Unraveling correlated materials' properties with noisy quantum computers:

Natural-orbitalized variational quantum eigensolving

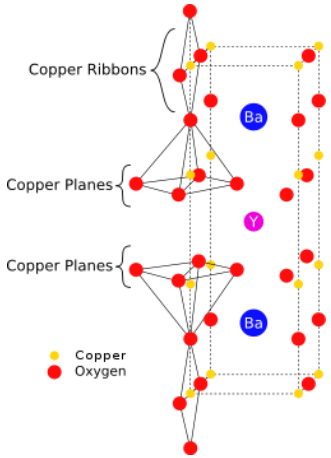
Thomas Ayral
Atos Quantum Laboratory (Les Clayes-sous-Bois)

Journées Prospectives du Réseau Français de Chimie Théorique

November 19, 2021

How to handle strongly correlated materials with a classical computer?

- Holy grail: High-temperature superconductors



- "Spherical cow": the Hubbard model (see David's talk)

$$H = -t \sum_{ij\sigma} c^{+}_{i\sigma} c_{j\sigma} + U \sum_{i} n_{i\uparrow} n_{i\downarrow}$$

- Failure of mean field (Hartree Fock)!

State-of-the-art classical method: Embedding

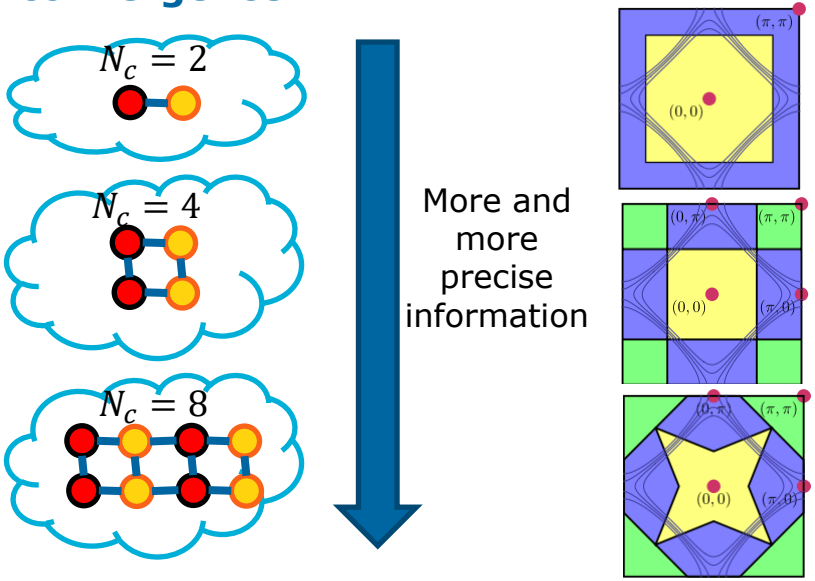
Map lattice problem to local **impurity problem:**

Boil problem down to "quantum quintessence"

Similar to **active space** selection: pick correlated degrees of freedom

Goals… and limitation of embedding methods

Increase N_c (~active space size) to reach convergence



Problems with increasing N_c:

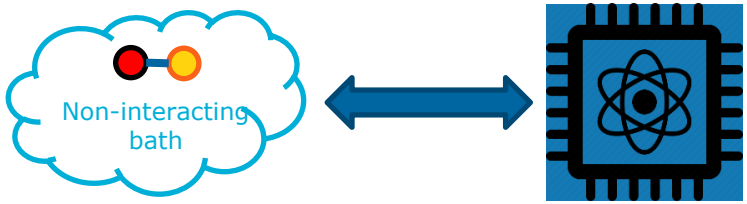
- Exponential size of Hilbert space (if using FCI/exact diagonalization)
- (or) Monte-Carlo sign problem
- (or) too much entanglement for tensor network methods (MPS… see Alberto's talk)

"Control": nothing changes when $N_c \to N_c + 1$

Atos

Solving impurity models with quantum computers: first attempts

To reach larger impurity sizes: use a quantum computer !?



- Early proposals: Bauer '16, Kreula '16

Embedding: Dynamical mean field theory

Impurity model solved with QC!

All the rest is classical

- Key quantity: impurity Green's function

$$G_{ij}(t) = \langle \psi_{GS} | c_i(t) c_j^+(0) | \psi_{GS} \rangle$$

with $|\psi_{GS}\rangle$: ground state of impurity model

Quantum algorithm:

- (classical) Truncate infinite bath (finite #qubits!)
- Prepare ground state $|\psi_{GS}\rangle$
- Time-evolve $|\psi_{GS}\rangle$ via Trotterization to measure $G(t)$

As expected:

- With no or "few" errors, it works: exponential speedup.
  - "few": error levels compatible with quantum error correction.

