) + \int_0^\beta \mathrm{d}\tau\, U\, n_\uparrow(\tau) n_\downarrow(\tau)$ is easily seen to be the action of a single correlated site, namely a single fermionic site with potential energy $-\mu$ and interaction energy U to penalize double occupancies. The hybridization term in the impurity action, $S_{\mathrm{hyb}} = \iint_0^\beta \mathrm{d}\tau \mathrm{d}\tau' \sum_\sigma c_\sigma^*(\tau) \Delta(\tau - \tau') c_\sigma(\tau')$, describes how this site is coupled to an environment that is completely characterized by the (dynamical) mean-field $\Delta^{\mathrm{R}}(\omega)$. This dynamical mean-field a priori describes an infinite number of degrees of freedom. Since the spectral function is the central object of interest, one adjusts the surrogate model (and hence $\Delta^{\mathrm{R}}(\omega)$) in such a way that its Green function $G_{\mathrm{imp}}^{\mathrm{R}}(\omega)$ coincides with the local component of $G^{\mathrm{R}}(\boldsymbol{k}, \omega)$

$$G_{\mathrm{imp}}^{\mathrm{R}}(\omega)\left[\Delta^{\mathrm{R}}\right] = \sum_{\boldsymbol{k}} G^{\mathrm{R}}(\boldsymbol{k}, \omega)\left[\Delta^{\mathrm{R}}\right]. \tag{6}$$

In this self-consistency equation, we made explicit the functional dependence on Δ^{R}: (i) as the Green function of S_{imp}, $G_{\mathrm{imp}}^{\mathrm{R}}$ is directly a functional of Δ^{R} (see Eq. (9) below); (ii) the dependence of $G^{\mathrm{R}}(\boldsymbol{k}, \omega)$ comes from the Dyson equation (3), which, when performing the DMFT approximation $\Sigma(\boldsymbol{k}, \omega) \approx \Sigma_{\mathrm{imp}}(\omega)$, explicitly reads

$$G^{\mathrm{R}}(\boldsymbol{k}, \omega) = \frac{1}{\omega - \varepsilon(\boldsymbol{k}) + \mu - \Sigma_{\mathrm{imp}}^{\mathrm{R}}(\omega)[\Delta^{\mathrm{R}}]}, \tag{7}$$

with $\varepsilon(\boldsymbol{k})$ the space Fourier transform of the hopping matrix t_{ij} of the Hubbard model (Eq. (1)). Finally, $\Sigma_{\mathrm{imp}}^{\mathrm{R}}(\omega)[\Delta^{\mathrm{R}}] = \omega + \mu - \Delta^{\mathrm{R}} - G_{\mathrm{imp}}^{\mathrm{R}}(\omega)^{-1}\left[\Delta^{\mathrm{R}}\right]$. Putting everything together, we obtain a fixed-point equation

$$G_{\mathrm{imp}}^{\mathrm{R}}(\omega)\left[\Delta^{\mathrm{R}}\right] = \sum_{\boldsymbol{k}} \frac{1}{\Delta^{\mathrm{R}}(\omega) + G_{\mathrm{imp}}^{\mathrm{R}}(\omega)^{-1}\left[\Delta^{\mathrm{R}}\right] - \varepsilon(\boldsymbol{k})}. \tag{8}$$

Solving DMFT amounts to adjusting $\Delta^{\mathrm{R}}(\omega)$ to fulfill the above equation. It is usually solved iteratively. The process crucially hinges on the ability to find a $G_{\mathrm{imp}}^{\mathrm{R}}(\omega)$ for a given $\Delta^{\mathrm{R}}(\omega)$. We will henceforth refer to this task as "solving the impurity model": it is the bottleneck of DMFT. We note that the derivation above can easily be adapted to the case of several correlated atoms (or "impurities") instead of one (this is then called "cluster DMFT"), and to the nonequilibrium case ("out-of-equilibrium DMFT").

Before turning to solving impurity models, let us emphasize that imposing constraints at the level of the (single-particle) Green function is a choice driven by the physical question at stake (here the study of phase transitions with order parameters related to G). Other choices are possible, leading to a whole spectrum of so-called quantum embedding theories (of which DMFT is the earliest representative): one may require self-consistency on “simpler” objects (like one-particle reduced density matrices $\langle c_i^\dagger c_j \rangle$ (same as DMFT but with no time dependence), leading to theories like density-matrix embedding theory (DMET, [4]) or rotationally-invariant slave bosons (RISB, [5, 6])), or on more sophisticated objects (like two-particle Green functions, leading to theories like the dynamical vertex approximation, DΓA [7]). In the former case, one can have the intuition that the corresponding surrogate model will be simpler; conversely, in the latter case, the impurity model should be more complicated [8]).

2.2 Classical impurity solvers. . . and their limitations

In this section, we briefly address the problem of finding the Green function G_{imp} for a given hybridization function $\varDelta$. In a Grassmann path integral formalism, it is given, as a function of imaginary time τ, by the expression

$$G_{\text{imp}}(\tau) = -\int \mathcal{D}\left[c, c^*\right] c_\sigma(\tau) c_\sigma^*(0)\, e^{-S_{\text{imp}}[\varDelta]}\,. \tag{9}$$

(We dropped the σ dependence in G_{imp} to simplify notation, but phases with spin-symmetry breaking can be studied). Impurity solvers can be classified in two main families, Hamiltonian-based solvers and action-based solvers.

2.2.1 Hamiltonian-based solvers

We start with Hamiltonian-based solvers, whose formalism is more directly translatable to quantum algorithms. In Hamiltonian-based solvers, one introduces a Hamiltonian model, dubbed the “Anderson impurity model” (AIM)

$$H_{\text{AIM}} = U\, n_\uparrow n_\downarrow - \mu \sum_\sigma c_\sigma^\dagger c_\sigma + \sum_{k\sigma} \varepsilon_k\, a_{k\sigma}^\dagger a_{k\sigma} + \sum_{k\sigma} V_k \left(a_{k\sigma}^\dagger c_\sigma + \text{h.c.} \right), \tag{10}$$

which describes a single atom (called the impurity) with potential energy $-\mu$ and interaction energy U coupled via hopping terms V_k to a bath of noninteracting fermions with energies ε_k (and creation (resp. annihilation) operators $a_{k\sigma}^\dagger$ (resp. $a_{k\sigma}$)). This model was originally introduced by Anderson to describe isolated impurities in metals [9], hence its name. It is also central to explain so-called Kondo physics (a Schrieffer-Wolff transformation of this model yields its low-energy simplification, the Kondo model). We are now going to show that the hopping V_k and energy ε_k parameters can be adjusted so that the Green function of this model coincides with the impurity Green function (Eq. (9)).

The Green function of the impurity site in H_{AIM} is given by

$$G_{\text{AIM}}(\tau) = -\int \mathcal{D}\left[c, c^*, a_k, a_k^*\right] c_\sigma(\tau) c_\sigma^*(0)\, e^{-S_{\text{AIM}}}, \tag{11}$$

where S_{AIM} is the action corresponding to H_{AIM}. Since S_{AIM} is quadratic in the bath fields $a_k(\tau)$ and $a_k(\tau)^*$, they can be integrated out

$$\int \mathcal{D}\left[a_k, a_k^*\right]\, e^{-S_{\text{AIM}}} = e^{-S_{\text{loc}} - \iint_0^\beta \mathrm{d}\tau \mathrm{d}\tau' \sum_\sigma c_\sigma^*(\tau) \varDelta_{\text{AIM}}(\tau-\tau') c_\sigma(\tau')}$$

with $\varDelta^{\mathrm{R}}_{\mathrm{AIM}}(\omega) = \sum_k V_k^2/(\omega{+}i\eta{-}\varepsilon_k)$. By comparing Eqs (11) and (9), we see that G_{AIM} can be made to coincide with G_{imp} provided $\varDelta^{\mathrm{R}}_{\mathrm{AIM}}(\omega) = \varDelta^{\mathrm{R}}(\omega)$. In other words, if one can find parameters (V_k, ε_k) such that

$$\varDelta^{\mathrm{R}}(\omega) = \sum_k \frac{V_k^2}{\omega + i\eta - \varepsilon_k}, \tag{12}$$

then the AIM's Green function coincides with that of the impurity. This exact correspondence (up to a fit) between our action-based impurity model and a Hamiltonian model makes it possible to tackle the impurity model with a variety of Hamiltonian-based techniques. Before we turn to the main methods and their limitations, let us emphasize that the fitting problem (Eq. (12)) itself is not straightforward: in practice, one needs to perform this fit with a finite number of bath orbitals, possibly leading to fitting errors and finite-size effects. Different choices of metrics to optimize the fitting parameters (V_k, ε_k) will a priori lead to different results.

Hamiltonian-based techniques consist in finding the eigenvectors $|\varPsi_\alpha\rangle$ and eigenvalues E_α of H_{AIM} to compute the Green function. Indeed, one can rewrite, in the operator formalism, $G_{\mathrm{AIM}}(\tau) = -\,\mathrm{Tr}\left[\rho\, c_\sigma(\tau) c^\dagger(0)\right]$, with $\rho = e^{-\beta H_{\mathrm{AIM}}}/Z$ (and $\tau > 0$ here). Inserting completeness relations and expanding the trace, we get $G_{\mathrm{AIM}}(\tau) = -\frac{1}{Z}\sum_{\alpha,\alpha'} e^{-\beta E_\alpha} e^{-\tau(E_{\alpha'}-E_\alpha)} \left|\langle\varPsi_\alpha|c|\varPsi_{\alpha'}\rangle\right|^2$ that becomes, once Fourier-transformed to Matsubara frequencies $i\omega_n$, the so-called Lehmann representation

$$G_{\mathrm{AIM}}(i\omega_n) = -\frac{1}{Z}\sum_{\alpha,\alpha'} \frac{e^{-\beta E_{\alpha'}} + e^{-\beta E_\alpha}}{i\omega_n{-}(E_\alpha - E_{\alpha'})} \left|\langle\varPsi_\alpha|c|\varPsi_{\alpha'}\rangle\right|^2 . \tag{13}$$

In the zero-temperature limit ($\beta \to \infty$)

$$G_{\mathrm{AIM}}(i\omega_n) = -\sum_{\alpha\in\mathcal{H}_{N_e+1}} \frac{1}{i\omega_n - (E_0{-}E_\alpha)} \left|\langle\varPsi_0|c|\varPsi_\alpha\rangle\right|^2 - \sum_{\alpha\in\mathcal{H}_{N_e-1}} \frac{1}{i\omega_n - (E_\alpha{-}E_0)} \left|\langle\varPsi_\alpha|c|\varPsi_0\rangle\right|^2. \tag{14}$$

where we supposed that the ground state $|\varPsi_0\rangle$ contains N_e electrons, and $\mathcal{H}_{N_e\pm1}$ denotes the eigenspace with $N_e \pm 1$ electrons. The spectral function $A_{\mathrm{imp}}(\omega) = -\frac{1}{\pi}\,\mathrm{Im}\, G_{\mathrm{AIM}}(z = \omega{+}i\eta)$ thus has Dirac delta peaks at energies $\omega = E_0{-}E_\alpha$ (with $\alpha \in \mathcal{H}_{N_e+1}$, namely minus the electron addition energy) and $\omega = E_\alpha{-}E_0$ (with $\alpha \in \mathcal{H}_{N_e-1}$, namely the electron removal energy).

The numerical challenge thus consists in finding the eigenvectors $|\varPsi_\alpha\rangle$ and eigenvalues E_α of H_{AIM} (at low temperatures, only the low-lying ones). The sheer size of the Hilbert space ($4^{N_{\mathrm{c}}+N_{\mathrm{b}}}$ if we denote by N_{b} the number of bath sites) makes it a difficult problem, but the methods below either try to exploit the sparsity of H_{AIM} and the fact that only the low-lying states are needed, or use a compressed state representation that works well when entanglement is low.

Exact diagonalization: Lanczos method. A first method, called the Lanczos method (see [10]), avoids the $O\left(\left(4^{N_c+N_b}\right)^3\right)$ cost of a direct diagonalization of H_{AIM} by using the sparsity of H_{AIM}: in the Fock basis representation of H_{AIM} (Eq. (10)), the number of nonzero elements per row or column of H_{AIM} is of the order of $s = O(N_c{+}N_b)$. Thus, matrix-vector multiplications $H_{\mathrm{AIM}}|\varPsi\rangle$ take only $O\left(s4^{N_c+N_b}\right)$ time. This property is exploited in the Lanczos method by finding the matrix of H_{AIM} in the so called Krylov basis $\{|\chi_n\rangle\}_{n=1\ldots K}$ of the K-dimensional vector space spanned by the family $\{|\Phi_0\rangle, H_{\mathrm{AIM}}|\Phi_0\rangle, H_{\mathrm{AIM}}^2|\Phi_0\rangle, \dots, H_{\mathrm{AIM}}^K|\Phi_0\rangle\}$, with $|\Phi_0\rangle$ is an initial state that must not be orthogonal to the exact ground state $|\Psi_0\rangle$. H_{AIM} is tridiagonal in this basis, which makes it easy to compute its ground state energy. The error on the ground state energy obtained in this reduced space decreases exponentially with K [10].

Let us now turn to the Green function. Using Eq. (2) in the case of the AIM and at zero temperature, we obtain

$$G^{\mathrm{R}}(t) = \Theta(t)\Big(-ie^{iE_0t}\langle\Psi_0|c_\sigma e^{-iH_{\mathrm{AIM}}t}c_\sigma^\dagger|\Psi_0\rangle - ie^{-iE_0t}\langle\Psi_0|c_\sigma^\dagger e^{iH_{\mathrm{AIM}}t}c_\sigma|\Psi_0\rangle\Big). \tag{15}$$

Taking the Fourier transform of the first term (the second term, $G^<(\omega)$, can be dealt with in a similar fashion), and inserting resolutions of the identity, we obtain

$$\begin{aligned} G^>(\omega) &= -i\sum_\alpha\int_0^\infty \mathrm{d}t\, e^{i\omega t}\langle\Psi_0|e^{iH_{\mathrm{AIM}}t}c_\sigma\, e^{-iH_{\mathrm{AIM}}t}|\Psi_\alpha\rangle\langle\Psi_\alpha|c_\sigma^\dagger|\Psi_0\rangle & (16)\\ &= -i\sum_\alpha\int_0^\infty \mathrm{d}t\, e^{i(\omega+E_0-E_\alpha)t}\,\big|\langle\Psi_\alpha|c_\sigma^\dagger|\Psi_0\rangle\big|^2. & (17)\end{aligned}$$

Performing the integration and adding a $\eta > 0$ factor to ensure convergence, we find the so-called resolvent form of the Green function

$$G^>(\omega+i\eta) = \sum_\alpha \frac{\big|\langle\Psi_0|c_\sigma|\Psi_\alpha\rangle\big|^2}{E_0-E_\alpha+\omega+i\eta} = \langle\Psi_0|c_\sigma\,\frac{1}{E_0-H+\omega+i\eta}\,c_\sigma^\dagger|\Psi_0\rangle. \tag{18}$$

Having computed $|\Psi_0\rangle$ and E_0 in the first Lanczos step, one can now perform a second Lanczos step, this time with $|\Phi_0\rangle \propto c_\sigma^\dagger|\Psi_0\rangle$. In the corresponding second Krylov basis, taking advantage of the fact that H is tridiagonal in any Krylov basis, one can easily write a continued fraction expression for $G^>(\omega+i\eta)$. One can thus obtain an approximation of the Green function in real frequency with a run time that scales (with a naive estimate) as $O(sK4^{N_c+N_b})$.

