Quantum Chemistry on a Quantum Computer

Why?

1. Curiosity

“A Quantum machine may be more efficient at simulating a quantum system than a classical machine.”

Feynman
Quantum Chemistry on a Quantum Computer

Why?

1. Curiosity
2. Difficulty

“The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known, and the difficulty lies only in the fact that application of these laws leads to equations that are too complex to be solved.”

\[ i\hbar \frac{\partial \Psi}{\partial t} = \hat{H} \Psi \]

Dirac
Quantum Chemistry on a Quantum Computer

Why?

1. Curiosity
2. Difficulty
3. Importance
# Quantum Chemistry Programs on CPUs (80 and counting)

<table>
<thead>
<tr>
<th>Package</th>
<th>License</th>
<th>Language</th>
<th>Basis</th>
<th>Periodic</th>
<th>Mol. mech.</th>
<th>Semi-emp.</th>
<th>HF</th>
<th>Post-HF</th>
<th>DFT</th>
<th>GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABINIT</td>
<td>GPL</td>
<td>Fortran</td>
<td>PW</td>
<td>3d</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ACES II</td>
<td>GPL</td>
<td>Fortran</td>
<td>GTO</td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ACES III</td>
<td>GPL</td>
<td>Fortran/C++</td>
<td>GTO</td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ADF</td>
<td>Commercial</td>
<td>Fortran</td>
<td>STO</td>
<td></td>
<td>Any</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Atomix ToolKit (ATK)</td>
<td>Commercial</td>
<td>C++/Python</td>
<td>NAO/EHT</td>
<td>3d</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>BigDFT</td>
<td>GPL</td>
<td>Fortran</td>
<td>Wavelet</td>
<td></td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>CADPAC</td>
<td>Academic</td>
<td>Fortran</td>
<td>GTO</td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>CASINO (OMC)</td>
<td>Academic</td>
<td>Fortran 95</td>
<td>GTO / PW / Spline / Grid / STO</td>
<td>Any</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>CASTEP</td>
<td>Academic (UK) / Commercial</td>
<td>Fortran</td>
<td>PW</td>
<td>3d</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>CFOURS</td>
<td>Academic</td>
<td>Fortran</td>
<td>GTO</td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>COLUMBUS</td>
<td>Academic</td>
<td>Fortran</td>
<td>GTO</td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CONQUEST</td>
<td>Academic</td>
<td>Fortran 90</td>
<td>NAO/Spline</td>
<td>3d</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CP2K</td>
<td>GPL</td>
<td>Fortran 95</td>
<td>Hybrid GTO / PW</td>
<td>3d</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CPMD</td>
<td>Academic</td>
<td>Fortran</td>
<td>PW</td>
<td>3d</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CRYSTAL</td>
<td>Academic (UK) / Commercial</td>
<td>Fortran</td>
<td>GTO</td>
<td></td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DACapo</td>
<td>GPL</td>
<td>Fortran</td>
<td>PW</td>
<td>3d</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DALTON</td>
<td>Academic</td>
<td>Fortran</td>
<td>GTO</td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DFTB+η</td>
<td>Academic / Commercial</td>
<td>Fortran 95</td>
<td>NAO</td>
<td></td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DFT++η</td>
<td>GPL</td>
<td>C++</td>
<td>PW / Wavelet</td>
<td>3d</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DIRAC</td>
<td>Academic</td>
<td>Fortran 77, Fortran 90, C</td>
<td>GTO</td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Dmol3</td>
<td>Commercial</td>
<td>Fortran</td>
<td>NAO</td>
<td></td>
<td>Any</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ELK</td>
<td>GPL</td>
<td>Fortran 95</td>
<td>FP-LAPW</td>
<td>3d</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Empire η</td>
<td>Academic / Commercial</td>
<td>Fortran</td>
<td>Minimal STO</td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>ErgoSCF η</td>
<td>GPL</td>
<td>C++</td>
<td>GTO</td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ERKALE η</td>
<td>GPL</td>
<td>C++</td>
<td>GTO</td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>EXCITING</td>
<td>GPL</td>
<td></td>
<td>FP-LAPW</td>
<td>3d</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>FLEUR η</td>
<td>Academic</td>
<td>Fortran 95</td>
<td>FP-(L)APW+lo</td>
<td>3d, 2d, 1d</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>FHI-aims</td>
<td>Commercial</td>
<td>Fortran</td>
<td>NAO</td>
<td></td>
<td>Any</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>FreeON</td>
<td>GPL</td>
<td>Fortran 95</td>
<td>GTO</td>
<td></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Firefly / PC GAMESS</td>
<td>Academic</td>
<td>Fortran, C, Assembly</td>
<td>GTO</td>
<td></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GAMESS (UK)</td>
<td>Academic (UK) / Commercial</td>
<td>Fortran</td>
<td>GTO</td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GAMESS (US)</td>
<td>Academic</td>
<td>Fortran</td>
<td>GTO</td>
<td></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Gaussian</td>
<td>Commercial</td>
<td>Fortran</td>
<td>GTO</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GPAW η</td>
<td>GPL</td>
<td>Python / C</td>
<td>Grid / NAO / PW</td>
<td>3d</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HILAPW η</td>
<td>Unknown</td>
<td>Unknown</td>
<td>FLAPW</td>
<td>3d</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Jaguar</td>
<td>Commercial</td>
<td>Fortran / C</td>
<td>GTO</td>
<td></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
The problem