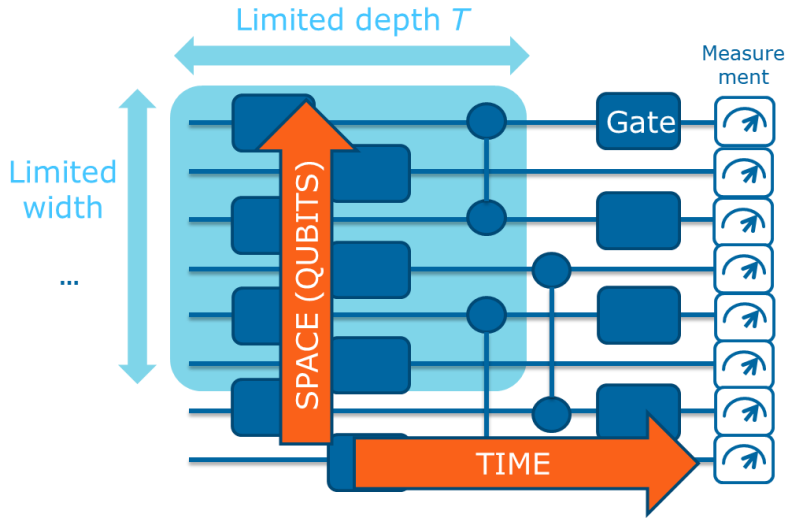
But what with today's QCs?

Atos

NISQ computers… and their issues

Limitations:

- **Few qubits + Short coherence times**



Expected issues:

- Truncate infinite bath
  - ***Few qubits: truncation error?***
- Prepare ground state $|\psi_{GS}\rangle$
  - ***Adiabatic preparation… too long circuit?***
- Time-evolve $|\psi_{GS}\rangle$ via Trotterization to measure $G(t)$
  - ***Implementation of e^{iHt}: too long circuit?***

A hybrid variational approach for shorter circuits: VQE

Variational Quantum Eigensolving, Peruzzo '14

- Prepare ground state $|\psi_{GS}\rangle$
  - ***Adiabatic preparation… too long circuit?***

Find ground state of H:

- **An old method**:

Variational principle: $\min_{\theta}\langle\psi_{\vec{\theta}}|H|\psi_{\vec{\theta}}\rangle \geq E_0$

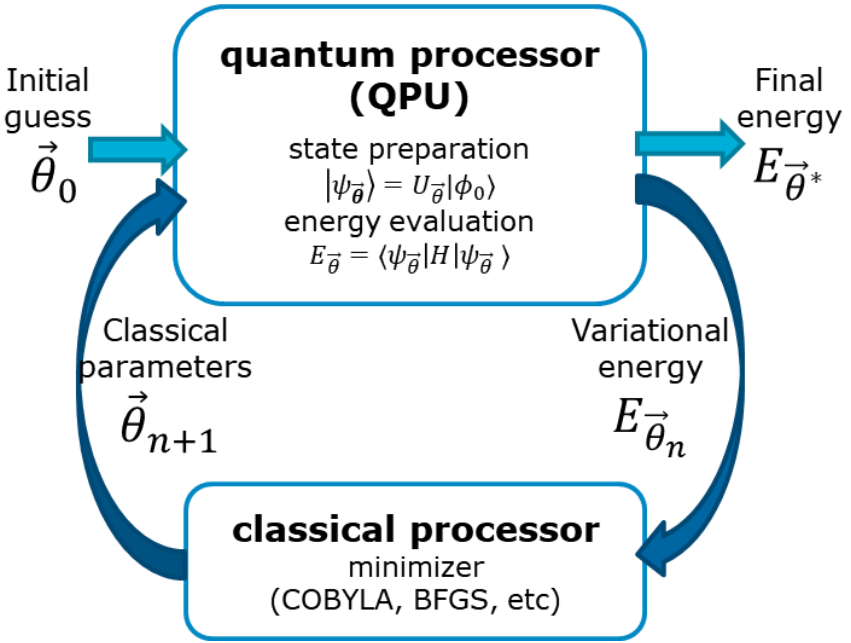
- **A new idea**:

Use QC to compute $\left\langle\psi_{\vec{\theta}}|H|\psi_{\vec{\theta}}\right\rangle$

- **Hybrid!**

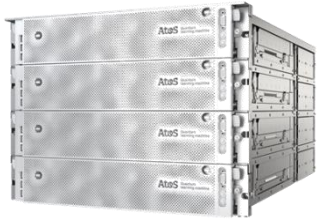
Bonus 1: QPU better at preparing and measuring quantum states

Bonus 2: Can choose ansatz circuit $U_{\vec{\theta}}$ to accomodate QPU constraints



Caveat: no speedup guarantee!

