Spectral Functions with Qiskit
Scientific Report

Diogo Cruz, Duarte Magano, Sagar Pratapsi

June 4, 2021

1 The Spectral Function of Quantum Many-Body Systems

The Green's function is a central quantity in the study of quantum many-body systems. We consider fermionic systems described by a Hamiltonian H composed of fermionic creation and annihilation operators c_a and $c_a^\dagger$, where a is a label of the fermionic mode (usually, the lattice site and spin). Let $|G\rangle$ denote the system's ground state. The zero-temperature retarded Green's function is defined as

$$\mathrm{G}^{\mathcal{R}}_{\mathrm{ab}}(t) := -i\Theta(t) \langle G|\{e^{iHt}c_a e^{-iHt}, c_a^\dagger\}|G\rangle . \tag{1.1}$$

Once the Green's function is known, it becomes easy to compute several physically-relevant observables. For example, the spectral function $\mathrm{A_a}$, which is the focus of this project, can be obtained by taking the imaginary part of the Fourier transform of the Green's function

$$\mathrm{A_a}(w) = -\frac{1}{\pi}\,\mathrm{Im}\,\tilde{\mathrm{G}}^{\mathcal{R}}_{\mathrm{aa}}(w), \tag{1.2}$$

$$\tilde{\mathrm{G}}^{\mathcal{R}}_{\mathrm{aa}}(w) := \int_{-\infty}^{+\infty} \mathrm{d}t\, e^{i(\omega+i\eta)t}\, \mathrm{G}^{\mathcal{R}}_{\mathrm{aa}}(t). \tag{1.3}$$

Unfortunately, calculating the Green's function of large strongly-correlated systems is notoriously hard. As such, quantum algorithms have been put forward for this task [1, 2, 3, 4], essentially relying on quantum simulation or quantum phase estimation.

Very recently, we have witnessed attempts at carrying these ideas to real NISQ devices [5, 6]. These are based on variational algorithms and can be classified into two categories. One is based on the Lehmann's representation of the Green's function function. The idea is to calculate excited states and the transition amplitudes of appropriate operators using generalizations of the variational quantum eigensolver (VQE) [7]. This approach is also followed by [8]. The second one uses variational quantum simulation algorithms to compute the Green's function in real time, and it is the one the we work with here. One disadvantage of this strategy is that, in order to obtain the spectral function, one first has to perform a Fourier transform on the real-time Green's function into the frequency domain, which requires evaluating $\mathrm{G}^{\mathcal{R}}_{\mathrm{aa}}(t)$ for numerous times t.

In our project, we addressed the following questions:

1. Recently, it has been shown that it is possible to variationally approximate a diagonalization of a short-time simulation and enable a longer-time simulation using a constant number of gates, effectively fast-forwarding the Hamiltonian evolution. This idea has been called variational fast forwarding (VFF) [9]. As far as we know, this algorithm has not been applied in the context of Green's functions. So, we ask: **can VFF be useful to compute Green's functions with NISQ computers?**

2. It is well-known that for systems that are not strongly-interacting the spectral function has a peak-like structure – in the limit of no interactions it is a sum of Dirac deltas. In those cases, the spectral function is said to be compressible. Then, the theory of compressed sensing [10] tells us that it is possible to reconstruct the spectral function from fewer samples of the real-time Green's function than what is required by the Nyquist-Shannon theorem. We test this idea: **can compressed sensing help reconstruct the Spectral Function?**

2 The Model

In this project, we focused on the single-impurity Anderson model (SIAM) with one fermionic bath site (at half-filling). The Hamiltonian reads

$$H = U\left(n_{0,\uparrow}n_{0,\downarrow} - \frac{1}{2}n_{0,\uparrow} - \frac{1}{2}n_{0,\downarrow}\right) + V\left(c^\dagger_{0,\uparrow}c_{1,\uparrow} + c^\dagger_{1,\uparrow}c_{0,\uparrow} + c^\dagger_{0,\downarrow}c_{1,\downarrow} + c^\dagger_{1,\downarrow}c_{0,\downarrow}\right), \tag{2.1}$$