Matrix product states: density matrix renormalization group method A second method allows avoiding the exponential cost of storing the wave function by assuming the bipartite entanglement in the system is low. Instead of representing the wavefunction as

$$|\Psi\rangle = \sum_{b_1,\dots b_N}\psi_{b_1,\dots b_N}|b_1,\dots,b_N\rangle, \tag{19}$$

with $N = 2\,(N_c+N_{\mathrm{b}})$, with associated storage cost 2^N, we assume the wavefunction amplitudes to be representable as

$$\psi_{b_1,\dots b_N} = \sum_{\alpha_1,\dots\alpha_{N-1}}\big[A^{(1)}\big]^{b_1}_{\alpha_1}\big[A^{(2)}\big]^{b_2}_{\alpha_1,\alpha_2}\cdots\big[A^{(2)}\big]^{b_{N-1}}_{\alpha_{N-2},\alpha_{N-1}}\big[A^{(1)}\big]^{b_N}_{\alpha_{N-1}}, \tag{20}$$

with N rank-3 tensors $\big[A^{(k)}\big]^{b_k}_{\alpha_{k-1},\alpha_k}$—or matrices if one considers b_k to be fixed, hence the name "matrix product state" (MPS) representation. This is illustrated in Fig. 2(a). Each tensor index has dimension 2, χ, χ for b_k, α_{k-1} and α_k, respectively, with χ called the bond dimension

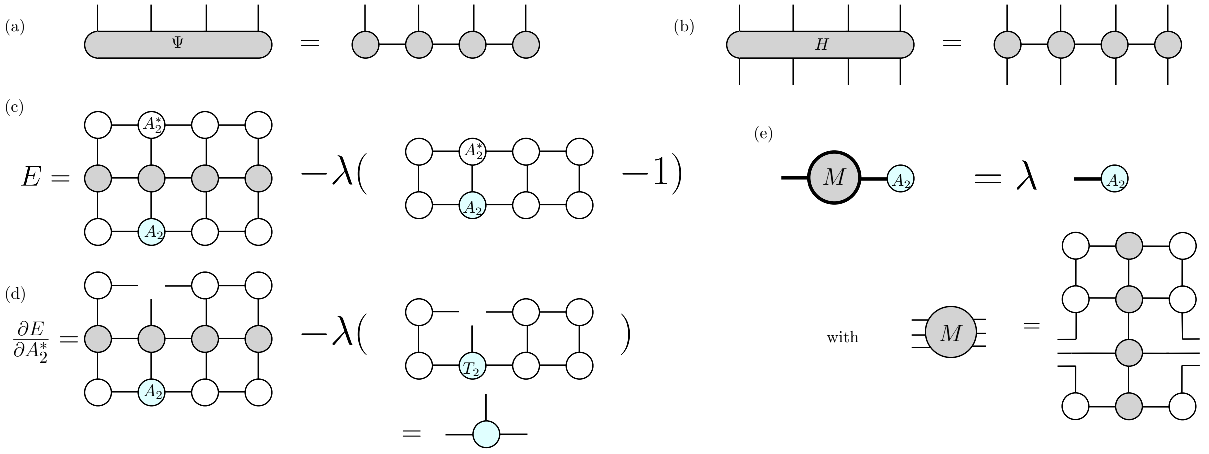

Fig. 2: *Matrix product states and the density matrix renormalization group method. (a) MPS representation of $|\Psi\rangle$. (b) MPO representation of H. (c) Tensor-network representation of the energy functional of DMRG. (d) TN representation of a partial derivative. (e) TN representation of the eigenproblem (thick lines denote groupings of tensor legs).*

of the MPS. The total storage cost thus scales as $O(N\chi^2)$: it is now linear in the number $N \propto N_\mathrm{c}+N_\mathrm{b}$ of sites! However, a fixed bond dimension χ places a limitation on how large the so-called von Neumann entanglement entropy S of the state can be. One can easily show that χ needs to fulfill $\chi \geq 2^S$ ([11]); otherwise, truncation errors appear. It turns out that for Anderson impurity models with $N_\mathrm{c} = 1$, one can always make the problem one-dimensional, in which case known results about the entanglement entropy of ground states of one-dimensional Hamiltonians apply: S follows an "area law", which in 1D means that $S \propto$ const. This allows for a very inexpensive storage of the ground state of the AIM for the $N_\mathrm{c} = 1$ case.

To find the actual MPS form of the ground state, one usually resorts to a variational algorithm known as the density matrix renormalization group algorithm (DMRG), which consists in minimizing $\big\langle \Psi(A^{(1)}, \ldots, A^{(N)})\big|H\big|\Psi(A^{(1)}, \ldots, A^{(N)})\big\rangle$ while imposing normalization, $\big\langle \Psi(A^{(1)}, \ldots, A^{(N)})\big|\Psi(A^{(1)}, \ldots, A^{(N)})\big\rangle = 1$, (Fig. 2(c)). Writing stationarity conditions for a given tensor $A^{(k)}$ (given a fixed value of the other $N{-}1$ tensors) leads to an eigenvalue problem $H^{(k)}_\mathrm{eff}\psi^{(k)} = E^{(k)}\psi^{(k)}$ with $\psi^{(k)}$ of size $2\chi^2$ (Fig. 2 (e)): it can be easily solved exactly. The process is then iterated over $k = 1, \ldots, N$ several times (sweeps) until convergence of the individual tensors. The computational cost thus scales as $N_\mathrm{s}\left(2\chi^2\right)^3$ with N_s the number of sweeps, if one uses a direct diagonalization procedure.

With the ground state in hand, one can now in principle compute the Green function: in Eq. (15), we see that to get the first term, we need to start from $|\Psi_0\rangle$, apply a creation operator to get $c^\dagger_\sigma|\Psi_0\rangle$, time evolve ($e^{-iH_\mathrm{AIM}t}$) and then project back onto $c^\dagger_\sigma|\Psi_0\rangle$. The operation $c^\dagger_\sigma|\Psi_0\rangle$ is straightforward to perform with matrix product states. The time evolution can be performed with a number of methods. The most straightforward one is time-evolving block decimation (TEBD), which consists in splitting the time-evolution operator $e^{-iH_\mathrm{AIM}t}$ in a sequence of so-called Trotter time steps

$$e^{-iH_{\mathrm{AIM}}t} = \left(e^{-iH_{\mathrm{AIM}}t/N_{\mathrm{t}}}\right)^{N_{\mathrm{t}}} \approx \left(1 - iH_{\mathrm{AIM}}\frac{t}{N_{\mathrm{t}}}\right)^{N_{\mathrm{t}}}. \tag{21}$$

Then, the operator $\delta U \equiv 1 - iH_{\mathrm{AIM}}\frac{t}{N_{\mathrm{t}}}$ (represented in so-called matrix product operator, MPO, format) is applied to the current wavevector (represented in MPS format) N_{t} times. This is in principle simple, but the bond dimension χ of the MPS is not conserved upon application of δU: it generically increases, because time evolving the ground state generates excited states that are usually more entangled than the ground state, and thus requires a larger bond dimension to be represented exactly. One faces a dilemma: either ones tries to keep up with the increasing entanglement by increasing χ (but one quickly reaches memory limits), or one truncates the MPS to a fixed maximal bond dimension, thereby incurring truncation errors. More sophisticated methods exists (see [12] for a review), but they generically meet the challenge of an increasing entanglement entropy with evolution time t.

2.2.2 Action-based solvers

The above methods rely on a Hamiltonian representation (Eq. (10)) of the impurity action (5). A major drawback of these methods is that they require a Hamiltonian model with a large bath (large N_{b}) to faithfully represent the hybridization function of DMFT. This large bath, in turn, leads to large numerical costs (unless one chooses a small bath… but then suffers from large finite-size effects).

Action-based solvers avoid this problem by working directly in the action formalism. Starting from Eq. (5), they fall into two main categories, depending on whether they use an expansion in powers of S_{loc} (interaction expansion solvers) or S_{hyb} (hybridization expansion solvers). The resulting infinite sum is sampled using Monte-Carlo methods. The two categories come with distinct properties: intuitively, interaction expansion solvers are better behaved when interactions are weak, while hybridization expansion solvers are better behaved in the strong interaction limit. Both families yield the impurity Green function in imaginary time $G_{\mathrm{imp}}(\tau)$ exactly up to statistical uncertainty. It turns out that this statistical uncertainty (namely the variance of the Monte-Carlo averages used to compute the Green function) generically blows up (exponentially) with decreasing temperature and with increasing cluster size (or number of orbitals) N_{c}, a manifestation of the so-called fermionic sign problem. To curb this variance, one in principle needs very long Markov chains and thus very long run times.

Let us also stress that even in the absence of this sign problem (say in the N_c= 1 case), obtaining the real-time frequency Green function $G^{\mathrm{R}}_{\mathrm{imp}}(\omega)$ from noisy (due to Monte-Carlo statistical noise), imaginary-time $G_{\mathrm{imp}}(\tau)$, a problem called the analytical continuation, is notoriously difficult (it amounts to inverting an ill-conditioned matrix), so that it would be highly desirable to work directly in real time. However, in real time, Monte-Carlo methods struggle with another sign problem stemming from the complex imaginary factors of the time evolution operator ("dynamical sign problem"), strongly limiting these methods to short times, and thus not allowing enough low-frequency accuracy in the resulting spectral functions.

3 Quantum computing tools for impurity models

In the previous section, we saw that DMFT can be seen as a classical heuristic that self-consistently reduces strongly correlated lattice models like the Hubbard model to simpler, yet still strongly-correlated models known as impurity models (Fig. 1). To compute the spectral function of the lattice model within the DMFT approximation, one needs to compute the impurity retarded Green function, namely the Green function of a few (N_c) correlated sites exchanging electrons with a large noninteracting fermionic bath. Essentially, as we saw, the computation of this Green function boils down to computing objects of the form

$$G^{>}(t) = -i\big\langle c_\sigma(t) c_\sigma^\dagger \big\rangle \tag{22}$$

which, together with its counterpart $G^{<}(t) = i\langle c_\sigma^\dagger c_\sigma(t)\rangle$, appear in the definition (2) of the retarded Green function (here for simplicity we dropped the σ dependence of G, and assumed $N_\mathrm{c} = 1$; also, because of the $\Theta(t)$ appearing in (2), we can assume $t > 0$).
The goal of this section is to show how quantum computers can be used to compute objects of form of $G^{>}(t)$ for the specific case of impurity models. If we expand its zero-temperature expression, we recall that we obtain

$$G^{>}(t) = -i\big\langle \Psi_0 | e^{iH_\mathrm{AIM}t}\, c_\sigma\, e^{-iH_\mathrm{AIM}t}\, c_\sigma^\dagger | \Psi_0 \big\rangle. \tag{23}$$

This form makes it clear that we can decompose the computation in two parts: the computation of the ground state $|\Psi_0\rangle$, and the time evolution $e^{-iH_\mathrm{AIM}t}$ of the ground state with one added particle. This is similar to the strategy we described above for matrix product states, but as we shall see, using a quantum processor comes, at least in theory, with added benefits.

3.1 Quantum computing in a nutshell

In this section, we introduce the main tools available to a quantum programmer, with a focus on tools that will be useful for solving impurity models. In particular, a goal of this section is to describe a simple way to perform the operation $e^{-iH_\mathrm{AIM}t}$ on a quantum computer.

3.1.1 Definition of a quantum computer

A quantum computer is a system described by a time-dependent Schrödinger equation

$$i\hbar \frac{\mathrm{d}|\Psi(t)\rangle}{\mathrm{d}t} = H(t)\big|\Psi(t)\big\rangle, \tag{24}$$

and whose initial state $|\Psi(t{=}0)\rangle$ and time-dependent Hamiltonian $H(t)$ can be *controlled* to reach a target final state $|\Psi(t_\mathrm{f})\rangle$, on which observables (hermitian operators) $\hat{O}$ of interest can be measured. This can be done either in a one-shot way (giving access to a bit of information λ with a probability given by the overlap of $|\Psi(t_\mathrm{f})\rangle$ with one of the eigenvectors $|\varphi_\lambda\rangle$ of $\hat{O}$), or in an average way (giving access to a statistical estimate of the average value $\langle\hat{O}\rangle = \big\langle\Psi(t_\mathrm{f})\big|\hat{O}\big|\Psi(t_\mathrm{f})\big\rangle$).

What this system is varies from one quantum computer implementation to the other. A distinction is usually made between analog computers (also known as quantum simulators)—whose Hamiltonian is very specific (e.g with a limited control on the individual degrees of freedom), and digital (or gate-based) computers—which are characterized by the possibility to control individually each degree of freedom, with a practical consequence: this individual (or local) control, complemented with an entangling operation between the degrees of freedom, leads to a form of "universality", namely any unitary operation U can be approximated by the unitary operator corresponding to the full time evolution

$$U(t) = T \exp\Big(-i \int_0^t H(\tau)\, \mathrm{d}\tau\Big). \tag{25}$$

A universal quantum computer can thus be regarded as a machine that implements arbitrary unitary operations. The space on which these operations are performed is, for mainstream quantum computers, the Hilbert space of N two-level systems (aka quantum bits or qubits) $\mathcal{H}_N$. Its size is exponential ($\dim\mathcal{H}_N = 2^N$), and a generic N-qubit state (recall Eq. (19)) a priori requires 2^N complex coefficients to be represented on a classical computer. While simulating such a time evolution on a classical computer would a priori involve resources scaling as $O(2^N)$, letting the quantum computer evolve "naturally" leads to resources linear in the number of degrees of freedom N.

Let us note that quantum computers are not limited to qubit (or spin-1/2) systems. In particular, a natural candidate implementation for dealing with fermionic systems would be a system with fermionic degrees of freedom like ultracold fermionic atoms in optical lattices, whose Hamiltonian is engineered as close as possible to, say, the Hubbard model. These systems are indeed promising platforms for gaining insights, among others, into Hubbard physics, with very recent major improvements in the reachable temperatures. In this lecture, however, we will focus only on qubit-based computers, and within this category, on gate-based quantum computers. As we shall see, using these computers to tackle fermionic systems will require a translation from the world of fermions to the world of qubits; but the exquisite control afforded by this type of computers makes them very convenient to design algorithms.

3.1.2 The circuit model

One usually picks, as a basis of the Hilbert space $\mathcal{H}_n$, the tensor basis of the eigenstates of Pauli z matrix σ_z: it is denoted as $|b_1, \ldots, b_N\rangle$, with $b_k \in \{0, 1\}$ and $\sigma_z^{(k)}|b_k\rangle = (-)^{b_k}|b_k\rangle$. It is usually called the computational basis. In the following, $\sigma_\alpha^{(k)}$ will denote a Pauli matrix ($\alpha = x, y, z$) acting on the k-th qubit.

A quantum computation consists in modifying the state $|\Psi\rangle$ by cleverly designing the time dependence of the Hamiltonian. In gate-based quantum computers, the time evolution $U \equiv U(t)$ (Eq. (25), we drop the t dependence for clarity) is split into a sequence

$$U = \prod_{k=1}^{N_g} U_k \tag{26}$$

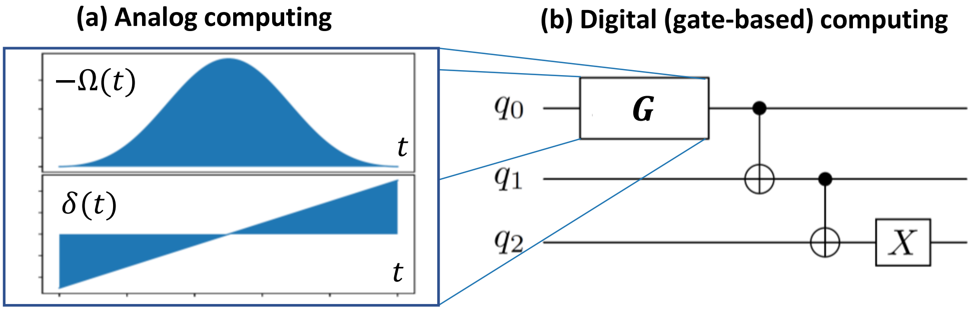

Fig. 3: *Circuit model: a quantum circuit (on the right) describes a sequence of gates, which are individually implemented in an analog fashion with time-dependent terms in the hardware Hamiltonian. Reproduced from [13].*

of N_g elementary unitary operations U_k, referred to as quantum "gates", that correspond to switching-on some terms in the Hamiltonian. For instance, in Fig. 3, the G gate, corresponding to a unitary U_G, may correspond to a Hamiltonian $H_G(t) = \Omega(t)\sigma_x^{(1)}/2 - \delta(t)\,\sigma_z^{(1)}$, with U_G given by the Dyson series $U_G = Te^{-i\int_0^t H_G(\tau)\,d\tau}$. Since H_G acts only on the first qubit, so does U_G. Such a single-qubit gate (depicted by a box acting only on one line, namely one qubit) does not create entanglement (acting with U_G on a factorized state yields a factorized state). A standard single-qubit gate is the Hadamard gate, whose expression in the $\{|0\rangle, |1\rangle\}$ basis reads

$$U_{\mathrm{H}} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{27}$$

In other words, $U_{\mathrm{H}}|0\rangle = \big(|0\rangle + |1\rangle\big)/\sqrt{2}$: the Hadamard gate creates superpositions. Other important standard single-qubit gates are single-qubit rotations, defined as

$$R_\alpha(\theta) = e^{-i\frac{\theta}{2}\sigma_\alpha}. \tag{28}$$

They rotate the qubit around the axis α by an angle θ.