Variational Quantum Algorithms on the Atos QLM



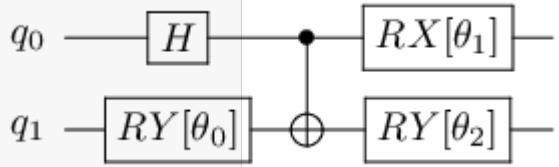
- QLM: powerful stack for variational algorithms
- VQE for the Heisenberg model $H = XX + YY + ZZ$:

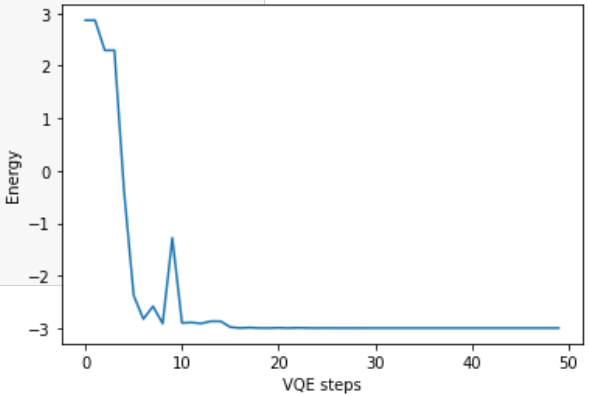
```
H_XY = Observable(2, pauli_terms=[Term(1., term , [0, 1]) for term in ["XX", "YY", "ZZ"]])

prog = Program()
qbits = prog.qalloc(2)
theta = [prog.new_var(float, "\\theta_%s"%i) for i in range(3)]
H(qbits[0])
RY(theta[0])(qbits[1])
CNOT(qbits)
RX(theta[1])(qbits[0])
RY(theta[2])(qbits[1])
circ = prog.to_circ()

optimizer_scipy = ScipyMinimizePlugin(method="COBYLA", x0=[0.4, -0.3, 0.6])
qpu = optimizer_scipy | LinAlg()
job = circ.to_job(job_type="OBS", observable=H_XY)
result = qpu.submit(job)
print("Minimum energy =", result.value)
```

Could be any QPU!






```
Minimum energy = -2.9999999851052666
```

Download+install: myqlm.github.io

Try it yourself!

Download+install: myqlm.github.io

myqlm.github.io/getting_started.html#a-simple-variational-algorithm

Applications BB Jira Wiki myQLM github myQLM doc SynapseS Home Page - Select... slack Jenkins Liste de lecture

myQLM

myQLM-1.4.0

Search docs

BASIC USAGE

Installing myQLM

Getting started

A simple Grover

A simple variational algorithm

Writing quantum circuits

```
from qat.core import Observable, Term
from qat.lang.AQASM import Program, RY, CNOT
from qat.qpus import get_default_qpu
from qat.plugins import ScipyMinimizePlugin

# we instantiate the Hamiltonian we want to approximate the grou
hamiltonian = Observable(nqbits=2, pauli_terms=[Term(1, op, [0, 1

# we construct the variational circuit (ansatz)
prog = Program()
reg = prog.qalloc(2)
theta = [prog.new_var(float, '\\theta_%s'%i) for i in range(2)]
RY(theta[0])(reg[0])
RY(theta[1])(reg[1])
CNOT(reg[0], reg[1])
circ = prog.to_circ()

# construct a (variational) job with the variational circuit and
job = circ.to_job(observable=hamiltonian,
                  nbshots=100)

# we now build a stack that can handle variational jobs
```

Run on an actual QPU!