where U and V are coupling constants and the subscripts 0/1 indicate the impurity/bath sites and ↑/↓ indicate up-/down spins, respectively. Note that the Hilbert space of this system can be mapped into that of four qubits. This model has been widely used in the context of quantum simulation due to its role in dynamical mean-field theory (DMFT) and the relation with the Hubbard model [11]. Here we'll focus on the parameters $U = 4$ and $V = 0.745356$, which are most suited to this setting [8]. To implement this model on quantum computers, the fermionic modes need to be mapped onto spin operators, which can be achieved via the Jordan-Wigner transform. The mapped hamiltonian reads

$$H = \frac{U}{4}Z_1Z_3 + \frac{V}{2}\left(X_1X_2 + Y_1Y_2 + X_3X_4 + Y_3Y_4\right), \tag{2.2}$$

where the subscripts indicate the qubit index. [2] shows that the Green's function can be written as

$$G(t) = \Theta(t)\frac{\mathrm{Re}\left[\langle Y_1U^\dagger(t)X_1U(t)\rangle\right] - \mathrm{Re}\left[\langle X_1U^\dagger(t)Y_1U(t)\rangle\right]}{2} - i\Theta(t)\frac{\mathrm{Re}\left[\langle X_1U^\dagger(t)X_1U(t)\rangle\right] + \mathrm{Re}\left[\langle Y_1U^\dagger(t)Y_1U(t)\rangle\right]}{2}, \tag{2.3}$$

and each of the terms in this expression can be obtained via a single-qubit interferometry scheme like the one presented in the circuit of Figure 1.

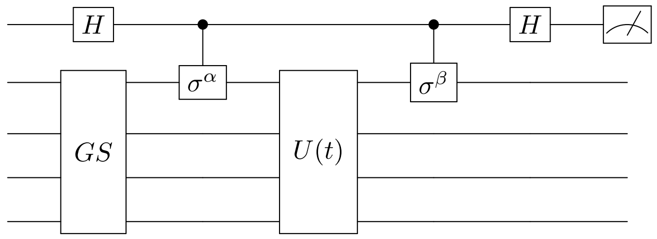

Figure 1: Quantum circuit to measure $\text{Re}\left[\langle \sigma^\alpha U^\dagger(t) \sigma^\beta U(t) \rangle\right]$, where $\sigma^{\alpha/\beta}$ can either mean an X or Y Pauli gate. The GS operator denotes the ground state preparation circuit and $U(t)$ means the time evolution operator e^{-iHt}.

3 Computing the Green's Function

Confronting the circuit of fig. 1, we see that there are two main routines involved in computing the Green's function: preparing the ground state and time-evolving the system. We describe how we have approached the two.

3.1 Ground State Preparation - VQE

We have used VQE to successfully prepare the ground state of the SIAM hamiltonian. We used a R_y ansatz with linear CNOT entanglement. In Figure 2 we show how the (in)fidelity of the prepared state with respect to the exact (analytically calculated) solution evolved with the number of CNOT layers in the ansatz and the training iteration.

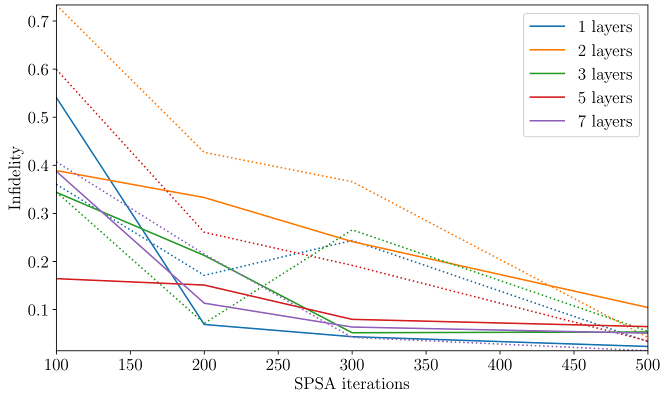

Figure 2: Convergence of the VQE algorithm, with the SPSA minimizer, to the ground state of the SIAM model, using the R_y ansatz with linear CNOT entanglement, for different numbers of CNOT layers. Both the cases of simulated realistic noise (solid lines) and noise-free (dashed) were considered. As expected, VQE converges to the desired state after a reasonable number of iterations. Few layers, and so small circuits, are sufficient for a good convergence.