Gates depicted as boxes spanning several lines, on the other hand, are many-qubit gates. They require Hamiltonians that couple several qubits (like, say, $H(t) = g(t)\sigma_x^{(1)} \otimes \sigma_x^{(2)}$, which generates a gate $U_{\mathrm{XX}}(t) = \exp\big(-i\int_0^t \mathrm{d}\tau\, g(\tau)\sigma_x^{(1)} \otimes \sigma_x^{(2)}\big)$). The specific two-qubit gate shown in Fig. 3 after gate G is a so-called CNOT gate. Its expression in the $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ basis reads

$$U_{\mathrm{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{29}$$

For instance, $U_{\mathrm{CNOT}}|10\rangle = |11\rangle$: if the first ("control") qubit is in state $|1\rangle$, the second ("target") qubit is flipped. This notion of controlled gate can be generalized: a controlled-U gate U_c is such that $U_c|0\rangle|\psi\rangle = |0\rangle|\psi\rangle$ and $U_c|1\rangle|\psi\rangle = |1\rangle U|\psi\rangle$.

Let us stress that several Hamiltonians $H(t)$ can yield the same unitary evolution (for instance several functions $g(t)$ generate the same U_{XX} gate in the example above, provided the integral

$\int_0^t g(\tau)\,\mathrm{d}\tau$ is the same). Thus, going to a circuit-level description of the time evolution instead of staying at the Hamiltonian level allows one to abstract away the implementation details. Depending on the architecture, we may have different $H(t)$ yielding the same unitary evolution. To benefit from this description, we of course need to be able to switch on and off terms of the Hamiltonian acting on selected qubits, a property we called "local control" above, and that analog computers do not possess. Finally, we call "quantum circuit" the graphical representation of the sequence of gates: it describes the unitary operation that is effected by the quantum computer.

3.1.3 Time evolution

In view of our goal of computing the Green function (Eq. (23)), there is a special unitary operation we would like to realize, namely a time evolution (like $U = e^{-iH_{\text{AIM}}t}$ in Eq. (23)).
Many quantum algorithms have been and are being developed to perform this task, also known as the "Hamiltonian simulation problem". With an analog quantum computer (or quantum simulator) that directly implements the Hamiltonian under study (say $H_{\text{hardware}} = H_{\text{AIM}}$), this just consists in letting the system evolve for a time t. For gate-based quantum computers, we need to find a sequence of gates that at least approximately implements e^{-iHt} (the same method applies to the more time-dependent case $Te^{-i\int_0^t H(\tau)\,d\tau}$). The most straightforward algorithm, known as the Trotterization or product formula method, relies on decomposing H as a weighted sum of, e.g., products of Pauli matrices

$$H = \sum_{i=1}^{M} \lambda_i P_i, \tag{30}$$

with $\lambda_i \in \mathbb{R}$ and $P_i = \bigotimes_{k=1}^{N} \sigma_{\alpha_k}^{(k)}$, where here the index α_k runs over $0, x, y, z$, where $\sigma_0 = I$ by convention. Because the individual Pauli terms P_i do not commute with one another, one then performs a similar time slicing (called Trotterization) as the one we encountered in the TEBD algorithm in a tensor network context (see Eq. (21))

$$e^{-iHt} = \left(e^{-iHt/N_t}\right)^{N_t} = \left(\prod_{i=1}^{M} e^{-i\frac{\lambda_i}{N_t}P_i t} + O\left(\frac{t}{N_t}\right)^2\right)^{N_t} = \prod_{m=1}^{N_t}\prod_{i=1}^{M} e^{-i\frac{\lambda_i}{N_t}P_i t} + O\left(\frac{t^2}{N_t}\right). \tag{31}$$

With this slicing, one obtains a sequence of $N_t{\cdot}M$ unitary operators with the form of a Pauli rotation $R_{P_i}(\theta_i) = e^{-i\frac{\theta_i}{2}P_i}$, with $\theta_i = 2\lambda_i t/N_t$.
Let us know show that each $R_{P_i}(\theta_i)$ is easy to implement in terms of a simple quantum circuit. Let us start with a simple case $P_i = \sigma_z^{(1)}\sigma_z^{(2)}$. It is easy to check that the circuit shown in Fig. 4 implements $R_{\sigma_z^{(1)}\sigma_z^{(2)}}(\theta)$. One easily shows, by induction, that $R_{\bigotimes_{m=1}^{K}\sigma_z^{(k_m)}}(\theta)$ is implemented by a circuit with CNOTS on pairs (k_1, k_2), (k_2, k_3), ..., (k_{K-1}, k_K), a rotation $R_z(\theta)$ on the K-th qubit, and then the reversed sequence of CNOTs, as illustrated in Fig. 4(c). Finally, to perform a general Pauli rotation (with Paulis on the x and y axis), one performs single-qubit rotations on the requisite qubits to go back to the z axis. For instance, for $P_i = \sigma_x^{(1)}\sigma_z^{(2)}$, noticing that

$$U_{\text{H}}\sigma_z U_{\text{H}}^{(\dagger)} = \sigma_x, \tag{32}$$

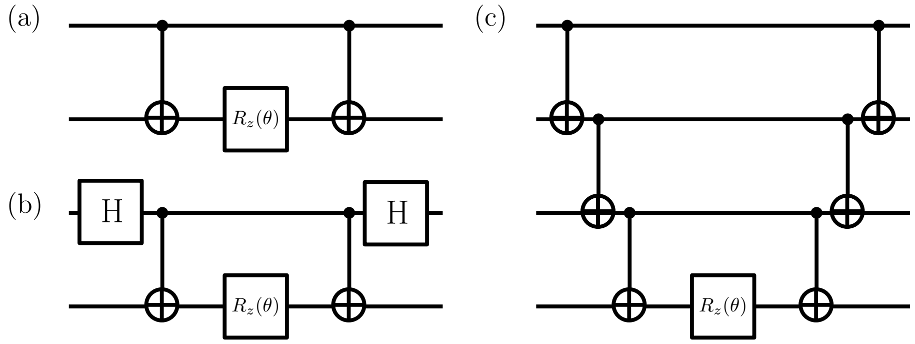

Fig. 4: *Circuits for $R_{P_i}(\theta)$. (a) Circuit for $R_{\sigma_z^{(1)}\sigma_z^{(2)}}(\theta)$. (b) Circuit for $R_{\sigma_x^{(1)}\sigma_z^{(2)}}(\theta)$. (c) Circuit for $R_{\sigma_z^{(1)}\sigma_z^{(2)}\sigma_z^{(3)}\sigma_z^{(4)}}(\theta)$.*

we see that

$$\begin{aligned} U_{\mathrm{H}}^{(1)} R_{\sigma_z^{(1)}\sigma_z^{(2)}}(\theta) U_{\mathrm{H}}^{(1)} &= U_{\mathrm{H}}^{(1)} \big(\cos(\theta) I - i \sin(\theta) \sigma_z^{(1)} \sigma_z^{(2)}\big) U_{\mathrm{H}}^{(1)} \\ &= \cos(\theta) I - i \sin(\theta) \sigma_x^{(1)} \sigma_z^{(2)} \\ &= R_{\sigma_x^{(1)}\sigma_z^{(2)}}(\theta). \end{aligned}$$

This is illustrated in Fig. 4(b).

We thus have a generic, approximate way to perform the time evolution of a Hamiltonian that we have decomposed as a sum of Pauli operators (Eq. (30)). Let us now count the number of operations contained in this circuit: there are N_t Trotter steps, each of which contains M operators of the form $R_{P_i}(\theta)$. The number of gates in each operator depends on the support s_i (number of non-identity Pauli operators) in P_i: we have $O(s_i)$ gates in the corresponding subcircuit. If we call $s = \max_i s_i$, we obtain $N_g = O(N_t M s)$, with a total error $\epsilon = O(t^2/N_t)$. One usually wants to adjust N_t to reach a desired error ϵ: $N_t = O(t^2/\epsilon)$ and thus,

$$N_g = O(M t^2 s/\epsilon). \tag{33}$$

This calls forth several remarks: (i) the number of terms M needs to be "small". A generic Hamiltonian in $\mathcal{H}_N$ will have $M = 2^N$, which is intractable. However, as we shall see, the Hamiltonians we are interested in have a number of terms M that scales polynomially with N (which is considered to be tractable); (ii) the longer the evolution time t, the more gates. Here, the quadratic dependence (instead of the expected linear dependence) comes from the fact we had to slice the time evolution. We can use higher-order Suzuki-Trotter formulas that will lead to a scaling that is closer to linear, but with an overhead in the number of gates per slice (see [14] for a general upper bound on the Trotter error). More advanced methods such as qubitization [15] achieve a linear scaling, but they usually require additional ("ancilla") qubits. They can also achieve a better scaling than $1/\epsilon$; (iii) finally, the support s of the terms in the decomposition of H plays an important role. As we shall see below, the fermionic nature of the Hamiltonian of interest here leads to quite large supports.

3.1.4 Quantum measurements and the variance issue

Before we turn to the specifics of fermionic problems on qubit quantum computers, let us examine one last issue with the quantity we need to compute, Eq. (23). It is expressed as a quantum mechanical average value $\langle\psi|\hat{O}|\psi\rangle$, with operator $\hat{O} = U^{\dagger}c_{\sigma}Uc_{\sigma}^{\dagger}$ and $U = e^{-iH_{\text{AIM}}t}$. We have seen in the previous section how to perform U as a quantum circuit, but what quantum mechanics gives us easy access to is the estimation of $\langle\psi|\hat{A}|\psi\rangle$, where $\hat{A}$ is a Hermitian operator, not any operator like $\hat{O}$.

Before we turn to specific quantum circuits that allow us to get access to $\langle\psi|\hat{O}|\psi\rangle$, let us focus on the Hermitian case. In quantum mechanics, the measurement of an observable $\hat{A}$ with eigenvalues λ and eigenvectors $|\varphi_{\lambda}\rangle$ results (in the nondegenerate case) in a collapse of $|\psi\rangle$ to one of the eigenvectors $|\varphi_{\lambda}\rangle$ with a probability given by Born's rule, $p(\lambda) = |\langle\varphi_{\lambda}|\psi\rangle|^2$, and gives access to the value λ of the corresponding eigenvalue. Repeating the circuit and measurement several (say N_s) times (usually called "shots") allows to compute a statistical estimator $\bar{A}(N_s)$ that converges to the expectation value of $\hat{A}$ in state $|\psi\rangle$ in the large N_s limit

$$\bar{A}(N_s) = \frac{1}{N_s}\sum_{i=1}^{N_s}\lambda_i \xrightarrow[N_s\infty]{} \sum_{\lambda} p(\lambda)\,\lambda = \langle\psi|\hat{A}|\psi\rangle = \langle A\rangle. \tag{34}$$

Quantum computers can either be used in a one-shot way or to compute averages. The one-shot way is particularly useful when the distribution $p(\lambda)$ is peaked around a value λ_0 which one wants to discover: in that case, a single (or a few) shots will yield λ_0. As for the computation of averages, it also works when $p(\lambda)$ is not peaked, but one has to contend with the statistical error (standard error on the mean)

$$\mathit{\Delta}A(N_s) = \sqrt{\left\langle \left(\langle A\rangle - \bar{A}(N_s)\right)^2 \right\rangle} = \sqrt{\frac{\text{Var}(A)}{N_s}}, \tag{35}$$

where the second equality holds because independent experiments (shots) are independent and identically distributed (iid). The variance is, explicitly,

$$\text{Var}(A) = \langle\psi|A^2|\psi\rangle - \left(\langle\psi|A|\psi\rangle\right)^2,$$

so the number of shots needed to attain a fixed accuracy $\mathit{\Delta}A$ depends both on the observable A to be measured and the state $|\psi\rangle$ on which it is measured. For instance, if one is trying to generate the ground state $|\mathit{\Psi}_0\rangle$ of some Hamiltonian H, and wants to measure its energy $\langle H\rangle$, then, because $|\mathit{\Psi}_0\rangle$ is an eigenstate of H, the variance will vanish and only one shot will suffice. This property, called variance reduction, is often exploited in (classical) variational Monte-Carlo algorithms: as the algorithm converges to the ground state, a fixed number of samples (shots) gives an increased accuracy.

However, not every observable $\hat{A}$ can be measured on quantum computers. In fact, in most technologies, only Pauli z matrices $\sigma_z^{(k)}$, for $k = 1, \ldots, N$, and tensor products thereof, can be measured. This is represented by a meter symbol in quantum circuits (see the symbols at the end of the first lines of Fig. 5(a) and (b)). Other Pauli expectation values can be computed by

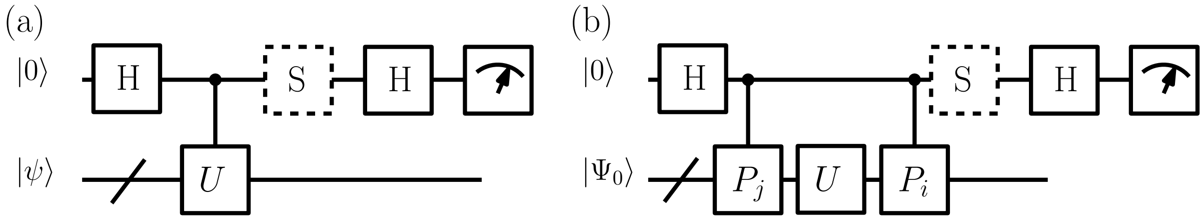

Fig. 5: *Hadamard test: (a) Computation of $\langle\psi|U|\psi\rangle$ by the Hadamard test circuit. The S gate is used only to compute the imaginary part. (b) Application to the computation of $\langle\Psi_0|U^\dagger P_i U P_j|\Psi_0\rangle$ terms.*

rotating the state to the right axis: for instance, to measure $\langle\sigma_x\rangle = \langle\psi|\sigma_x|\psi\rangle$, one can apply a Hadamard gate to get $|\psi'\rangle = U_{\mathrm{H}}|\psi\rangle$ and then, using (32), measure $\langle\sigma_x\rangle' = \langle\psi'|\sigma_x|\psi'\rangle$.

As for generic, multi-qubit observables $\hat{A}$, they can, as we saw earlier for the Hamiltonian, always be decomposed as a weighted sum of products of Pauli matrices

$$\hat{A} = \sum_{i=1}^{M} \lambda_i P_i. \tag{36}$$

One can thus in principle compute $\langle A\rangle$ by computing separately the M terms $\langle P_i\rangle$. For example, to measure $\left\langle\sigma_x^{(1)}\sigma_z^{(2)}\right\rangle$, one first applies a Hadamard gate on the first qubit, and then measures the operator $\sigma_z^{(1)}\otimes\sigma_z^{(2)}$ repeatedly.

We now turn to the problem at hand, namely computing $\langle\psi|\hat{O}|\psi\rangle$ for a generic operator. A generic strategy is to decompose $\hat{O}$ as a sum of unitaries, $\hat{O} = \sum_i \lambda_i \hat{U}_i$ (with $\lambda_i \in \mathbb{C}$), and compute $\langle\psi|\hat{U}_i|\psi\rangle$ separately. (Note that a Pauli decomposition like Eq. (36) is also a decomposition as a linear combination of unitaries since Pauli matrices are unitary).