Download+install: myqlm.github.io

myqlm.github.io/myqlm_specific/qiskit.html

Applications BB Jira Wiki myQLM github myQLM doc SynapseS Home Page - Select... slack Jenkins Liste de lecture

The device library

Plugins

QPUs

Interoperability with other frameworks

Interoperability with myQLM

Qiskit interoperability

PyQuil interoperability

Cirq interoperability

ProjectQ interoperability

OpenQASM Compiler

Combinatorial optimization and QAOA

Connecting to a QPU/Backend

myQLM can be used to connect to a Qiskit Backend. This module is composed of three main classes:

- `BackendToQPU`: Synchronous QPU, capable of running in a Qiskit backend

```
from qat.interop.qiskit import BackendToQPU

# Declare your IBM token
MY_IBM_TOKEN = "..."

# Wrap a Qiskit backend in a QPU
qpu = BackendToQPU(token=MY_IBM_TOKEN, ibmq_backend="ibmq

# Submit a job to IBMQ
result = qpu.submit(job)
```

First attempts on NISQ computers: summary

- Truncate infinite bath
  - ***Few qubits: truncation error?***
- Prepare ground state $|\psi_{GS}\rangle$
  - ***Adiabatic preparation… too long circuit?***
- Time-evolve $|\psi_{GS}\rangle$ via Trotterization to measure $G(t)$
  - ***Implementation of e^{iHt}: too long circuit?***

	Keen '20	Rungger '19	Jaderberg '20	Yao '21
	One bath site ('two site DMFT')	One bath site ('two site DMFT')	One bath site ('two site DMFT')	One bath site (slave boson)
	VQE ('hardware efficient ansatz')	VQE with excited states	VQE (with machine learning techniques)	VQE (unitary coupled cluster ansatz)
	Trotterization.	Via computation of excited states.	Trotterization.	No need for Green's function!
Number N_c of impurities	1 (4 qubits)	1 (4 qubits)	1 (4 qubits)	1 (4 qubits)

Our goal: reach larger impurity models (N_c) with same noise

Atos

Step 1/2: pick the right embedding method

TA, Lee, Kotliar '17

- Several embedding methods on the market:

	Dynamical Mean Field Theory	Rotationally-Invariant Slave Bosons (Gutzwiller)	Density Matrix Embedding Theory
# bath levels	infinite	N_c	N_c
Impurity observable	Green's function $\langle\psi_{GS}\|c_i(t)c_j^+(0)\|\psi_{GS}\rangle$	**1-RDM** $\langle\boldsymbol{\psi_{GS}}\|\boldsymbol{c_i f_j^+}\|\boldsymbol{\psi_{GS}}\rangle$	1-RDM
Final output	Freq-dependent self-energy	**Low-energy self-energy**	Static self-energy

- **Most NISQ-compatible choice: RISB**
  - Well-defined bath truncation
  - Simpler observable
  - Access to quasiparticle renormalization factor

Bath truncation

Green's function…

Keen '20	Rungger '19	Jaderberg '20	Yao '21
One bath site ('two site DMFT')	One bath site ('two site DMFT')	One bath site ('two site DMFT')	One bath site (slave boson)
VQE ('hardware efficient ansatz')	VQE with excited states	VQE (with machine learning techniques)	VQE (unitary coupled cluster ansatz)
Trotterization.	Via computation of excited states.	Trotterization.	No need for Green's function!
1 (4 qubits)	1 (4 qubits)	1 (4 qubits)	1 (4 qubits)

Atos

Step 2/2: pick the right ansatz

Goal: compute $\langle\psi_{GS}|c_i f_j^+|\psi_{GS}\rangle$ with VQE. How to choose the ansatz $|\psi_{\vec{\theta}}\rangle$?

Key requirement: "expressivity"

- **Gold standard: Unitary coupled-cluster (UCC)**

$$|\text{UCCSD}\rangle = e^{T\,-T^+}|HF\rangle$$

with cluster operator

$$T = \sum_{\{i\in occ, a\in virt\}} \theta_a^i\; c_a^+\, c_i + \sum_{\{i>j\in occ, a>b\in virt\}} \theta_{ab}^{ij}\, c_a^+ c_b^+ c_i c_j$$

Issue: too many terms, too deep (long) circuit!

- **A shallower ansatz: the Low-Depth Circuit Ansatz (LDCA)**

Dallaire-Demers et al '20

Key insight:

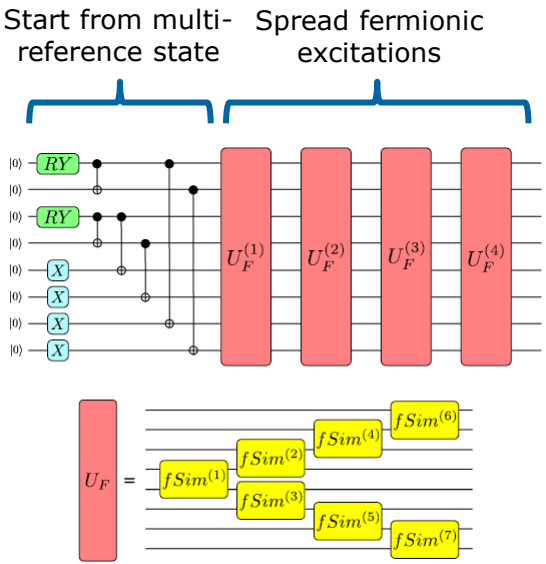
- pick mostly "Gaussian operations" ($c_a^+\, c_i$ terms) that create single-Slater determinant states
- … and few selected non-Gaussian ($c_a^+ c_b^+ c_i c_j$) terms to include correlations (~multi reference)

Issue: still too deep!