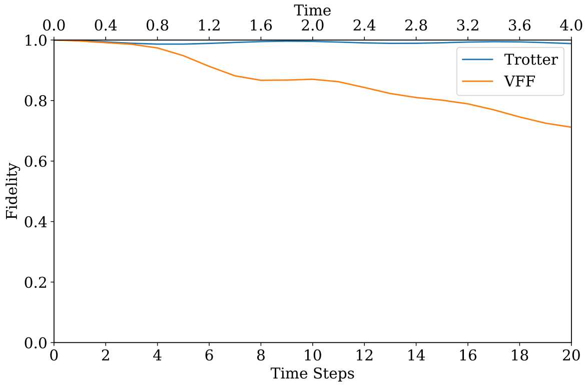

Figure 3: Fidelity of the evolved state, using a Lie-Trotter expansion (blue) and VFF (orange). We observe that VFF manages to maintain a high fidelity similar to that for the Lie-Trotter expansion, for a reasonable number of steps. Although it is not sufficient to directly apply the Fourier transform, it may be possible to extract the Spectral Function using compressed.

3.2 Time Evolution - VFF

The VFF algorithm [12] assumes that we are able to approximate $e^{-iH\Delta t}$ for a short time Δt with, say, a single-step Lie-Trotter expansion $U(\Delta t)$. We would then like to evolve the system for longer times, but deeper quantum circuits are unavailable due to hardware limitations. The idea is to variationally approximate $U(\delta t)$ by compiling it to a unitary of the form $W(\theta)D(\alpha)W(\theta)^\dagger$, where $D(\alpha)$ is a parametrized diagonal unitary encoding the eigenvalues of H and $W(\theta)$ is a parameterized unitary with the corresponding eigenvectors. Then, we approximate $e^{-iHN\Delta t} \approx W(\theta)D(N\alpha)W(\theta)^\dagger$. In fig. 3, we show how the fixed state VFF algorithm [13] (a variant of the original algorithm) permits attaining good fidelities for long evolution times.

4 Reconstructing the Spectral Function with Compressed Sensing

In practice, in the NISQ era, we are generally limited to compute the Green's function to small time values. If we use trotterization-based methods, the number of gates becomes prohibitively large for high t. We may circumvent this problem by using VFF, but to train its ansatz requires us to use a Trotter expansion that has negligible error for the time values we wish to obtain. Therefore, we must still stick with low t, to have a manageable circuit size for the VFF training.

We will then obtain a set of $G(t)$ values $\{G(t_i) + \epsilon_i\}$ for some choice of time values $\{t_i\}$. Each result will have some associated small error ϵ_i, as a result of our circuit only approximating $G(t)$, using noisy gates, and stochastic noise from the limited number of measurements.

To convert $G(t)$ into $G(\omega)$ (and $A(\omega)$), we would like to perform a Fourier transform. If we wished to have a small frequency spacing $\Delta\omega$, to obtain a precise spectral function, we would need time samples up to time $T = 2\pi/\Delta\omega \gg 1$, to be able to use the Discrete Fourier Transform (DFT).

Such high time values are simply not computable in practice, which is why the common approach in the literature has been to find all the Hamiltonian's eigenvalues and compute $G(\omega)$ by using the Lehmann representation [2, 8]. Unfortunately, this approach is not scalable, as the number of eigenvalues scales exponentially with the number of qubits/sites.