The circuit shown in Fig. 5(a), dubbed a Hadamard test, does the job of computing $\langle\psi|\hat{U}|\psi\rangle$ for a given unitary U: starting from a state $|0\rangle\otimes|\psi\rangle$, applying the first Hadamard gate yields $\big(|0\rangle+|1\rangle\big)/\sqrt{2}\otimes|\psi\rangle$. Then, the controlled unitary yields $\big(|0\rangle|\psi\rangle+|1\rangle U|\psi\rangle\big)/\sqrt{2}$, and the final Hadamard yields the final state $|\psi_f\rangle = \big(|0\rangle\,(I+U)\,|\psi\rangle + |1\rangle\,(I-U)\,|\psi\rangle\big)/2$. The meter symbol stands for a $\sigma_z^{(0)}$ measurement. The probability of getting outcome 0 is given by Born's rule $p(0) = \langle\psi_f|\,(|0\rangle\langle 0|\otimes I)\,|\psi_f\rangle = \big\|(I+U)|\psi\rangle\big\|^2/4 = \big(1+\mathrm{Re}\langle\psi|U|\psi\rangle\big)/2$, so that

$$\mathrm{Re}\langle\psi|U|\psi\rangle = 2p(0) - 1. \tag{37}$$

If one adds an S gate to the circuit (dashed box in Fig. 5(a), whose matrix is $U_S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$), U changes to iU in the above computation, giving $\mathrm{Im}\langle\psi|U|\psi\rangle = 1-2p(0)$. Thus, provided the ability to implement a controlled-U evolution, we can compute terms of the form $\langle\psi|U|\psi\rangle$, and thus arbitrary values $\langle\psi|\hat{O}|\psi\rangle$ via the linear combination of unitaries.

The last step we have to go through is how to express $\hat{O} = U^\dagger c_\sigma U c_\sigma^\dagger$ as a linear combination of unitaries. For this, we need to go from fermionic operators to qubit ones.

3.1.5 Quantum computing for fermions

In the previous sections, we have learned to perform time evolutions of Hamiltonian operators decomposed as a weighted sum of Pauli operators (Eq. (30)), and to measure quantities of the form $\langle\psi|\hat{O}|\psi\rangle$ given a decomposition of $\hat{O}$ as a weighted sum of unitaries. We now explain how to obtain the precise form of these decompositions in the case where H is a fermionic operator like the AIM, and $\hat{O} = U^\dagger c_\sigma U c_\sigma^\dagger$.

The task consists in finding a mapping between a Fock space with N fermionic orbitals, with a Fock basis $\left\{|n_1,\dots,n_N\rangle = \prod_{k=1}^N \left(c_k^\dagger\right)^{n_k} |0,\dots,0\rangle, n_k \in \{0,1\}\right\}$, and a Hilbert space with N qubits, with a basis $\{|b_1\rangle \otimes |b_2\rangle \otimes \dots \otimes |b_N\rangle, b_k \in \{0,1\}\}$.

The Jordan-Wigner mapping The simplest mapping (also called encoding) between these two bases consists in defining, for all $k = 1,\dots,N$,

$$b_k = n_k. \tag{38}$$

It is called the Jordan-Wigner encoding. Having chosen this correspondence between states, we want to find the qubit operator $\tilde{c}^\dagger$ that acts on the $|b_1,\dots,b_N\rangle$ the same way as $c^\dagger$ acts on the $|n_1,\dots,n_N\rangle$, namely:

$$\begin{aligned} c_k^\dagger|n_1,\dots,0,\dots,n_N\rangle &= (-)^{\sum_{k'=1}^{k-1} n_{k'}} |n_1,\dots,1,\dots,n_N\rangle \\ c_k^\dagger|n_1,\dots,1,\dots,n_N\rangle &= 0 \end{aligned} \tag{39}$$

The operator that turns a two-level system $|0\rangle$ into $|1\rangle$ and vanishes on $|1\rangle$ is $\sigma_+ = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \big(\sigma_x - i\sigma_y\big)/2$. To take into account the $(-)^{\sum_{k'=1}^{k-1} n_{k'}}$ factor, one needs to introduce σ_z operators on qubits $k' < k$, so that

$$\tilde{c}_k^\dagger = \sigma_z^{(1)} \cdots \sigma_z^{(k-1)} \sigma_+^{(k)}. \tag{40}$$

For a spinful fermionic model, we further need to pick an ordering of the (i,σ) index. Two natural choices arise: either $(1,\uparrow),(1,\downarrow),(2,\uparrow),(2,\downarrow),\dots$ or $(1,\uparrow),(2,\uparrow),\dots,(1,\downarrow),(2,\downarrow),\dots$. These choices will yield equivalent, but different quantum circuits. Choosing the second ordering (the one that does not mix different spins), a $c_{i\sigma}^\dagger c_{j\sigma}$ term appearing in the kinetic term of the Hubbard model will transform into

$$\tilde{c}_{i\sigma}^\dagger \tilde{c}_{j\sigma} = \sigma_+^{(i\sigma)} \sigma_z^{(i+1,\sigma)} \cdots \sigma_z^{(j-1,\sigma)} \sigma_-^{(j\sigma)} \tag{41}$$

while Hubbard interaction terms like $n_{i\uparrow}n_{i\downarrow}$ will transform into $\tilde{n}_{i\uparrow}\tilde{n}_{i\downarrow} = \big(I - \sigma_z^{(i\uparrow)}\big)\big(I - \sigma_z^{(i\downarrow)}\big)/4$. In other words, the Anderson Hamiltonian, with terms acting on at most 2 fermionic orbitals at the same time, is turned into a spin Hamiltonian with terms acting on up to $2N_b$ orbitals due to the strings of σ_z operators, called Jordan-Wigner strings, appearing in (41). Explicitly: mapping the impurity orbital ($c_\sigma^\dagger$) to spin index 1 and the k-th bath orbital ($a_{k\sigma}^\dagger$) to spin index $1+k$ we obtain

$$\tilde{H}_{\text{AIM}} = \left(\frac{U}{4} - \frac{\mu}{2} + \sum_{k\sigma} \frac{\varepsilon_k}{2}\right) I - \sum_{k\sigma} \frac{\varepsilon_k}{2} \sigma_z^{(1+k,\sigma)} + \left(\frac{\mu}{2} - \frac{U}{4}\right) \sum_{\sigma} \sigma_z^{(1,\sigma)} + \frac{U}{4} \sigma_z^{(1,\uparrow)} \sigma_z^{(1,\downarrow)}$$
$$+ \sum_{k\sigma} \frac{V_k}{4} \Big(\big(\sigma_x^{(1\sigma)} - i\sigma_y^{(1\sigma)}\big) \sigma_z^{(2,\sigma)} \cdots \sigma_z^{(k,\sigma)} \big(\sigma_x^{(1+k,\sigma)} + i\sigma_y^{(1+k,\sigma)}\big) + \text{h.c.} \Big) \tag{42}$$

We now have a Pauli decomposition of $\tilde{H}_{\text{AIM}}$ with a total of $2(N_b+1)$ 1-qubit terms (disregarding the identity term), one two-qubit term stemming from the Hubbard interaction, and $8N_b$ terms stemming from the hybridization term, with supports ranging from 2 to N_b+1.
Using this decomposition, we can apply the Trotterization method introduced above to construct a quantum circuit that approximates $e^{-iH_{\text{AIM}}t}$. One Trotter step will consist in applying $R_z(\theta)$ rotation gates to all qubits (with angles given by the prefactors in the terms of the first line of expression (42)). As for the second line of (42), it gives rise to four subcircuits of the type $R_{P_i}(\theta)$ that we described in subsection 3.1.3, with circuits of length up to $O(N_b)$, and thus a general scaling of $N_g = O(Mt^2 s/\epsilon) = O\left(N_{\text{b}}^2 t^2/\epsilon\right)$ for the time evolution circuit.
Expression (40) also allows us to express $\hat{O}_\sigma = U^\dagger c_\sigma U c_\sigma^\dagger$ as a linear combination of unitaries: replacing the creation and annihilation operators by (40), we obtain

$$\begin{aligned} 4\hat{O}_\uparrow &= U^\dagger \left(\sigma_x^{(1,\uparrow)} + i\sigma_y^{(1,\uparrow)}\right) U \left(\sigma_x^{(1,\uparrow)} - i\sigma_y^{(1,\uparrow)}\right) \\ &= U^\dagger \sigma_x^{(1,\uparrow)} U \sigma_x^{(1,\uparrow)} - iU^\dagger \sigma_x^{(1,\uparrow)} U \sigma_y^{(1,\uparrow)} + iU^\dagger \sigma_y^{(1,\uparrow)} U \sigma_x^{(1,\uparrow)} + U^\dagger \sigma_y^{(1,\uparrow)} U \sigma_y^{(1,\uparrow)}, \end{aligned} \tag{43}$$

which is in the requisite form.
We thus have achieved our goal. We note that in the single-impurity case that we have picked as an example, we could have "linearized" the bath, namely transformed the current problem, where the impurity is hybridized to every bath site (a so-called "star geometry"), to a problem where the ↑ impurity site is hybridized only to one bath site (itself hybridized to a single other bath site and so on and so forth), and likewise for the ↓ impurity site. This would have allowed for a "chain geometry" where the Jordan-Wigner strings disappear, yielding a scaling of $N_{\text{g}} = O\left(N_{\text{b}} t^2/\epsilon\right)$ gates for the time evolution circuit. However, in the $N_c > 1$ case, we can no longer perform this trick.

Other encodings The Jordan-Wigner encoding is not the only possible fermion-to-spin mapping. While the Jordan-Wigner encoding stores all the information about the orbital occupation in the states (see Eq. (38)) and the information about the parity in the operators (see the string of σ_z operators in (40)), a mirror encoding called the parity encoding does the reverse. For instance, in the parity encoding, $b_k = \sum_{k'=1}^{k} n_k \bmod 2$, namely the states store the parity information, and likewise a string of Pauli operators in the expression for the operators will store the occupation... yielding, as for the Jordan-Wigner transformation, terms with a support $O(N)$ (with N the number of fermionic orbitals) in the Pauli decomposition of the corresponding qubit Hamiltonian. A more sophisticated encoding called the Bravyi-Kitaev encoding [16, 17] mixes the occupation and parity information in a tree structure that allows one to obtain terms with

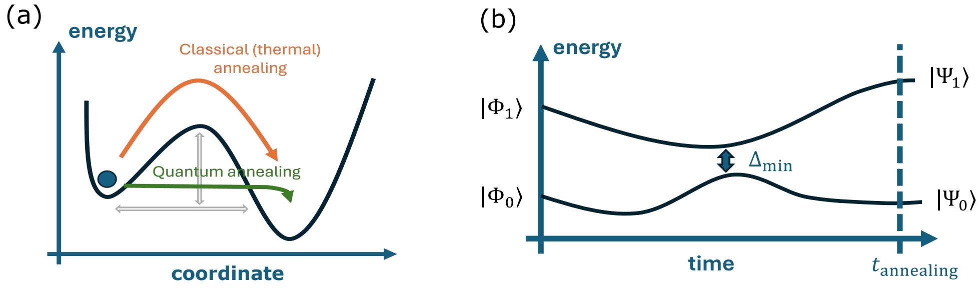

Fig. 6: *Adiabatic quantum annealing method. (a) Quantum tunneling (green) vs. thermal hopping (orange). (b) Time evolution of the instantaneous energy levels and definition of the minimum gap.*

$O(\log(N))$ support. Let us also note that while the aforementioned encodings use a Hilbert space of the same size as the fermionic Fock space, other encodings use larger Hilbert spaces (and thus more qubits than fermionic modes) to obtain local qubit Hamiltonians (at the expense of ancillary qubits) [18–20].

3.2 Quantum algorithms for Green functions

In the previous section, we introduced the main quantum computing building blocks to deal with fermionic models. In this section, we use these building blocks to introduce several methods to compute Green functions with gate-based quantum computers.

3.2.1 Ground state preparation: the example of the adiabatic method

The greater Green function $G^{>}(t)$ (see Eq. (23) for a definition) requires the preparation of the ground state $|\Psi_0\rangle$ (at zero temperature) or the Gibbs state $\rho = e^{-\beta H}/Z$ at finite temperature. In this section we tackle only the zero-temperature case, although quantum algorithms have also been developed for Gibbs state preparation, usually with the help of additional qubits.

A standard method to perform this state preparation is the adiabatic method. It consists of initializing the qubit register in an easy-to-prepare state $|\Phi_0\rangle$ that should also be the ground state of a Hamiltonian H_0. This Hamiltonian, sometimes called the mixer Hamiltonian, must be simple enough that its ground state $|\Phi_0\rangle$ is easy to prepare, and it must be such that it couples the different (usually unknown) eigenstates of the Hamiltonian H whose ground state $|\Psi_0\rangle$ we want to prepare. One then slowly deforms H_0 into H over a period of time $t_{\mathrm{annealing}}$, for instance

$$H(t) = \big(1-t/t_{\mathrm{annealing}}\big) H_0 + t/t_{\mathrm{annealing}} H \, . \tag{44}$$

In this time evolution, the role of H_0 can be seen as allowing for tunneling events between local minimal of the energy landscape of H, as pictorially represented in Fig. 6 (a). Over time, H_0 is slowly turned off, hence the name “quantum annealing” given to this method, in analogy to the classical (thermal) annealing method, which consists in slowly lowering the temperature. The adiabatic theorem (see [21]) guarantees that for long enough annealing times the system remains in the instantaneous ground state of $H(t)$. More precisely,

$$t_{\mathrm{annealing}} \gg V/\Delta^2_{\mathrm{min}}, \tag{45}$$

with V a matrix element, and Δ_{min} the minimum gap over time between the instantaneous ground state of $H(t)$ and the first excited state, as represented in Fig. 6 (b).

Thus, to prepare the ground state $|\Psi_0\rangle$ using the quantum annealing method, we need to construct a circuit that implements the time evolution $H(t)$ for a long enough $t_{\text{annealing}}$. The construction of the quantum circuit to perform this time evolution uses the methods described in subsection 3.1.3, for instance Trotterization. We note that a longer $t_{\text{annealing}}$ will generically lead to a deeper state preparation circuit: the best Hamiltonian evolution methods (like qubitization) achieve a linear scaling of the number of gates as a function of the evolution time. This scaling is optimal owing to the so-called no-fast-forwarding theorem, which stipulates that time evolutions of sparse Hamiltonians of duration t will require a number of gates at best linear in t [22]. As a consequence, an important practical question is the size Δ_{min}, which itself depends on H_0, H and the shape of the interpolation between both. It is in general hard to compute, but we at least know that $\Delta_{\text{min}} \geq \Delta$, with Δ the gap of H. Therefore, trying to prepare the ground state of a small-gap Hamiltonian (and for that matter a gapless one, like a metal) will take very long circuits.

There exists heuristic alternatives around adiabatic quantum annealing that set out to find shortcuts to adiabaticity (see [23] for a review). Variational methods like the ones we will introduce below (see subsection 3.4) can be regarded as examples of such heuristics.

The complexity of ground state preparation On a more formal level, let us note that the generation of the ground state of a many-body Hamiltonian is generally expected to be hard (that is, exponential) even on quantum computers: it has been proven to be "QMA-complete" for $k \geq 2$-local Hamiltonians [24] (and hence for the Hubbard model [25]). Here, k denotes the maximal support of the terms appearing in, e.g., the Pauli decomposition of H, and QMA—for quantum Merlin-Arthur—is a quantum analog of the NP class. Impurity problems may be less difficult in terms of ground state preparation: they are in the QCMA (for quantum-classical Merlin-Arthur) class [26], a class that stands between NP and QMA in terms of hardness. If they are gapped, they become easy (efficiently solvable) both for classical and quantum computers [26]. If not, recent work claims evidence that their ground state can be efficiently prepared by a quantum computer, while remaining hard for classical computers [27]. This would provide an a posteriori justification of quantum embedding methods like DMFT: using locality properties of the interaction, they would help reduce an exponentially problem (the Hubbard model) to an efficiently solvable problem (the impurity problem).