Schrödinger Equation

\[ i \frac{\partial \Psi}{\partial t} = \hat{H} \Psi \]

\[ \hat{H} = - \sum_i \frac{1}{2m_i} \nabla_i^2 + \sum_{i>j} \frac{Z_i Z_j}{r_{ij}} \]

\[ \Psi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n, \mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_N, t) \]

Water: \( N = 3 \quad n = 10 \)

Protein: \( N = 10000 \quad n = 50000 \)
The problem

\[ i \frac{\partial \Psi}{\partial t} = \hat{H} \Psi \]

\[ \hat{H} = - \sum_i \frac{1}{2m_i} \nabla_i^2 + \sum_{i>j} \frac{Z_i Z_j}{r_{ij}} \]

\[ \Psi(r_1, r_2, \ldots, r_n, R_1, R_2, \ldots, R_N, t) \]

some steps

\[ \hat{H} \mid \psi \rangle = E \mid \psi \rangle \]

\[ \hat{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \sum_{pqrs} g_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \]

\[ \mid \psi \rangle = \sum_P C_P \mid P \rangle \]
The problem

Step 1. Adiabatically separate electronic and nuclear motion

\[ \Psi(r, R, t) \rightarrow \psi_e(r; R) \psi_n(R, t) \]

Yields the time-independent Schrödinger Equation for the electrons

\[ \hat{H} \Psi = E \Psi \]

\[ \hat{H} = -\frac{1}{2} \sum_i \nabla_i^2 - \sum_{I,i} \frac{Z_I}{r_{iI}} + \sum_{i>j} \frac{1}{r_{ij}} \]

\[ \Psi(r_1, r_2, \ldots, r_n) \]
The problem

Step 2. Select a (finite) basis of 1-p functions (LCAO, PW)

\[ \chi_\mu(\mathbf{r}) = x^i y^j z^k e^{-\alpha r^2} \]

\[ \phi_p(\mathbf{r}) = \sum_\mu c_{\mu p} \chi_\mu(\mathbf{r}) \]

- Mean field approximation (independent particle model)

\[ \psi_{\text{HF}}(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n) = \hat{A} \prod_i \phi_i(\mathbf{r}_i) \]

\[ \hat{F} \phi_p(\mathbf{r}) = \varepsilon_p \phi_p(\mathbf{r}) \]

Defines a set of one-particle states and an n-particle Hilbert space

\[ \hat{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \sum_{pqrs} g_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \]
The problem

Step 3. Find the eigenstates

$$\hat{H} |\psi\rangle = E |\psi\rangle$$

$$\hat{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \sum_{pqrs} g_{pqrs} a_p^\dagger a_q^\dagger a_r a_s$$

Dimension of n-p Hilbert space is combinatorial in the number of electrons (n) and available 1-p states (m)

$$|\psi\rangle = \sum_{P} C_P |P\rangle \begin{pmatrix} m \\ n \end{pmatrix}$$

Water: $m = 30 \quad n = 10 \quad : 10^{10}$

Protein: $m = 150000 \quad n = 50000 \quad : 10^{10000}$
Three layers of approximation

Simulation window

I-p Representation cut-off

basis set incompleteness

n-p Representation cut-off

\[ \hat{H} |\psi\rangle = E |\psi\rangle \]

\[ |\psi\rangle = \sum_P C_P |P\rangle \]

polynomial number of parameters
The problem

Step 3. Find approximations to the eigenstates

\[ \hat{H} |\psi\rangle = E |\psi\rangle \]
\[ \hat{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \sum_{pqrs} g_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \]

- \( E + \epsilon \) is acceptable
- Provided \( \epsilon/n < \text{“Chemical accuracy”} \)
Classical Algorithm 1: Coupled Cluster

\[ \hat{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \sum_{pqrs} g_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \]

- Factorised many-body expansion

\[ |\psi\rangle = e^T |0\rangle \quad T = \sum_{ai} t_i^a a^\dagger_a a_i + \sum_{abij} t_{ij}^{ab} a^\dagger_a a^\dagger_b a_j a_i \]

- Obtain energy and coefficients via projection (like PT)
Classical Algorithm 1: Coupled Cluster

For many cases, convergence with respect to truncation of many-body expansion is near exponential

$$|\psi\rangle = e^{T_1 + T_2 + T_3 + \ldots} |0\rangle$$
Coupled Cluster State of the art

- For insulators, the interactions are short range:
  polynomial number of parameters and operations: $O(n)$

$m > 8800$

$n > 900$

$N > 450$

time $10^6$ seconds

(2 weeks, 1 CPU)