There is an alternative that, so far, has never been explored in the literature. If we know that the spectral function is sparse - a sum of Dirac deltas - or compressible, which is always the case for finite-dimensional Hamiltonians, then we may employ compressed sensing. Given a transformation matrix $\mathbf{A} \in \mathbb{R}^{m \times n}, m \ll n$, and a measurement vector $\mathbf{t}$, it consists of finding the frequency vector $\boldsymbol{\omega}$ such that

$$\min_{\boldsymbol{\omega} \in \mathbb{R}^n} \|\boldsymbol{\omega}\|_p, \qquad \text{subject to } \mathbf{t} = \mathbf{A}\boldsymbol{\omega} \tag{4.1}$$

where $p = 0$. In practice, $p = 1$ is used. Using the convention of [14], we call this latter variant `'EL1'`. In the real world case where our measurement vector $\mathbf{t}$ has some error, we may use instead

$$\texttt{'NL1'}: \quad \min_{\boldsymbol{\omega} \in \mathbb{R}^n} \|\boldsymbol{\omega}\|_1, \quad \text{subject to } \|\mathbf{t} - \mathbf{A}\boldsymbol{\omega}\|_2^2 \leq \epsilon \tag{4.2}$$

$$\texttt{'L1'}: \quad \min_{\boldsymbol{\omega} \in \mathbb{R}^n} \|\mathbf{t} - \mathbf{A}\boldsymbol{\omega}\|_2^2 + \lambda \|\boldsymbol{\omega}\|_1 \tag{4.3}$$

choosing some ϵ and λ, respectively. Smoothed ℓ_0 minimization [15] is also available as `'SL0'`. For our use case, we are interested in having a transformation matrix $\mathbf{A}$ that acts as the inverse Fourier transform. These results can be seen in fig. 4, for method `'L1'`. We started with the exact implementation of the ground state (using `QuantumCircuit().initialize()`) and ran the circuit using `qasm_simulator`.

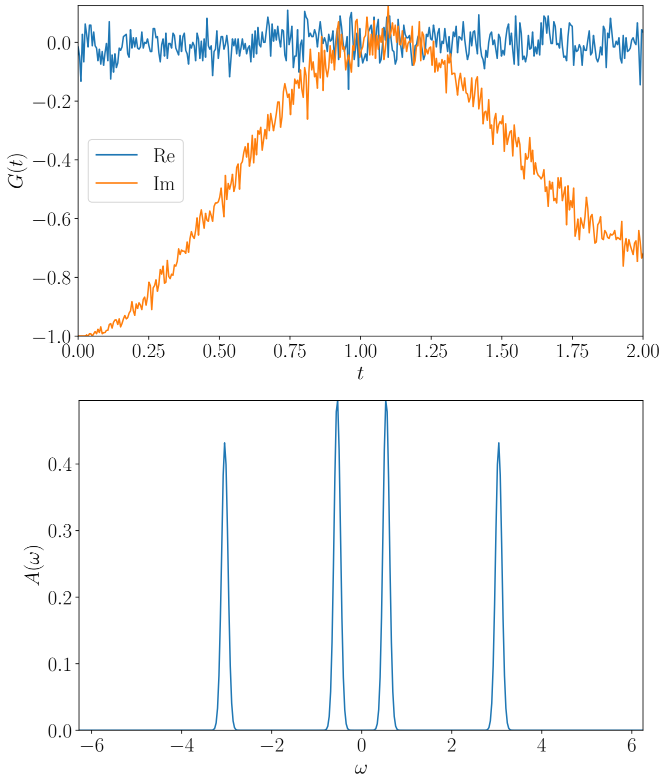

Figure 4: Green's function $G(t)$ (left) and spectral function $A(\omega)$ (right) for the SIAM model, for site 1, after regularization. With $\Delta\omega = 0.0314$, we would have needed to compute the Green's function up to $T = 200$ if we wished to use the Discrete Fourier Transform directly. By using compressed sensing, only time samples for $T < 2$ were necessary, each with high stochastic error, as only 256 shots were taken. This is the time interval where the Suzuki-Trotter expansion with $k = 4, r = 2$ still has negligible error. By allowing us to use a low-order Trotter expansion, compressed sensing enables us to implement a NISQ quantum circuit based on VFF or even trotterization directly.