3.2.2 Time domain: Hadamard test

Once the ground state is prepared, the Hadamard test method to compute the greater Green function $G^{>}(t)$ (Eq. (23)) is a straightforward application (first introduced by [28], and well summarized in [29]) of the few methods we just introduced.

Using the decomposition (43) of $U^{\dagger}c_{\sigma}Uc_{\sigma}^{\dagger}$, we see that computing $G^{>}(t)$ amounts to summing four contributions of the form $\langle\Psi_0|U^{\dagger}P_iUP_j|\Psi_0\rangle$, with $P_{i,j} \in \left\{\sigma_x^{(1\uparrow)}, \sigma_y^{(1\uparrow)}\right\}$. Since $U^{\dagger}P_iUP_j$

is itself unitary, we only have to use the Hadamard test depicted in Fig. 5, with a controlled-$U^\dagger P_i U P_j$ operation. A naive implementation of this operation would be a sequence of a controlled P_j, controlled U, controlled P_i and then controlled $U^\dagger$. Yet, the precise structure of the $U^\dagger P_i U$ allows for one simplification: the controls on the time evolution U and its inverse $U^\dagger$ can be removed. Let us show that we get the same outcome with and without controls: with controls, if the ancilla qubit is in state $|0\rangle$ and the state register in state $|\psi\rangle$, we end up with $|0\rangle|\psi\rangle$. If the ancilla is in state $|1\rangle$, we end up with $|1\rangle U^\dagger P_i U|\psi\rangle$. Without controls, if the ancilla qubit is in state $|0\rangle$ and the state register in state $|\psi\rangle$, we end up with $|0\rangle U^\dagger U|\psi\rangle = |0\rangle|\psi\rangle$. If the ancilla is in state $|1\rangle$, we end up with $|1\rangle U^\dagger P_i U|\psi\rangle$. QED.

The corresponding circuit can be further simplified by noticing that the $U^\dagger$ operation has no influence on the final measurement as it commutes with the remaining operations (that act only on the ancilla qubit). It can thus be removed, which yields the circuit shown in Fig. 5(b): this circuit makes it clear that computing the Green function amounts to first generating the ground state $|\Psi_0\rangle$, coupling it to an ancilla qubit (controlled-P_j operation), time evolving the system (operation U), and then coupling again to the ancilla (controlled-P_i operation) and measuring the ancilla.

Let us end this subsection by estimating the scaling of the run time of the Hadamard test method with respect to the precision ϵ achieved on the Green function. Since it requires an estimate of $p(0)$ (Eq. (37)), it is characterized by a $1/\epsilon^2$ scaling: the statistical error on $p(0)$ scales as $\epsilon \sim 1/\sqrt{N_\text{samples}}$, with N_samples the number of repetitions of the circuit to estimate the probability of getting a 0 outcome. Therefore, the run time scales as $N_\text{samples} \sim 1/\epsilon^2$. This is typical of classical Monte-Carlo methods, and is a direct consequence of the central limit theorem. As we shall see in the next subsection, this scaling can be improved by resorting to quantum interferences.

3.2.3 Frequency domain: quantum phase estimation

High-level description Another quantum algorithm allows one to compute the Green function directly in the frequency domain, and with a better accuracy scaling. Starting from Eq. (17), and using $\int_0^\infty \mathrm{d}t\, e^{i\omega t} = \pi\delta(\omega) + i\mathcal{P}\frac{1}{\omega}$ ($\mathcal{P}$ denotes the Cauchy principal value), we obtain

$$G^>(\omega) = \sum_\alpha \left(-i\pi\delta\left(\omega - (E_\alpha - E_0)\right) + \mathcal{P}\left(\frac{1}{\omega - (E_\alpha - E_0)}\right)\right) \left|\langle\Psi_\alpha|c_\sigma^\dagger|\Psi_0\rangle\right|^2. \tag{46}$$

The imaginary part of $G^>(\omega)$ is therefore a distribution of peaks of weight $\left|\langle\Psi_\alpha|c_\sigma^\dagger|\Psi_0\rangle\right|^2$ at locations $E_\alpha - E_0$.

One of the most important quantum algorithms, quantum phase estimation (QPE, [30]), can be applied to this problem, as first pointed out by [29]. Given a unitary operator $U = e^{iHt}$ and one of its eigenvectors $|\Psi_\alpha\rangle$ with eigenvalue $e^{iE_\alpha t}$, it returns, in a single repetition of the QPE circuit, an m-bit estimate $\hat{E}_\alpha$ of E_α (t is adjusted so that $E_\alpha t$ is normalized: resolving smaller energies requires longer times). When starting from any state $|\Psi\rangle$ as an input, it returns a m-bit estimate $\hat{E}_\alpha$ of one of the eigenvalues E_α and projects the input state to $|\Psi_\alpha\rangle$ with probability $|\langle\Psi_\alpha|\Psi\rangle|^2$.

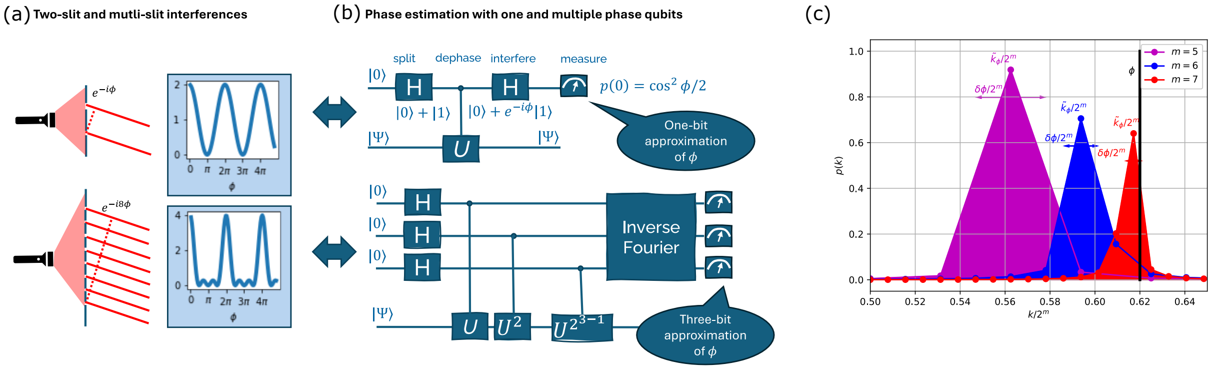

Fig. 7: *Phase estimation. Analogy between interferometry in optics (a) and quantum phase estimation circuits (b), with $m = 1$ ancilla qubit (two slits, top) or $m = 3$ ancilla qubits (8 slits, bottom). Adapted from [31]. (c) Histogram of the bitstring probabilities $p(j) = |\tilde{a}_j|$ for $\phi = E_\alpha t = 0.62$ and a variable number m of ancilla qubits.*

Therefore, several calls to QPE on an input state $|\Psi\rangle$ will return a histogram with peaks of height $|\langle\Psi_\alpha|\Psi\rangle|^2$ at location $\hat{E}_\alpha$. We thus see that applying QPE to the input state $c_\sigma^\dagger|\Psi_0\rangle$ and with $H = H_{\text{AIM}} - E_0$ will directly yield an estimate of $\operatorname{Im} G^>(\omega)$ (and hence, via Kramers-Kronig relations, of its real part).

In practice, however, $c_\sigma^\dagger|\Psi_0\rangle$ is not a properly normalized state because $c_\sigma^\dagger$ is not unitary. However, in a Jordan-Wigner encoding, assuming the impurity orbital is the first orbital in the orbital order, $c_\sigma^\dagger + c_\sigma$ maps to $\tilde{c}_\sigma^\dagger + \tilde{c}_\sigma = \sigma_x^{(1)}$, which is a bona fide unitary gate. Since the QPE circuit contains gates that conserve the electron number, an input state $c_\sigma^\dagger|\Psi_0\rangle + c_\sigma|\Psi_0\rangle$ will be projected onto eigenstates $|\Psi_\alpha\rangle$ of H with $N_e{+}1$ or $N_e{-}1$ electrons (where N_e is the number of electrons in the ground state $|\Psi_0\rangle$). To obtain the histogram of the greater Green function, we need to take into account only those measurements that stem from $N_e{+}1$ states. To select them, we just need to measure the number of electrons in the final state.

Circuit implementation We now turn to the implementation of QPE. QPE is essentially a multi-slit Mach-Zehnder interferometry experiment [32]. It can be thought of as a Young slit experiment, with a number of slits exponential in the number m of ancilla bits. Since interference experiments yield an accuracy scaling inversely with the number of slits, this will give a $1/2^m$ accuracy (Fig. 7(a)). Note that the $m{=}1$ circuit reduces to the Hadamard test circuit (or two Young slits).

The construction of the phase estimation circuit is illustrated in Fig. 7(b). It consists in two groups (registers) of qubits: the lower one, called the state register, for preparing a state $|\Psi\rangle$ close to an eigenstate state $|\Psi_\alpha\rangle$, and the upper one for m ancilla qubits ($m = 3$ on Fig. 7(b), lower circuit), that will be used to read a m-bit estimate of E_α.

If $|\Psi\rangle = |\Psi_\alpha\rangle$, the state after the wall of Hadamard gates and the controlled-$U^{2^{k-1}}$ ($k{=}1,\dots,m$) gates is

$$|\Psi_{\text{tot}}\rangle = \bigotimes_{k=1}^{m} \left(\frac{|0\rangle + e^{iE_\alpha 2^{k-1} t}|1\rangle}{\sqrt{2}} \right) \otimes |\Psi_\alpha\rangle = \frac{1}{2^{m/2}} \sum_{b_1,\dots b_m} e^{iE_\alpha t \sum_{k=1}^{m} b_k 2^{k-1}} |b\rangle \otimes |\Psi_\alpha\rangle.$$

Defining $b = \sum_{k=1}^{m} b_k \, 2^{k-1}$ (the integer representation of the binary string $(b_1, \ldots, b_m)$), we see that the state before the inverse Fourier transform is $|\Psi_{\text{tot}}\rangle = \sum_b a_b(E_\alpha)|b\rangle \otimes |\Psi_\alpha\rangle$, with $a_b(E_\alpha) = \frac{1}{2^{m/2}} e^{iE_\alpha tb}$. We further note that $a_{b+2\pi/(E_\alpha t)} = a_b$, if one assumes for the sake of argument that $2\pi/(E_\alpha t)$ is an integer. That is, the first part of the QPE circuit creates a state on the ancilla register whose amplitudes are a periodic function of period $2\pi/(E_\alpha t)$. The role of the Fourier transform is to extract this period by producing the associated spectrum, which we expect to be peaked at a frequency $\sim E_\alpha t$, from which we will extract the sought-after E_α.

Let us check this intuition by doing the math: the inverse quantum Fourier transform performs the (inverse) discrete Fourier transform (DFT) of the amplitudes a_b: it yields ket $\sum_j \tilde{a}_j(E_\alpha)|j\rangle$ on the ancilla register, with

$$\begin{aligned}\tilde{a}_j(E_\alpha) &= \text{DFT}^{-1}\left[a_b(E_\alpha)\right] = \frac{1}{\sqrt{2^m}} \sum_b e^{i2\pi bj/2^m} a_b(E_\alpha) \\ &= \frac{1}{2^m} \sum_{b=0}^{2^m-1} e^{i(2\pi j/2^m + E_\alpha t)b} = \frac{1}{2^m} \frac{\sin\left(\left(j + E_\alpha t \frac{2^m}{2\pi}\right)\pi\right)}{\sin\left(\left(j + E_\alpha t \frac{2^m}{2\pi}\right) 2^m \pi\right)}.\end{aligned}$$

Defining $\phi = E_\alpha t$, this function is peaked at $j = k_\phi$ with $k_\phi = \lfloor 2^m \phi \rfloor$, as shown in Fig. 7(c). We can thus take as an estimate of the actual phase ϕ (and thus E_α):

$$\hat{\phi} = \frac{k_\phi}{2^m} \tag{47}$$

with an error $\hat{\phi} - \phi = \delta\phi/2^m$: we see that we are making an error $\delta\phi/2^m$ on the actual phase ϕ: QPE does yield an estimate of the phase (and hence E_α) to an m-bit precision, as advertised.

Resource estimation To conclude this subsection, let us evaluate the depth (number of gates) of the QPE circuit, neglecting the state preparation: the dominant contribution is given by the controlled-U^{2^k} operations ($k{=}\,0$ to $m{-}1$). Each U^{2^k} operation (neglecting the controls, which will result in an additional overhead) will at least require a depth of 2^k as per the non-fast-forwarding theorem ($U^{2^k} = e^{-iH(2^k t)}$ so the evolution time is $\propto 2^k$), adding up to $O(2^m)$ gates. The quantum Fourier transform requires $O(m^2)$ gates, which is negligible compared to the controlled evolutions. Here, we did not specify the implementation of U itself, but we could use Trotterization, which will introduce a dependence of the gate count on the number of terms M and support of each term s as per Eq. (33), and achieve a t^2 depth (at first order). Qubitization will achieve a linear dependence in t, with additional dependence on the coefficients of the sum-of-unitaries decomposition.

Let us emphasize that the $O(2^m)$ depth is not to be considered an exponential cost: it helps achieve an error $\epsilon = O(1/2^m)$ on the estimate, so that the run time of QPE scales as $O(1/\epsilon)$, a scaling known as "Heisenberg scaling". This scaling can be shown to be optimal. It is quadratically better than the $1/\epsilon^2$ scaling of the Hadamard test method above, at the cost of a longer circuit.

Let us finally point out that in both methods, the circuit depth (excluding state preparation) scales polynomially with the system size, in contrast to the classical methods that all come with

some sort of exponential scaling. As for the state preparation itself, it may require very long circuits as discussed in subsection 3.2.1.
Alternative methods exist. [33] uses the resolvent form of the Green function, Eq. (18). Using this form, we can transform the problem of computing the Green function to the solution of a linear system of equations $Ax = b$ with $b = c_\sigma^\dagger|\Psi_0\rangle$ and $A = E_0 - H + \omega + i\eta$ (typical algorithms for this task use QPE as a subroutine). The overlap with solution $x = \frac{1}{E_0 - H + \omega + i\eta} c_\sigma^\dagger |\Psi_0\rangle$ can then be computed with variations of the Hadamard test. Alternatively, [33] advocates the use of other advanced methods to compute the matrix inverse appearing in the resolvent formula.

3.3 The exponential wall of decoherence: noisy quantum states and gates

The algorithms presented above in principle allows for a tractable (polynomial) computation of the (greater) Green function in real time or frequency, provided the ground or Gibbs state can be prepared... and provided we are working with a perfect, that is, decoherence-free, quantum computer. Here, we discuss the impact of decoherence.