Atos

Step 2/2: pick the right ansatz

- **Our ansatz: the Multi-Reference Excitation-Preserving (MREP) ansatz**



- How to read such a circuit?

One line (qubit): one orbital

("Jordan-Wigner" fermion-qubit mapping)

  - **After X gates**: state $|00001111\rangle$: single Slater det.
  - **After MR pattern**: MR state $\alpha|00001111\rangle + \beta|11001100\rangle + \gamma|00110011\rangle + \delta|10101010\rangle$
  - **Action of "fSim" gate**: $|01\rangle$ becomes $u|01\rangle + v|10\rangle$ (and dephase $|11\rangle$)

Keen '20	Rungger '19	Jaderberg '20	Yao '21
One bath site ('two site DMFT')	One bath site ('two site DMFT')	One bath site ('two site DMFT')	One bath site (slave boson)
VQE ('hardware efficient ansatz')	VQE with excited states	VQE (with machine learning techniques)	VQE (unitary coupled cluster ansatz)
Trotterization.	Via computation of excited states.	Trotterization.	No need for Green's function!
1 (4 qubits)	1 (4 qubits)	1 (4 qubits)	1 (4 qubits)

Cf Sugisaki '19: circuits for molecules with diradical character

Intermezzo: Demo

Beyond VQE with the natural orbitalization procedure

3

The failure of plain-vanilla VQE

- RISB embedding with MREP ansatz in VQE
  - Sanity check:
    without noise, works
  - **With noise… no convergence!**
    (too far from true impurity ground state)
- How to overcome this limitation?
  - Key idea: use freedom in **choice of orbital basis**

The role of the orbital basis

- Goal: minimize $\langle\psi_{\vec{\theta}}|H|\psi_{\vec{\theta}}\rangle$.
- $|\psi_{\vec{\theta}}\rangle$ and H: expressed in a given orbital basis. E.g:

$$H = \sum_{p,q} h_{pq} c_p^+ c_q + \frac{1}{2}\sum_{p,q,r,s} h_{pqrs} c_p^+ c_q^+ c_r c_s$$

where c_p^+ creates electron in orbital $\phi_p(r)$.

- We can freely change basis:

$$\tilde{c}_p^+ = \sum_q U_{pq} c_q^+$$

- Equivalent $|\psi_{\vec{\theta}}\rangle$ and H!

- **How to exploit this freedom?**

Although equivalent, different bases:

more or less sparse representations

- E.g state $|11\rangle$ in c_0^+, c_1^+ basis

… becomes…

state $\frac{|00\rangle-|01\rangle+|10\rangle-|11\rangle}{2}$

in rotated basis:

- $\tilde{c}_0^+ = (c_0^+ + c_1^+)/\sqrt{2}$
- $\tilde{c}_1^+ = (c_0^+ - c_1^+)/\sqrt{2}$

- Different overhead to prepare on QC!

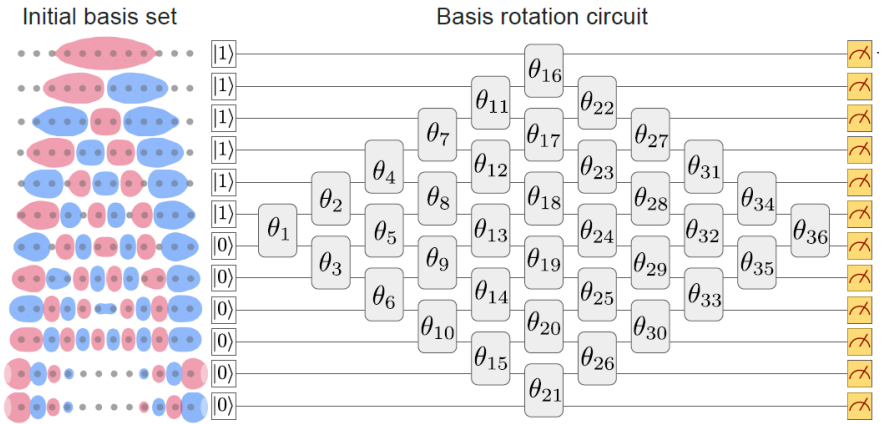
Aside: Hartree-Fock on a quantum computer

Arute et al, 2020

- Goal: rotate to "molecular-orbital" basis

(orbital basis that minimizes energy of a single Slater determinant)

- Circuit:



- Accurate measured energies (not shown here)… w.r.t HF on classical computer…

- But… if rotation to MO basis on classical computer (instead of QC):
  - circuit would have been trivial!