5 Contributions

The various parts here described were equally divided by the different authors. Sagar Pratapsi developed the code to implement VQE to find the ground state for a general Hamiltonian. Duarte Magano was responsible for the portion of the package that takes this state as input as computes the Green's function $G(t)$, using either an exact, a trotterization-based, or a VFF-based method. Diogo Cruz was then tasked with getting the $G(t)$ data and computing the spectral function, by either using the Fourier transform directly or compressed sensing.

We attempted to apply this method to the Ising model, the Anderson impurity model, and the Hubbard model, and use real hardware. Unofortunately, time constraints meant that we only fully studied the Anderson model, and partially studied the Ising model. As implementing this in real hardware would have required some effort, in optimizing the circuit and adapting it to the device layout, we ended up focusing on using the `statevector_simulator` and `qasm_simulator` to obtain our results. For the initial VQE tests, noisy gate models were also considered.

A walkthrough of the different steps of our method may be found in Jupyter Notebook `Tutorial.ipynb`, in this project's Github page **QuGreen**.

6 Outlook

We have considered two original ways to improve the computation of spectral functions on quantum computers. The first one was to use VFF as a subroutine in the calculation of the real-time Green's function. We have seen that it can execute time evolution for long times with high fidelity in a noiseless scenario (no decoherence time). However, we have not tested how it performs in the context of the natural gate decoherence. The second one was to use compressed sensing to reconstruct the spectral functions with fewer samples of the Green's function. We have shown that it works for signals suffering from high stochastic noise, even when only taking measurements for small times. Consequently, as long as the $G(t)$ circuit (fig. 1) is a good approximation for the measured times, we expect compressed sensing to work reasonably well. Ultimately, we would like to run our approach from beginning to end on IBM's real quantum hardware.

We hope that this tool can be useful for other interested researchers in computing Spectral Functions using quantum hardware, in the NISQ era. In the future, both the Hamiltonians that were not covered in the tutorial, and the other interesting systems may be added directly to our package, so that any user may quickly test various cases. It would also be interesting to determine, more concretely, the parameter range where we may expect VFF and compressed sensing to work, as that would be the regime most amenable to a NISQ implementation in real hardware. We would also wish to run concrete tests in IBM's real hardware, which we expect would add empirical strength to these methods.

References

[1] D. Wecker *et al.*, Physical Review A - Atomic, Molecular, and Optical Physics **92** (2015), 1506.05135.

[2] J. M. Kreula *et al.*, EPJ Quantum Technology **3**, 11 (2016).

[3] B. Bauer, D. Wecker, A. J. Millis, M. B. Hastings, and M. Troyer, Physical Review X **6**, 031045 (2016).

[4] B. Jaderberg, A. Agarwal, K. Leonhardt, M. Kiffner, and D. Jaksch, (2020), 2002.04612.

[5] S. Endo, I. Kurata, and Y. O. Nakagawa, (2019), 1909.12250.

[6] T. Keen, T. Maier, S. Johnston, and P. Lougovski, (2019), 1910.09512.

[7] A. Peruzzo *et al.*, Nature Communications **5**, 4213 (2014), Number: 1 Publisher: Nature Publishing Group.

[8] I. Rungger *et al.*, arXiv:1910.04735 [cond-mat, physics:quant-ph] (2020), arXiv: 1910.04735.

[9] C. Cîrstoiu *et al.*, npj Quantum Information **6**, 1 (2020), Number: 1 Publisher: Nature Publishing Group.

[10] M. Davenport, M. Duarte, Y. Eldar, and G. Kutyniok, Preprint **93** (2012).

[11] A. Georges, G. Kotliar, W. Krauth, and M. J. Rozenberg, Reviews of Modern Physics **68** (1996).

[12] C. Cîrstoiu *et al.*, npj Quantum Information **6**, 82 (2020).

[13] J. Gibbs *et al.*, p. 1 (2021), 2102.04313.

[14] Q. Lyu, Z. Lin, Y. She, and C. Zhang, Neurocomputing **119**, 413 (2013).

[15] H. Mohimani, M. Babaie-Zadeh, I. Gorodnitsky, and C. Jutten, arXiv:1001.5073 [cs, math] (2010), arXiv: 1001.5073.