Decoherence: density matrix and quantum channels In subsection 3.1, we described perfect quantum computers. The physical implementations of quantum computers that have become available in the recent years differ from this ideal model. Their main constraint is decoherence, a phenomenon that stems from the entanglement of the quantum computer with the outside world, often called the environment. Denoting by $|\psi_i\rangle$ a basis of the quantum computer's Hilbert space and $|\chi_j\rangle$ a basis of said environment, the total wavefunction reads $|\Psi_{\mathrm{tot}}\rangle = \sum_{ij} c_{ij} |\psi_i\rangle \otimes |\chi_j\rangle$. It is however impractical to work with such an object given the size of the environment. All average observables on the quantum computer, of the form $\langle O\rangle = \langle \Psi_{\mathrm{tot}}|\hat{O} \otimes I|\Psi_{\mathrm{tot}}\rangle$, can be computed from the so-called density matrix $\rho = \mathrm{Tr}_{\mathrm{env}}\left(|\Psi_{\mathrm{tot}}\rangle\langle\Psi_{\mathrm{tot}}|\right)$ of the quantum computer (here $\mathrm{Tr}_{\mathrm{env}}$ denotes the partial trace $\sum_j \langle\chi_j|\cdots|\chi_j\rangle$) through the formula

$$\langle O\rangle = \mathrm{Tr}(\rho\,\hat{O}), \tag{48}$$

where the trace here is to be understood as the trace over the quantum computer's Hilbert space. Therefore, for all practical purposes, quantum states in the presence of decoherence will no longer be described by a wavefunction $|\Psi\rangle$ but by a density matrix ρ, a unit trace and positive definite matrix (as can be checked from its definition). Of course, the density matrix also captures so-called pure states (quantum states described by a single wavefunction $|\Psi\rangle$): in a pure state has density matrix $\rho = |\Psi\rangle\langle\Psi|$. Similarly, the time evolution of quantum states (which we conveniently described with unitary gates) will no longer be described by a unitary operator $|\Psi_{\mathrm{f}}\rangle = U|\Psi_{\mathrm{i}}\rangle$. Instead we will consider linear, completely positive and trace-preserving (CPTP) mappings $\mathcal{E}$ that act on the density matrix as

$$\rho_{\mathrm{f}} = \mathcal{E}(\rho_{\mathrm{i}}) = \sum_{k=1}^{K} E_k\,\rho_{\mathrm{i}} E_k^\dagger, \tag{49}$$

with $\sum_k E_k^\dagger E_k = I$. A noisy gate (also called a quantum channel or CPTP map) is thus completely characterized by K operators $\{E_k\}$ known as Kraus operators. The trace preservation property ensures that the normalization of the state ($\mathrm{Tr}\,\rho = 1$) is preserved, while complete positivity means that applying the channel $\mathcal{E}$ on any subsystem (i.e applying $\mathcal{E} \otimes \mathcal{I}$ with $\mathcal{I}$ the identity channel) keeps the total density matrix positive definite (applying a gate on, say, a single qubit should keep the many-qubit quantum register physical). Note that when the Kraus rank K is one, we revert to the unitary case as $K_1^\dagger K_1 = I$ and $\rho_\mathrm{f} = K_1|\Psi_\mathrm{i}\rangle\langle\Psi_\mathrm{i}|K_1^\dagger = |\Psi_\mathrm{f}\rangle\langle\Psi_\mathrm{f}|$.

Standard error models Decoherence in quantum processors is usually described with a few error metrics. An oft-used metric is the T_1 relaxation time and the T_2 dephasing time, which are obtained through Rabi and Ramsey experiments, respectively. From T_1, one can extract a typical relaxation probability $p_\mathrm{AD}(T_1)$ that enters the quantum channel $\mathcal{E}_\mathrm{AD}$, dubbed amplitude damping, that corresponds to this noise process. It is characterized by the Kraus operators

$$K_1 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p_\mathrm{AD}} \end{pmatrix}, \quad K_2 = \sqrt{p_\mathrm{AD}}\,\sigma_-. \tag{50}$$

This means that with probability p_AD, the state relaxes from $|1\rangle$ to $|0\rangle$. Similarly, from T_1 and T_2 one can compute a so-called pure-dephasing time $T_\varphi = \big(1/T_2 - 1/(2T_1)\big)^{-1}$, which in turn determines a pure-dephasing probability $p_\mathrm{PD}(T_\varphi)$ that enters the pure-dephasing quantum channel $\mathcal{E}_\mathrm{PD}$. It is characterized by the Kraus operators

$$K_1 = \sqrt{1-p_\mathrm{PD}}\,I, \quad K_2 = \sqrt{p_\mathrm{PD}}\,\sigma_z. \tag{51}$$

This means that with probability p_PD, a superposition state $|0\rangle+|1\rangle$ is dephased to $|0\rangle-|1\rangle$. A similar channel, called the bit-flip channel (with σ_x to replace σ_z in K_2), corresponds to bit-flip errors.

Finally, let us introduce a widely used noise channel called the depolarizing channel that, contrary to the pure-dephasing and bit-flip channels, does not favor errors of one type (e.g. z or x) over others. It is often defined (in the N-qubit case) as

$$\mathcal{E}_\mathrm{D}(\rho_\mathrm{f}) = (1-p_\mathrm{D})\,\rho_\mathrm{f} + p_\mathrm{D} I/2^N. \tag{52}$$

Namely, with probability p_D the state is changed into the so-called completely mixed state $I/2^N$, where all quantum information is erased. (One can easily show that this channel has four Kraus operators proportional to I, σ_x, σ_y and σ_z, respectively, with the same coefficient in front of the Pauli matrices, hence the claim that it does not favor one error type over the others). The depolarizing channel is often used for lack of more precise information as to the type of errors. For instance, when one knows only the error rate of a certain unitary gate $U^{(\mathrm{g})}$, one often models it as the composition $\mathcal{E}^{(g)} = \mathcal{E}_\mathrm{D} \circ \mathcal{E}^{(g)}_\text{perfect}$, with $\mathcal{E}^{(g)}_\text{perfect}(\rho) = U^{(g)}\rho\, U^{(g)\dagger}$.

Let us end this section by examining the overall effect of noise on a quantum circuit with N_g gates that we assume to be plagued by the same depolarizing noise $\mathcal{E}_\mathrm{D}$. The final state can be written as

$$\rho_\mathrm{f} = \mathcal{E}^{(N_\mathrm{g})} \circ \mathcal{E}^{(N_\mathrm{g}-1)} \circ \cdots \circ \mathcal{E}^{(1)}(\rho_\mathrm{i}). \tag{53}$$

Using the simple compositional noise model we just used, we easily get the expression

$$\rho_{\mathrm{f}} = (1-p_{\mathrm{D}})^{N_{\mathrm{g}}}\, U|\Psi_{\mathrm{i}}\rangle\langle\Psi_{\mathrm{i}}|U^\dagger + \Big(1-(1-p_{\mathrm{D}})^{N_{\mathrm{g}}}\Big)\, I/2^N,$$

with $U = U^{(N_{\mathrm{g}})}\cdot U^{(N_{\mathrm{g}}-1)}\cdots U^{(1)}$ the perfect circuit. We can readily evaluate the so-called fidelity of the noisy final state with the perfect final state $|\Psi_{\mathrm{f}}\rangle = U|\Psi_{\mathrm{i}}\rangle$

$$\mathcal{F}(\rho_{\mathrm{f}}, |\Psi_{\mathrm{f}}\rangle) = \langle\Psi_{\mathrm{f}}|\rho_{\mathrm{f}}|\Psi_{\mathrm{f}}\rangle = (1-p_{\mathrm{D}})^{N_{\mathrm{g}}} + \Big(1-(1-p_{\mathrm{D}})^{N_{\mathrm{g}}}\Big)\, 1/2^N \approx (1-p_{\mathrm{D}})^{N_{\mathrm{g}}} \approx e^{-p_{\mathrm{D}}N_{\mathrm{g}}}. \quad (54)$$

The fidelity of the final state of a quantum circuit decays exponentially with the number of gates N_{g} in this circuit. Conversely, this means that for a given error rate p_{D} only circuits of size $N_{\mathrm{g}} \lesssim 1/p_{\mathrm{D}}$ are viable.

Today's error rates are between 1% and 0.01% for the limiting operations (two-qubit gates), meaning only circuits that contain between 100 and 10,000 of these gates are viable. This, in practice, places very severe constraints on the quantum circuits that can actually be run on current and near-term processors. For instance, this rules out the quantum phase estimation circuits we discussed in subsection 3.2.3. Even simple Trotterization is limited to very short times and hence poor energy resolutions. In fact, it already severely limits the preparation of the ground state needed for the Green function. In particular, the use of adiabatic state preparation methods presented in subsection 3.2.1, which require long circuits due to the long annealing times, is prohibited.

To address this issue, new algorithms have been developed in the past decade to reduce the number of requisite gates. This is the topic of the next section.

3.4 Near-term algorithms. Variational quantum algorithms for ground states and for time evolution.

Current and near-term quantum processors come with drastic limitations in terms of the number of gates that can be executed before decoherence sets in. Starting in the mid-2010s, algorithms have been designed for these "noisy, intermediate-scale quantum" (NISQ [34]) devices.

3.4.1 Ground state preparation: the variational quantum eigensolver

The most prominent example is the variational quantum eigensolver (VQE, [35], see [36] for a review), which is used to prepare ground states of Hamiltonians and measure their energy. Like DMRG above (paragraph 2.2.1), VQE is a variational method that sets out to minimize a variational energy $E_{\boldsymbol{\theta}} = \big\langle\Psi(\boldsymbol{\theta})|H|\Psi(\boldsymbol{\theta})\big\rangle$, with $|\Psi(\boldsymbol{\theta})\big\rangle$ a family of variational states (also called "ansatz"). The specificity of VQE is that the variational states are generated on a quantum processor by a unitary operation $U(\boldsymbol{\theta})$

$$|\Psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|\Phi_{\mathrm{init}}\rangle,$$

where $|\Phi_{\mathrm{init}}\rangle$ is usually state $|0\rangle\otimes|0\rangle\cdots\otimes|0\rangle$, and $U(\boldsymbol{\theta})$ represents a quantum circuit with parameters $\boldsymbol{\theta}$ (that may be, for instance, the angles of rotation gates $R_\alpha(\theta)$). Then, the energy

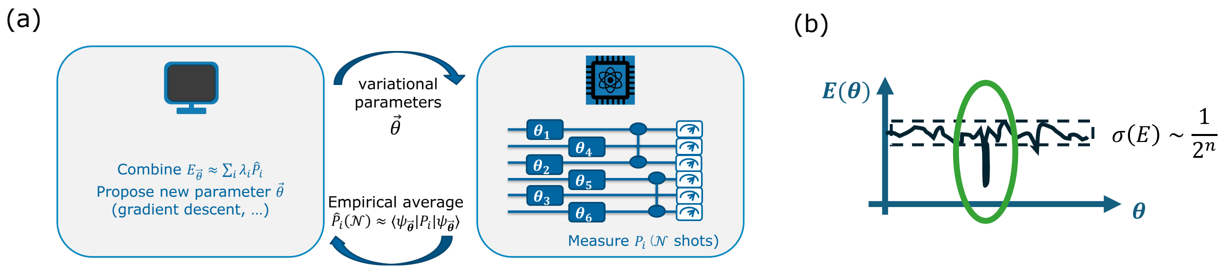
(a)
Combine $E_{\vec{\theta}} \approx \sum_i \lambda_i \hat{P}_i$
Propose new parameter $\vec{\theta}$
(gradient descent, ...)
variational parameters $\vec{\theta}$
Empirical average $\hat{P}_i(\mathcal{N}) \approx \langle\psi_{\vec{\theta}}|P_i|\psi_{\vec{\theta}}\rangle$
θ_1
θ_4
θ_2
θ_5
θ_3
θ_6
Measure P_i ($\mathcal{N}$ shots)
(b)
$E(\theta)$
$\sigma(E) \sim \frac{1}{2^n}$
θ

Fig. 8: *(a) Sketch of the variational quantum eigensolver (VQE) algorithm. (b) Illustration of the barren plateau problem: the variance vanishes exponentially on average.*

$E_{\boldsymbol{\theta}}$ of this state is measured on the quantum computer. As explained in subsection 3.1.4, this measurement can be done by achieving a Pauli decomposition of H (see Eq. (30)) and then summing up each contribution:

$$E_{\boldsymbol{\theta}} = \sum_{i=1}^{M} \lambda_i \langle \Psi(\boldsymbol{\theta})|P_i|\Psi(\boldsymbol{\theta})\rangle. \tag{55}$$

Each expectation value $\langle \Psi(\boldsymbol{\theta})|P_i|\Psi(\boldsymbol{\theta})\rangle$ is then estimated by an empirical average over N_{s} shots. Based on the computed $E_{\boldsymbol{\theta}}$, a classical optimizer updates the parameters $\boldsymbol{\theta}$. This is summarized in Fig. 8(a). VQE comes with three main challenges: decoherence, the measurement of $E_{\boldsymbol{\theta}}$, and the variational optimization of the parameters.

Decoherence and the design of ansätze Decoherence, as we just saw, limits the size of the circuit $U(\boldsymbol{\theta})$ that can be run in VQE. The main advantage of VQE is its flexibility in terms of the ansatz: given a fixed gate budget, one is free to pick the ansatz that best fits the problem at hand. Ansatz construction strategies fall into two main categories: hardware-efficient ansätze [37] and physics-inspired ones. The former aims at achieving the most expressivity given the gates available on the quantum processor. Typically, hardware-efficient ansätze feature a repetition of the following pattern: a sequence of one qubit rotations applied to each qubit, followed by the hardware's native two-qubit gate (there is usually only one) applied to all qubits pairs compatible with the connectivity of the qubits. The rotations introduce variational parameters, while the two-qubit gates generate entanglement. In fact, this construction guarantees a fast growth of entanglement—an important feature given the limited coherence time. However, it does not take into account the specifics of the Hamiltonian at hand. This is what the latter approach does.

For instance, the Hamiltonian variational ansatz (HVA, [38]) draws inspiration from the adiabatic quantum annealing algorithm (subsection 3.2.1): based on the intuition that one should start with the ground state $|\Phi_0\rangle$ of a simple Hamiltonian H_0 and deform the latter to the Hamiltonian at hand H, it uses a Trotterized form of the time evolution

$$|\Psi(t_{\text{annealing}})\rangle = U(t_{\text{annealing}})|\Phi_0\rangle = \prod_{k=1}^{M} e^{-i\left(1-t_k/t_{\text{annealing}}\right)H_0 t_k} e^{-it_k/t_{\text{annealing}} H t_k} |\Phi_0\rangle$$

(for $t_k = kt_{\text{annealing}}/M$) as an inspiration for the following variational form

$$|\Psi(\boldsymbol{\theta})\rangle = \prod_{k=1}^{M} e^{-i\theta_{2k}H_0} e^{-i\theta_{2k+1}H} |\Phi_0\rangle \tag{56}$$

with $2M$ parameters $\boldsymbol{\theta}$. (Then the operators $e^{-i\theta_{2k}H_0}$ and $e^{-i\theta_{2k+1}H}$ can be translated to quantum circuits using the Trotterization tools we introduced in subsection 3.1.3). In the limit of large M, the assignment $\theta_{2k} = (1 - t_k/t_{\text{annealing}})t_k$ and $\theta_{2k+1} = t_k^2/t_{\text{annealing}}$ of the parameters guarantees a convergence to the ground state. However, the premise of HVA is that the variational optimization of $\boldsymbol{\theta}$ can lead to a good enough approximation of the ground state in a much shorter time (that is, compatible with the coherence time)... providing a sort of shortcut to adiabaticity. Typically, in the context of the AIM (Eq. (10)), a natural choice of the simple Hamiltonian is $H_0 = Un_\uparrow n_\downarrow - \mu \sum_\sigma c_\sigma^\dagger c_\sigma + \sum_{k\sigma} \varepsilon_k\, a_{k\sigma}^\dagger a_{k\sigma}$ because its ground state, after a Jordan-Wigner encoding, is a computational basis state that can be prepared with σ_x gates only. Then, one can replace H with $H' = \sum_{k\sigma} V_k \big(a_{k\sigma}^\dagger c_\sigma + \text{h.c.}\big)$.

Other approaches are targeted more specifically at fermionic states: for instance, the low-depth circuit ansatz (LDCA, [39]) proposes to intercalate so-called Gaussian gates (which generate only Gaussian states, namely states generated by Hamiltonians quadratic in the c and $c^\dagger$ operators) with non-Gaussian gates (corresponding to quartic terms or beyond).

Once the ansatz is picked, other optimizations to reduce the circuit depth and thus counter decoherence is the choice of the orbital basis in which the states are expressed. For instance, while a Fock state $\prod_{k=1}^{N} \big(c_k^\dagger\big)^{n_k} |0, \dots, 0\rangle$ expressed in the Fock basis can be generated, upon Jordan-Wigner encoding, simply by applying σ_x gates to the qubits with occupations $n_k = 1$, if the same state is expressed in any other basis, longer circuits (possibly including entangling gates) will be necessary. For a generic state $|\Psi\rangle$, the choice of the natural-orbital basis (that diagonalizes the one-particle reduced density matrix $D_{ij} = \langle\Psi|c_i^\dagger c_j|\Psi\rangle$) is supposed to yield the most compact representation of the state and should therefore require shorter quantum circuits [40–42].