  (a single Slater determinant: only a few X gates!)

- Orbital rotations: easy for classical computers… and potentially harmful on QCs!
  - **Can we go (classically) to the "optimal" orbital basis?**

Optimizing the orbital basis: Natural orbitals

Besserve, TA, 2108.10780

- **Natural orbitals (NOs):**

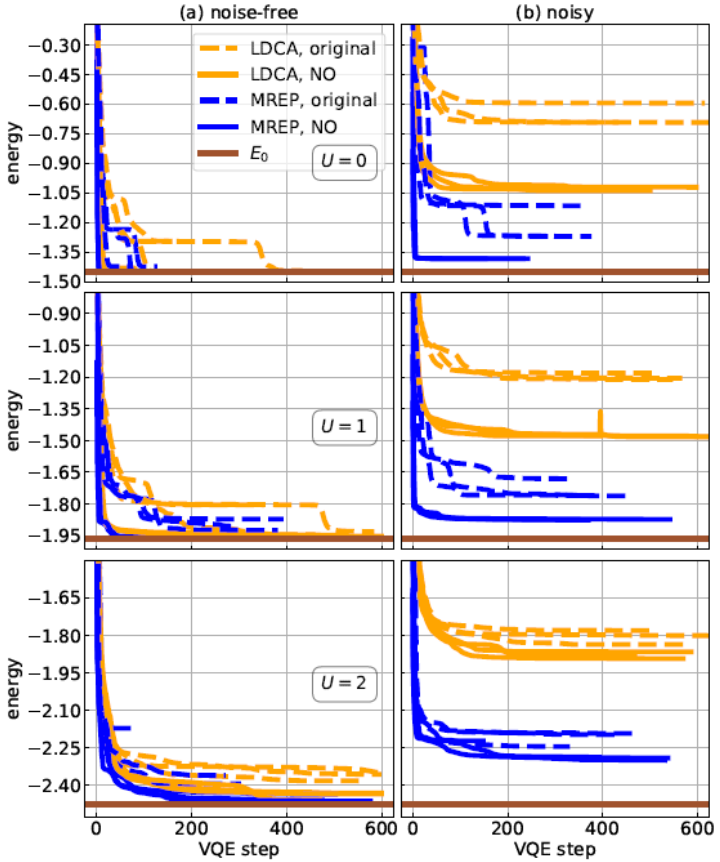
basis that diagonalizes the one-particle density matrix $D_{ij} = \langle \psi_{GS} | c_i^\dagger c_j | \psi_{GS} \rangle$

"Basis in which GS can be written with the least number of Slater determinants"

- Hence, basis in which preparation circuit should be the shortest!