Finally, one can also build the ansatz iteratively instead of starting from a fixed ansatz: this is what the ADAPT-VQE method ([43,44]) does. At each step, it picks from a pool of gates the gate that maximizes the gradient of the energy with respect to the gate's parameter, therefore ensuring the fastest convergence to the minimum.

A major mostly uncharted territory is the design of ansätze specific to the nature of the AIM (beyond what the HVA does). In particular, the fact that the bath is non-interacting, which allows for drastic simplifications in action-based impurity solvers (subsection 2.2.2), has not been exploited so far.

The measurement problem VQE resorts to a classical summation of individual samples to compute the variational energy $E_{\boldsymbol{\theta}}$ (Eq. (55)). This means that the statistical error will scale as $\epsilon = O(1/\sqrt{N_{\text{s}}})$, with N_{s} the number of samples (shots). As explained before, this scaling of $1/\epsilon^2$ of the run time is much less favorable than the $1/\epsilon$ scaling of quantum phase estimation.

The prefactor of this scaling can be mitigated by many methods. One can, for instance, group terms that commute with one another: they can be measured without regenerating the state. One can also allocate the total shot budget based on the importance of each term in the Pauli decomposition: larger coefficients $|\lambda_i|$ in Eq. (55) should receive more shots. The optimal shot allocation can be worked out [45] (and see [46] for an adaptive shot allocation strategy), leading to a standard error on the mean that satisfies the upper bound:

$$\Delta E_{\boldsymbol{\theta}} \leq \frac{\sum_{i=1}^{M} |\lambda_i|}{\sqrt{N_{\mathrm{s}}}}. \tag{57}$$

The numerator of the upper bound is referred to as the one-norm H. Proposals have been made to minimize it by orbital rotations [47].

Another method called classical shadows [48] is based on the realization that measuring $E_{\boldsymbol{\theta}}$ by measuring each P_i (or group thereof) individually on the final state $|\Psi(\boldsymbol{\theta})\rangle$ (or $\rho_{\boldsymbol{\theta}}$ in the general case where it is afflicted by noise) is a poor reuse of the information in $\rho_{\boldsymbol{\theta}}$. Classical shadows thus first produce an estimate (called classical shadow) $\hat{\rho}_{\boldsymbol{\theta}}$ of $\rho_{\boldsymbol{\theta}}$ by appending gates drawn randomly for smartly chosen groups of unitaries to the circuit and then measuring along the σ_z axis…and then return estimates $\langle \hat{P}_i \rangle = \mathrm{Tr}(\hat{\rho}_{\boldsymbol{\theta}} P_i)$ to compute the energy $E_{\boldsymbol{\theta}}$. By so doing, the statistical error is still behaving as $1/\sqrt{N_{\mathrm{s}}}$(as per the central limit theorem), but the numerator can be mitigated (achieving, in some cases, a $\log(M)$ scaling, with M the number of Pauli terms). Adaptations to fermionic settings have appeared recently, albeit not in a VQE context [49].

The optimization problem Supposing the ansatz is short enough to beat decoherence, and the energy $E(\boldsymbol{\theta})$ can be measured to enough statistical accuracy, the VQE practitioner is still facing the problem of finding the minimum of $E_{\boldsymbol{\theta}}$. The difficulty of this optimization task has been a major focus of recent years.

On the practical side, methods have been developed to measure the gradients required in gradient-based minimization methods, like the gradient descent method, which updates parameters through

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \alpha \boldsymbol{\nabla}_{\theta} E(\boldsymbol{\theta}_i), \tag{58}$$

with a learning rate α and $[\boldsymbol{\nabla}_{\theta} E(\boldsymbol{\theta})]_k = \frac{\partial E(\boldsymbol{\theta})}{\partial \theta_k} = \frac{\partial}{\partial \theta_k} \langle \Psi_\theta | H | \Psi_\theta \rangle$. It turns out that if the parametric gates used in the ansatz are of the form $U_k(\theta_k) = e^{-i\theta_k/2P_k}$, with $P_k^2 = I$, then the following identity (called "parameter shift rule") can be easily derived

$$\frac{\partial E(\boldsymbol{\theta})}{\partial \theta_k} = \frac{1}{2}\big(E(\theta_k{+}\pi/2) - E(\theta_k{-}\pi/2)\big), \tag{59}$$

where $E(\theta_k{+}\alpha)$ is shorthand for $E(\theta_1, \ldots, \theta_k{+}\alpha, \ldots, \theta_L)$. Estimating gradients is therefore relatively straightforward. It however requires a number of evaluations proportional to the number L of parameters, a major overhead compared to the $O(1)$ cost of computing gradients in classical deep neural networks using the backpropagation algorithm [50]. Other machine-learning-inspired methods like natural gradients can also be implemented [51], albeit at a much larger cost since the required quantum geometric tensor necessitates $O(L^2)$ energy evaluations.

On the more formal side, the potential landscape $E(\boldsymbol{\theta})$ (and more specifically its variance) has been studied for some classes of variational circuits $U(\boldsymbol{\theta})$ and observables H. The empirically observed vanishing of gradients in large regions of the variational space, dubbed "barren plateau problem" [52] is now understood as a curse of dimensionality issue [53, 54]: the larger the dimension of the effective space explored by the variational search, the more concentrated $E(\boldsymbol{\theta})$ around its mean and therefore the smaller the gradients (on average, as illustrated in Fig. 8(b)). This issue is particularly pressing for the hardware-efficient parametrized circuits we introduced above: if they are deep enough, the ensemble of unitaries $U(\boldsymbol{\theta})$ is very close to the group of Haar random matrices and the variance $\left\langle \left(E(\boldsymbol{\theta}) - \langle E(\boldsymbol{\theta})\rangle\right)^2\right\rangle$ over the parameter landscape vanishes exponentially with the number N of qubits. In other words, the more expressive, the less trainable parametrized circuits become. On the other hand, physics-inspired ansätze, which usually come with more structure, could suffer less from this issue. Note also that the existence of barren plateaux is an average phenomenon: on average, energies are (exponentially) close to their mean. However, non-zero gradient portions of the parameter space exist (see circled region in 8(b)). The challenge lies in finding them. In particular, finding a good starting point $\boldsymbol{\theta_0}$ for the optimization appears crucial. There again, physics-informed choices (by, e.g., perturbation theory or classical pretraining of the variational ansatz) could play a crucial role. Adaptive techniques like the aforementioned ADAPT-VQE methods, or other ways of solving the eigenvalue problem (see e.g [55]) could also mitigate this issue.

3.4.2 Time evolution: variational circuits for time evolution.

The Trotter circuits described in a previous section are often too long compared to the available number of gates given the coherence time. Methods, like "variational quantum simulation" [56], have been designed to train variational circuits to perform a given time evolution. They are a simple adaptation of the variational principle used in VQE to the time-dependent case: given a variational state $|\Psi(\boldsymbol{\theta}(t))\rangle = U(\boldsymbol{\theta}(t))|\Psi_{\text{init}}\rangle$ ($|\Psi_{\text{init}}\rangle = c^\dagger|\Psi_0\rangle$ for the greater Green function), one can use the McLachlan variational principle

$$\min \delta \left\| \left(\frac{\mathrm{d}}{\mathrm{dt}} + iH\right) |\Psi(\boldsymbol{\theta}(t))\rangle \right\|$$

to write a differential equation for the parameters $\boldsymbol{\theta}(t)$

$$\sum_j M_{ij}\dot{\theta}_j = V_i$$

with $M_{i,j} = \mathrm{Re}\left(\frac{\partial\langle\Psi(\boldsymbol{\theta}(t))|}{\partial\theta_i}\frac{\partial|\Psi(\boldsymbol{\theta}(t))\rangle}{\partial\theta_j}\right)$ and $V_i = \mathrm{Im}\left(\langle\Psi(\boldsymbol{\theta}(t))|H\frac{\partial|\Psi(\boldsymbol{\theta}(t))\rangle}{\partial\theta_i}\right)$. Both quantities can be evaluated with dedicated quantum circuits similar to the Hadamard test [57, 58]. Interestingly, the aforementioned natural gradient approach can be obtained by invoking the McLachlan principle with imaginary times.

3.4.3 Reducing the effect of imperfections: error mitigation.

All the methods above aim at creating shorter circuits, but still suffer from decoherence. A variety of techniques known as error mitigation has been developed over the years to counter the exponential decay of fidelity (Eq. (54)). They generically trade a systematic bias generated by decoherence for an increased classical cost—that essentially helps recover the lost information. A representative example of this tradeoff is probabilistic error cancellation [59, 60]. To suppress the bias in energy in a noisy VQE, caused by the deviation between the noisy energy $E_{\text{noisy}} = \text{Tr}(\rho_{\text{f}} H)$, with ρ_{f} given by Eq. (53), and the perfect energy $E_{\text{perfect}} = \langle \Psi_{\text{f}} | H | \Psi_{\text{f}} \rangle$, one decomposes the "perfect" channels $\mathcal{E}^{(k)}_{\text{perfect}}(\rho)$ as a linear combination of the actual (noisy) channels $\mathcal{E}^{(k)}(\rho)$ that are realized in the experiment

$$\mathcal{E}^{(k)}_{\text{perfect}} = \sum_{l} q^{(k)}_{l} \mathcal{E}_{l}, \tag{60}$$

with the underlying assumption that these noisy channels form an independent family, and the $\sum_l q^{(k)}_l = 1$ to preserve the trace-preserving character (but the $q^{(k)}_l$ may be negative). The perfect energy can thus be decomposed as a very large sum, which is sampled through Monte-Carlo

$$\begin{aligned} E_{\text{perfect}} &= \sum_{l_1 \ldots l_{N_{\text{g}}}} q^{(1)}_{l_1} \cdots q^{(N_{\text{g}})}_{l_{N_{\text{g}}}} \,\text{Tr}\big(H \mathcal{E}_{l_{N_{\text{g}}}} \circ \mathcal{E}_{l_{N_{\text{g}}-1}} \circ \cdots \mathcal{E}_{l_1}(\rho_{\text{i}})\big) \\ &= \Gamma \sum_{l_1 \ldots l_{N_{\text{g}}}} p^{(1)}_{l_1} \cdots p^{(N_{\text{g}})}_{l_{N_{\text{g}}}} \text{s}(l_1, \ldots l_{N_{\text{g}}}) \,\text{Tr}\big(H \mathcal{E}_{l_{N_{\text{g}}}} \circ \mathcal{E}_{l_{N_{\text{g}}-1}} \circ \cdots \mathcal{E}_{l_1}(\rho_{\text{i}})\big) && (61) \\ &\approx \frac{\Gamma}{N_s} \sum_{i=1}^{N_s} \text{s}(l^{(i)}_1, \ldots l^{(i)}_{N_{\text{g}}}) \,\text{Tr}\big(H \mathcal{E}_{l^{(i)}_{N_{\text{g}}}} \circ \mathcal{E}_{l^{(i)}_{N_{\text{g}}-1}} \circ \cdots \mathcal{E}_{l^{(i)}_1}(\rho_{\text{i}})\big). && (62) \end{aligned}$$

Here, the probabilities $p^{(k)}_l = q^{(k)}_l / \sum_{l'} q^{(k)}_{l'}$ are introduced to deal with the possible negativity of the $q^{(k)}_l$, $\Gamma = \prod_{k=1}^{N_g} \sum_{l'} \big| q^{(k)}_{l'} \big|$ and $\text{s}(l_1, \ldots l_{N_{\text{g}}}) = \prod_{k=1}^{N_g} \text{sign}(q^{(k)}_{l_k})$. The advantage of the summand of Eq. (62) is that it can be estimated with the noisy computer at hand by running a circuit with (noisy) gates $\mathcal{E}_{l^{(i)}_1}, \ldots, \mathcal{E}_{l^{(i)}_{N_{\text{g}}}}$ and measuring H at the end, resulting in an unbiased estimate of E_{perfect}! The price to pay is, however, twofold: one needs to know precisely the noise models of the hardware to be able to perform the decomposition of Eq. (60), which requires quite expensive process tomography experiments. More fundamentally, the estimator comes with a statistical uncertainty that scales with the Γ factor, which can be rewritten as $\Gamma = \prod_{k=1}^{N_g} (1+2\eta_k)$, with the so-called "negativity" $\eta_k = \sum_{l, q_l < 0} q^{(k)}_l$. To get an intuition how it scales, let us suppose that all gates have the same negativity η. Then

$$\Gamma \approx e^{2\eta N_{\text{g}}}. \tag{63}$$

In other words, to get a fixed statistical error bar $\epsilon \propto \Gamma / \sqrt{N_{\text{s}}}$, one needs to scale the number N_{s} of samples as $\Gamma^2 = e^{4\eta N_{\text{g}}}$, namely exponentially with the number of gates! This essentially tells us that to fight against the exponential loss in fidelity (Eq. (54)), we need to pay an exponential price. In practice, this exponential can be manageable provided the negativity is not too large. The better the hardware, the closer η to unity, the lower the sampling overhead.

4 Conclusion: state of the art, challenges and ways ahead

The tools we just introduced have been used in the past 10 years to solve the impurity model of DMFT with the help of quantum processors. Table 1 summarizes the various attempts. After the original proposal by [61] and [62], most works [63–66,68,71,73] focused on the simplest DMFT scheme, namely two-site DMFT [76], which consists in limiting the bath to only one bath site, requiring only 4 qubits. Most works used VQE in combination with the Hadamard test circuit we introduced before, with various ansätze and methods to optimize the time evolution needed in the Hadamard test. The Hadamard test method is replaced by a Lanczos-type algorithm in [69, 75] that builds the Krylov states using a variational ansatz $|\chi_n\rangle = U(\boldsymbol{\theta}_n)|\Phi_{\text{init}}\rangle$.

Driven by the optimization issues of variational methods, quantum subspace expansions are also proposed, in an impurity model context, by [70]: the Krylov bases (whether for the ground state or for the Green function, see paragraph 2.2.1) are constructed not in the full Hilbert space but in a reduced subspace built with states obtained by Trotter iterations: $|\psi_k\rangle = U(\delta t)^k|\Phi_{\text{init}}\rangle$, with $U(\delta t) = e^{-iH_{\text{AIM}}\delta t}$. The conjecture of the method is that Trotter iterates will yield a good enough subspace of the full Hilbert space to perform the Lanczos method in. The first (ground state) Lanczos iteration can also be replaced by a purely classical method, as advocated in [72]: there, DMRG is used to compute the ground state in the matrix product state form. The MPS is converted to a quantum circuit and then Trotter iterates $|\psi_k\rangle$ are generated on top of this state.

The above quantum Krylov variants require the computation of various matrix elements of the Trotter iterates $|\psi_k\rangle$, which incurs the aforementioned $1/\epsilon^2$ overhead. [77] proposes a way to avoid this overhead by simply sampling bitstrings $s_{i,k}$ from the $|\psi_k\rangle$'s and building the classical representation of H in the subspace corresponding to these bitstrings. [77] then classically finds the ground state energy by diagonalizing the restriction of H to this subspace for systems of up to 41 bath sites (but does not compute the Green function).

[63] looks at a nonequilibrium setting, which requires extending the formalism we introduced to Keldysh Green functions.

Alternative embedding techniques—which can be regarded as low-energy approximations of DMFT [6]—have been used, motivated by the fact that these techniques require an impurity problem with fewer bath sites ($N_b = N_c$), and require not the full time-dependent Green function, but only the one-particle reduced density matrix (essentially the t=0 Green function). This includes rotationally-invariant slave bosons (aka Gutzwiller [67,40]) and density-matrix embedding theory [78–80]. Finally, simpler slave-particle methods like $\mathbb{Z}_2$ slave spins were used to reduce the Hubbard model to even simpler (namely spin-based, as opposed to fermionic) effective models [81].

The current state of affairs is that the small sizes considered in the aforementioned publications are not yet large enough (except perhaps for [72, 77]) that ground state preparation issues with VQE (like barren plateaux) become severe, so that whether a variational preparation of low-energy states of the AIM with quantum methods can succeed is still an open question. As for the time evolution needed to compute the Green function, the decoherence rates of current processors have so far prevented one from reaching numbers of bath sites competitive with

Ref.	N_{c}	N_{b}	State preparation	Green function method	Noisy emul.	Physical implem.	Remark
[62]	any	any	Adiabatic state preparation	Hadamard test	No	No	Generic proposal.
[61]	1	10	N-A	Hadamard (Trotter)	Yes	Ions (th)	Nonequilibrium: kinetic ramp
[63]	1	1	Exact	Hadamard with Trotter	Yes	SC (th)	Study of noise in bath.
[64]	1	1	VQE (HEA)	Lehmann (excited-VQE)	No	SC + Ions (exp)	No self-consistency
[65]	1	1	Exact and VQE	Hadamard (Trotter)	Yes	No	Variational compression.
[66]	1	1	VQE (adhoc)	Hadamard (Trotter)	No	SC (exp)	
[67]	1	1	VQE (UCC)	N/A	No	SC (exp)	Periodic Anderson model. Parity mapping
[68]	1	1	VQE (adhoc)	Hadamard (fast-forwarding)	No	SC (exp)	
[40]	2	2	VQE (adhoc)	N/A	Yes	No	RISB.
[69]	1	3	VQE (HEA)	Krylov variational algorithm	No	No	QSGW+DMFT La_2CuO_4
[70]	1	7	Quantum subspace expansion	Quantum subspace expansion	No	No	
[71]	1	1	VQE (symmetric HEA)	Lehmann (excited-VQE)	Yes	SC (exp)	
[72]	3	17	Matrix product states	Quantum subspace expansion	No	No	$SrVO_3$, chain geometry.
[73]	1	1	VQE ([66])	Hadamard (Trotter)	No	No	
[74]	1	6	VQE (HEA, classical optim.)	Lehmann (qEOM)	No	SC (exp)	$Ca_2CuO_2Cl_2$, chain geometry
[75]	1	6	VQE (sym.-preserving ansatz)	Krylov variational algorithm	No	No	

Table 1: *Summary of early implementations of impurity solvers with quantum computers, sorted in chronological order. N_{c}: number of impurities. N_{b}: (maximum) number of bath sites. SC stands for superconducting qubits. All DMFT computations except those on materials are done on the Bethe lattice. The encoding is always Jordan-Wigner, unless otherwise stated.*

the best classical methods: larger bath sizes mean that larger evolution times, and thus larger circuits, are needed, at least with the Hamiltonian representation of the impurity model we worked with so far.

The bath size of a Hamiltonian-based representation of the AIM of DMFT will likely remain a major issue. Large bath sizes (many Dirac delta peaks, as in (12)) are needed to represent the spectral features of the hybridization function, and these in turn entail large numbers of qubits, and thus large circuits. From a nonequilibrium perspective, one can also argue that reaching long times requires large baths to avoid finite-size effects. In principle, the fact that the bath is noninteracting could be exploited to simplify circuits or perhaps reduce the number of qubits, but this has not been attempted yet, to the best of our knowledge—other than switching from DMFT to simpler embedding techniques like RISB and DMET, which can be seen as well-defined prescriptions for truncating the bath, albeit with access only to one-particle reduced density matrices (as opposed to frequency-dependent Green functions).

An alternative, promising route is to use another representation of the impurity model. For instance, open-system representations of the impurity model have been proposed [82–84] that use a dissipative bath instead of a noninteracting closed bath: each bath site can exchange electrons with an environment. The rate of dissipation $\varLambda$ (as well as the usual other hybridization parameters) can be fixed by matching the hybridization function not with a sum of Dirac peaks (Eq. (12)) but of Lorentzians of width $\varLambda$

$$\varDelta^{\mathrm{R}}(\omega) = \sum_k \frac{V_k^2}{\omega + i\varLambda - \varepsilon_k}. \tag{64}$$

On a classical processor, this allows to use fewer peaks to fit the same hybridization function (with many possible sophistications of the open-system representation [85]), and then, using a master equation-based solver, compute the corresponding Green function. Alternatively, one could use a noisy quantum processor to perform the corresponding dissipative time evolution [86]. This way, the natural decoherence of the processor could, at least partly, be used to mimic the physics of impurity electrons in solids.

This open-system representation may also solve a deeper issue of quantum algorithms for impurity models, namely the preparation of a ground (or low-energy) state. Working with dissipative models changes the perspective to preparing the steady state of an open quantum system... In such a context, more noise could help reach this steady state faster.

References

[1] R.P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982)

[2] J. Hubbard, Proc. Roy. Soc. A **276**, 238 (1963)

[3] A. Georges, G. Kotliar, W. Krauth, and M.J. Rozenberg, Rev. Mod. Phys. **68**, 13 (1996)

[4] G. Knizia and G.K.-L. Chan, Phys. Rev. Lett. **109**, 186404 (2012)

[5] F. Lechermann, A. Georges, G. Kotliar, and O. Parcollet, Phys. Rev. B **76**, 155102 (2007)

[6] T. Ayral, T.-H. Lee, and G. Kotliar, Phys. Rev. B **96**, 235139 (2017)

[7] K. Held: In E. Pavarini, E. Koch, D. Vollhardt, and A.I. Lichtenstein (Eds.): *Autumn School on Correlated Electrons. DMFT at 25: Infinite Dimensions* (Forschungszentrum Julich, 2014), Vol. 4, chap. 10

[8] T. Ayral and O. Parcollet, Phys. Rev. B **94**, 075159 (2016)

[9] P.W. Anderson, Phys. Rev. **124**, 41 (1961)

[10] E. Koch: In E. Pavarini, E. Koch, D. Vollhardt, and A. Lichtenstein (Eds.): *The LDA+DMFT approach to strongly correlated materials* (Forschungszentrum Jülich, 2011), Vol. 1, chap. 8

[11] U. Schollwöck, Ann. Phys. **326**, 96 (2011)

[12] S. Paeckel, T. Köhler, A. Swoboda, S.R. Manmana, U. Schollwöck, and C. Hubig, Ann. Phys. **411**, 167998 (2019)

[13] T. Ayral, P. Besserve, D. Lacroix, and E.A. Ruiz Guzman, Europ. Phys. J. A **59**, 227 (2023)

[14] A.M. Childs, Y. Su, M.C. Tran, N. Wiebe, and S. Zhu, Phys. Rev. X **11**, 011020 (2021)

[15] G.H. Low and I.L. Chuang, Quantum **3**, 163 (2019)

[16] S.B. Bravyi and A.Y. Kitaev, Ann. Phys. **298**, 210 (2002)

[17] V. Havlíček, M. Troyer, and J.D. Whitfield, Phys. Rev. A **95**, 032332 (2017)

[18] F. Verstraete and J.I. Cirac, J. Stat. Mech.: Theory Exp. **2005**, P09012 (2005)

[19] C. Derby and J. Klassen, Phys. Rev. B **104**, 035118 (2020)

[20] M.G. Algaba, P.V. Sriluckshmy, M. Leib, and F. Simkovic, arXiv:2302.01862

[21] T. Albash and D.A. Lidar, Rev. Mod. Phys. **90**, 015002 (2018)

[22] D.W. Berry, G. Ahokas, R. Cleve, and B.C. Sanders, Commun. Comp. Phys. **270** 359 (2007)

[23] D. Guéry-Odelin, S. Mart, and J.G. Muga, Rev. Mod. Phys. **91** 045001 (2019)

[24] J. Kempe, A. Kitaev, and O. Regev in K. Lodaya, M. Mahajan: *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science* (Springer, Heidelberg, 2004), pp. 372–383

[25] N. Schuch and F. Verstraete, Nat. Phys. **5**, 732 (2009)

[26] S. Bravyi and D. Gosset, Commun. Math. Phys. **356**, 451 (2017)

[27] M. Erakovic, F. Witteveen, D. Harley, J. Günther, M. Bensberg, O.R. Meitei, M. Cho, T. Van Voorhis, M. Reiher, and M. Christandl, PRX Life **3**, 013003 (2025)

[28] G. Ortiz, J.E. Gubernatis, E. Knill, and R. Laflamme, Phys. Rev. A **64**, 022319 (2001)

[29] D. Wecker, M.B. Hastings, N. Wiebe, B.K. Clark, C. Nayak, and M. Troyer, Phys. Rev. A **92**, 062318 (2015)

[30] A.Y. Kitaev, arXiv:quant-ph/9511026

[31] T. Ayral, Photoniques **131** 66 (2025)

[32] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Proc. Roy. Soc. A **454**, 339 (1998)

[33] Y. Tong, D. An, N. Wiebe, and L. Lin, arXiv:2008.13295

[34] J. Preskill, Quantum **2**, 79 (2018)

[35] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P.J. Love, A. Aspuru-Guzik, and J.L. O'Brien, Nat. Commun. **5**, 4213 (2013)

[36] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G.H. Booth, and J. Tennyson, Phys. Rep. **986**, 1 (2022)

[37] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J.M. Chow, and J.M. Gambetta, Nature **549**, 242 (2017)

[38] D. Wecker, M.B. Hastings, and M. Troyer, Phys. Rev. A **92**, 042303 (2015)

[39] P.-L. Dallaire-Demers, J. Romero, L. Veis, S. Sim, and A. Aspuru-Guzik, arXiv:1801.01053

[40] P. Besserve and T. Ayral, Phys. Rev. B **105**, 115108 (2022)

[41] P. Besserve, M. Ferrero, and T. Ayral, arXiv:2406.14170

[42] P. Besserve: *Quantum-classical hybrid algorithms for quantum many-body physics* (PhD Thesis, École Poytechnique, 2023)

[43] H.R. Grimsley, S.E. Economou, E. Barnes, N.J. Mayhall, Nat. Commun. **10**, 3007 (2019)

[44] H.L. Tang, V. Shkolnikov, G.S. Barron, H.R. Grimsley, N.J. Mayhall, E. Barnes, and S.E. Economou, PRX Quantum **2**, 020310 (2021)

[45] N.C. Rubin, R. Babbush, and J. McClean, New J. Phys. **20**, 053020 (2018)

[46] A. Arrasmith, L. Cincio, R.D. Somma, and P.J. Coles, arXiv2004.06252

[47] E. Koridon, S. Yalouz, B. Senjean, F. Buda, T.E. O'Brien, and L. Visscher, Phys. Rev. Res. **3**, 033127 (2021)

[48] H.-Y. Huang, R. Kueng, and J. Preskill, Nat. Phys. **16**, 1050 (2020)

[49] K. Wan, W.J. Huggins, J. Lee, and R. Babbush, Commun. Math. Phys. **404**, 629 (2023)

[50] G. Atilim, B.A. Pearlmutter, A.A. Radul, and J.M. Siskind, J. Machine Learning Res. **18**, 1 (2017)

[51] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum **4**, 269 (2020)

[52] J.R. McClean, S. Boixo, V.N. Smelyanskiy, R. Babbush, and H. Neven, Nat. Commun. **9**, 4812 (2018)

[53] M. Ragone, B.N. Bakalov, F. Sauvage, A.F. Kemper, C. Ortiz Marrero, M. Larocca, and M. Cerezo, Nat. Commun. **15**, 7172 (2024)

[54] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P.J. Coles, L. Cincio, J.R. McClean, Z. Holmes, and M. Cerezo, arXiv:2405.00781

[55] M. Plazanet and T. Ayral, arXiv:2410.15129

[56] S. Endo, I. Kurata, and Y.O. Nakagawa, arXiv:1909.12250

[57] K. Mitarai and K. Fujii, Phys. Rev. Res. **1**, 013006 (2019)

[58] S. McArdle, A. Mayorov, X. Shan, S. Benjamin, and X. Yuan, Chem. Sci. **10**, 5725 (2019)

[59] S. Endo, S.C. Benjamin, and Y. Li, Phys. Rev. X **8**, 031027 (2018)

[60] K. Temme, S. Bravyi, and J.M. Gambetta, Phys. Rev. Lett. **119**, 180509 (2017)

[61] J.M. Kreula, S.R. Clark, and D. Jaksch, Sci. Rep. **6**, 32940 (2016)

[62] B. Bauer, D. Wecker, A.J. Millis, M.B. Hastings, and M. Troyer, Phys. Rev. X **6**, 031045 (2016)

[63] J.M. Kreula, L. García-Álvarez, L. Lamata, S.R. Clark, E. Solano, and D. Jaksch, EPJ Quantum Technol. **3**, 11 (2016)

[64] I. Rungger, N. Fitzpatrick, H. Chen, C.H. Alderete, H. Apel, A. Cowtan, A. Patterson, D.M. Ramo, Y. Zhu, N.H. Nguyen, E. Grant, S. Chretien, L. Wossnig, N.M. Linke, and R. Duncan, arXiv:1910.04735

[65] B. Jaderberg, A. Agarwal, K. Leonhardt, M. Kiffner, and D. Jaksch, Quantum Sci. Technol. **5**, 034015 (2020)

[66] T. Keen, T. Maier, S. Johnston, and P. Lougovski, Quantum Sci. Technol. **5**, 035001 (2020)

[67] Y. Yao, F. Zhang, C.-Z. Wang, K.-M. Ho, and P.P. Orth, Phys. Rev. Res. **3**, 013184 (2021)

[68] T. Steckmann, T. Keen, E. Kökcü, A.F. Kemper, E.F. Dumitrescu, and Y. Wang, arXiv:2112.05688

[69] F. Jamet, A. Agarwal, C. Lupo, D.E. Browne, C. Weber, and I. Rungger, arXiv:2105.13298

[70] F. Jamet, A. Agarwal, and I. Rungger, arXiv:2205.00094

[71] J. Ehrlich, D. Urban, and C. Elsässer, arXiv:2311.10402

[72] F. Jamet, L.P. Lindoy, Y. Rath, C. Lenihan, A. Agarwal, E. Fontana, F. Simkovic, B.A. Martin, and I. Rungger, arXiv:2304.06587

[73] A. Baul, H.F. Fotso, H. Terletska, J. Moreno, and K.-m. Tam, arXiv:2308.01392

[74] J. Selisko, M. Amsler, C. Wever, Y. Kawashima, G. Samsonidze, R.U. Haq, F. Tacchino, I. Tavernelli, and T. Eckl, arXiv:2404.09527

[75] E.B. Jones, C.J. Winkleblack, C. Campbell, C. Rotello, E.D. Dahl, M. Reynolds, P. Graf, and W. Jones, arXiv:2405.15069

[76] M. Potthoff, Phys. Rev. B **64**, 165114 (2001)

[77] J. Yu, J.R. Moreno, J.T. Iosue, L. Bertels, D. Claudino, B. Fuller, P. Groszkowski, T.S. Humble, P. Jurcevic, W. Kirby, T.A. Maier, M. Motta, B. Pokharel, A. Seif, A. Shehata, K.J. Sung, M.C. Tran, V. Tripathi, A. Mezzacapo, and K. Sharma, arXiv:2501.09702

[78] N.C. Rubin, arXiv:1610.06910

[79] H. Ma, M. Govoni, and G. Galli, npj Comp. Mater. **6**, 85 (2020)

[80] C. Cao, J. Sun, X. Yuan, H.-s. Hu, H.Q. Pham, and D. Lv, npj Comp. Mater. **9**, 78 (2023)

[81] A. Michel, L. Henriet, C. Domain, A. Browaeys, and T. Ayral, Phys. Rev. B **109**, 174409 (2024)

[82] E. Arrigoni, M. Knap, W.V.D. Linden, and W. Von Der Linden, Phys. Rev. Lett. **110**, 1 (2012)

[83] A. Dorda, M. Sorantin, W. von der Linden, and E. Arrigoni, New J. Phys. **19**, 063005 (2017)

[84] F. Schwarz, M. Goldstein, A. Dorda, E. Arrigoni, A. Weichselbaum, and J. von Delft, Phys. Rev. B **94**, 155142 (2016)

[85] Z. Huang, G. Park, G.K. Chan, and L. Lin, arXiv:2506.10308

[86] C. Bertrand, P. Besserve, M. Ferrero, and T. Ayral, Phys. Rev. B **111**, 245159 (2024)