- **Comparison of original vs NO basis**

  NO leads to faster+more accurate convergence

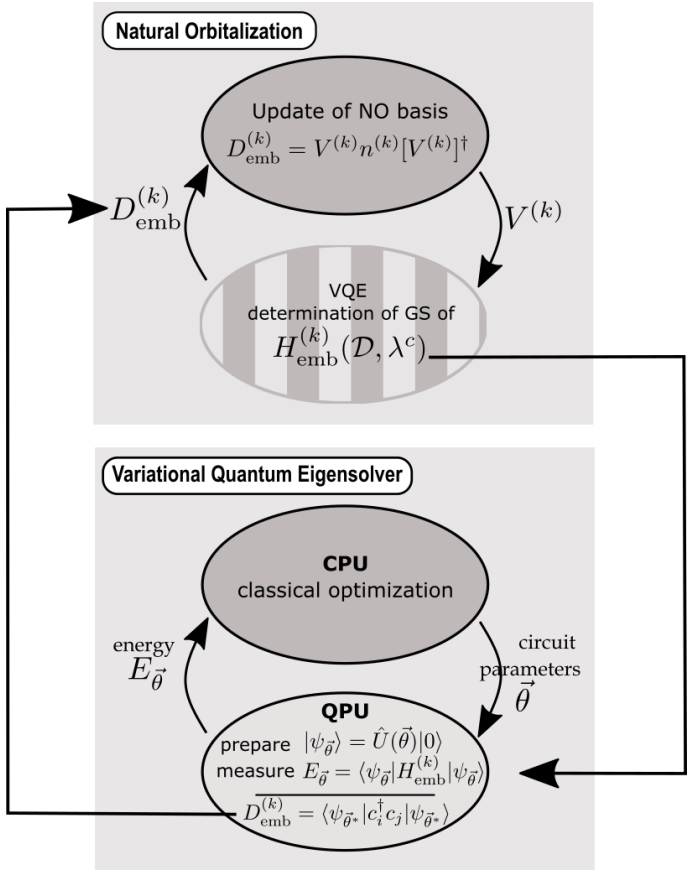


On orbital optimization: see also Saad's talk

In practice: iterative construction

Besserve, TA, 2108.10780

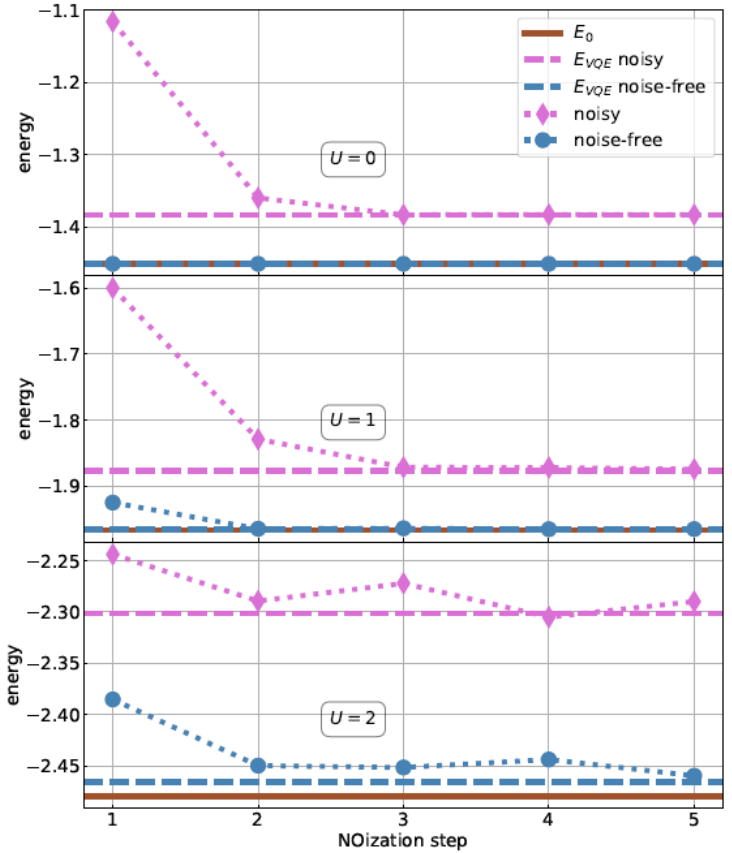
- In practice:
  - the ground state $|\psi_{GS}\rangle$ is unknown
  - so the NO basis is unknown!
- **Iterative procedure: "NOization"**
  - Compute RDM for current approximation
  - Rotate to approximate NO basis
  - Repeat until convergence



NOization: results

Besserve, TA, 2108.10780

- Iterative procedure converges to exact NO energy
- Noise-free case:
  - Ansatz reaches exact energy for $U = 0$ and $U = 1$
  - Good (but not perfect) for $U = 2$
- Noisy case
  - Bias is reduced by NOization



Putting everything together: self-consistent embedding loop

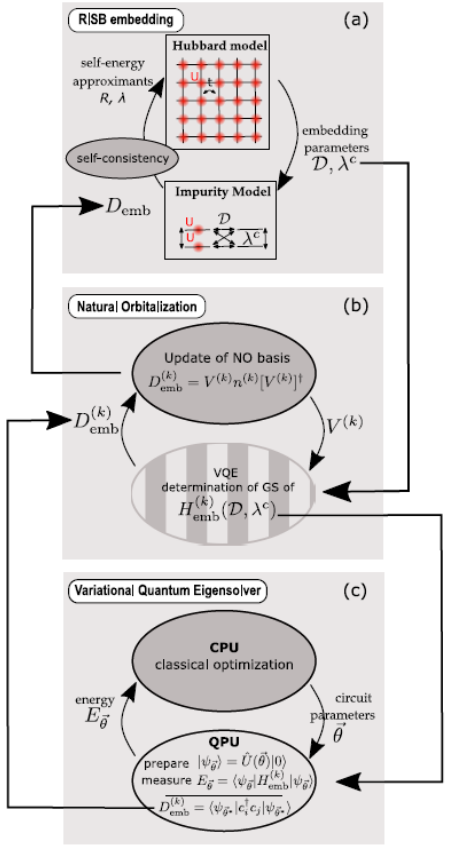
Besserve, TA, 2108.10780

- Half-filled, paramagnetic phase of Hubbard model:

Upon increasing interaction U, expect "Mott transition" (see David's talk)

- Embedding: Rotationally-Invariant Slave Boson:
  - $N_c = 2$ (2 impurities, 2 bath sites: 8 qubits)
  - Solved by minimization procedure
- MREP ansatz

- **Realistic noisy simulations on Atos QLM:**
  - Depolarizing gate noise
  - Noise level to reproduce gate error rates: 0.6% (2-qubit gates), 0.16% (1-qubit gates).

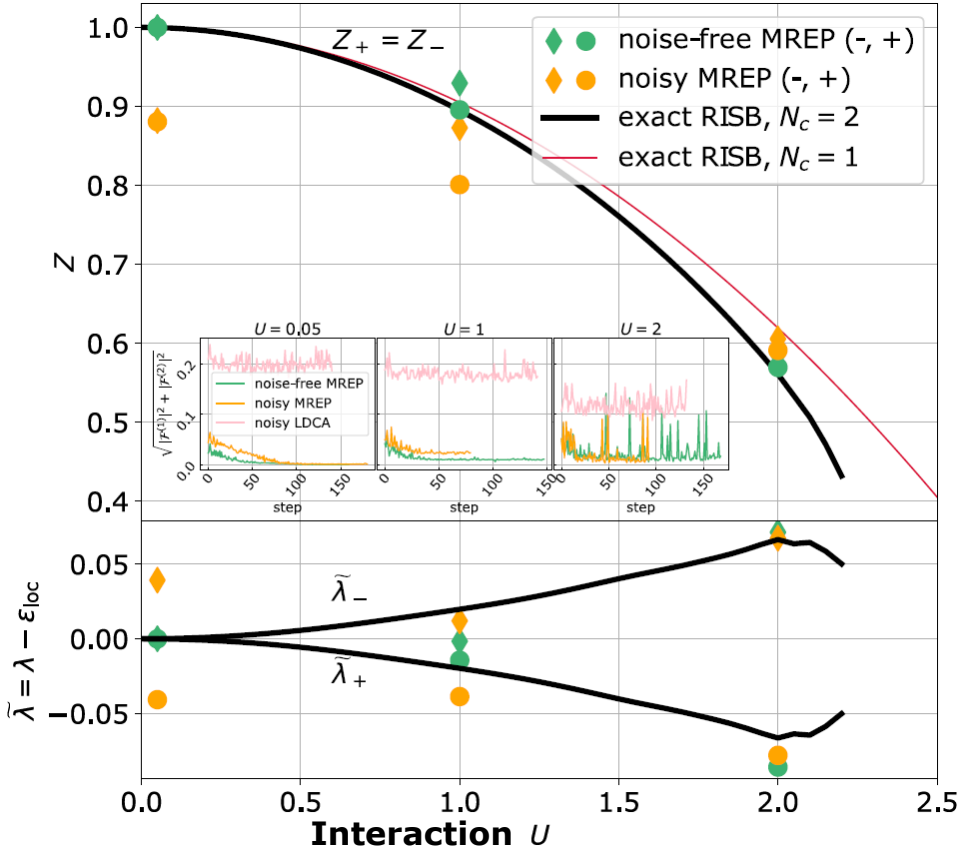


Final results

Besserve, TA, 2108.10780

Quasiparticle weight $Z = R^+R$

Static self-energy λ



$$\Sigma(\omega) = \omega(I - (R^+R)^{-1}) + R^{-1}\lambda(R^+)^{-1}$$

Conclusion

- **Classical preprocessing matters**
  - Embedding
  - NOization
  - VQE

- **Algorithms for NISQ must be tested under noisy conditions**
  - Noisy simulation on QLM

- **NOization:**
  - Here, used for impurity model… behavior for **quantum chemistry** problems?
  - With **analog** quantum simulators?
    (Variational Quantum Simulation, Kokail '19)

Thank you

thomas.ayral@atos.net