Chemical accuracy for e.g. binding energies
Coupled Cluster success and failure

Simple example of $H_2$ with varying bond length

(b)

Energy $E_h$ vs Bond Length $a_0$

| 00⟩$ + $|11⟩$ vs |0⟩$ and |1⟩$

Works well

Fails
Classical Algorithm 2: DMRG

Tensor train factorisation of the CI vector

\[ C_{\text{DMRG}}^{n_1 n_2 n_3 n_4} = \sum_{i_1 i_2 i_3 \ldots}^{M} A_{i_1}^{n_1} A_{i_2}^{n_2} A_{i_3}^{n_3} A_{i_4}^{n_4} \ldots \]

State-of-the-art

\[ m = 64 \quad n = 30 \quad : \quad 10^{17} \]
Classical Algorithm 3: FCI-QMC

A stochastic realisation of the imaginary-time Schrödinger Equation in n-particle Hilbert-space

\[
i \frac{\partial \Psi}{\partial t} = \hat{H} \Psi \quad \Rightarrow \quad \frac{\partial \Psi}{\partial \tau} = -\hat{H} \Psi
\]

The CI coefficients are represented through a population of walkers in Hilbert space. After reaching steady state, energies and properties are extracted through time-averaging

\[
- \frac{\partial C_P}{\partial \tau} = (H_{PP} - E)C_P + \sum_{Q \neq P} H_{PQ} C_Q
\]

State-of-the-art : \( > 10^{20} \)
Classical Algorithm 4: Density Functional Theory

There is an existence proof that there is a one-to-one mapping between the wave function and the electron density

$$\psi(r_1, r_2, \ldots, r_n) \leftrightarrow \rho(r)$$

$$\min_{\rho} E[\rho] \quad \text{for n-representable densities}$$

Kohn-Sham: search over non-interacting mean-field states

$$\hat{A} \prod_i \phi_i(r_i) \rightarrow \rho(r)$$

$$E[\rho] = T_s + V[\rho] + J[\rho] + V_{xc}[\rho]$$
Approximate Density Functionals in G09

<table>
<thead>
<tr>
<th>EXCHANGE</th>
<th>CORRELATION</th>
<th>EXCHANGE ONLY</th>
<th>PURE</th>
<th>HYBRID</th>
<th>RANGE-SEPARATED HYBRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>VWN</td>
<td>HFS</td>
<td>VSXC</td>
<td>B3LYP</td>
<td>HSEH1PBE</td>
</tr>
<tr>
<td>XA</td>
<td>VWN5</td>
<td>XAlpha</td>
<td>HCTH</td>
<td>B3P86</td>
<td>OHSE2PBE</td>
</tr>
<tr>
<td>B</td>
<td>LYP</td>
<td>HFB</td>
<td>HCTH93</td>
<td>B3PW91</td>
<td>OHSE1PBE</td>
</tr>
<tr>
<td>PW91</td>
<td>PL</td>
<td>HCTH147</td>
<td>B1B95</td>
<td>wB97XD</td>
<td></td>
</tr>
<tr>
<td>mPW</td>
<td>P86</td>
<td>HCTH407</td>
<td>mPW1PW91</td>
<td>wB97</td>
<td></td>
</tr>
<tr>
<td>G96</td>
<td>PW91</td>
<td>tHCTH</td>
<td>mPW1LYP</td>
<td>wB97X</td>
<td></td>
</tr>
<tr>
<td>PBE</td>
<td>B95</td>
<td>M06L</td>
<td>mPW1PBE</td>
<td>LC-wPBE</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>PBE</td>
<td>B97D</td>
<td>mPW3PBE</td>
<td>CAM-B3LYP</td>
<td></td>
</tr>
<tr>
<td>TPSS</td>
<td>TPSS</td>
<td>B97D3</td>
<td>B98</td>
<td>HISSbPBE</td>
<td></td>
</tr>
<tr>
<td>BRx</td>
<td>KCIS</td>
<td>SOGGA11</td>
<td>B971</td>
<td>M11</td>
<td></td>
</tr>
<tr>
<td>PKZB</td>
<td>BRC</td>
<td>M11L</td>
<td>B972</td>
<td>N12SX</td>
<td></td>
</tr>
<tr>
<td>wPBEh</td>
<td>PKZB</td>
<td>N12</td>
<td>PBE1PBE</td>
<td>MN12SX</td>
<td></td>
</tr>
<tr>
<td>PBEh</td>
<td>VP86</td>
<td>MN12L</td>
<td>B1LYP</td>
<td>O3LYP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BHandH</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BHandHLYP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BMK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M06</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M06HF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M062X</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>tHCTHhyb</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>APFD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>APF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SOGGA11X</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PBEh1PBE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TPSSh</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X3LYP</td>
<td></td>
</tr>
</tbody>
</table>

LONG RANGE CORRECTION

LC-
State-of-the-art for DFT

Accuracy - twice “Chemical Accuracy” if the molecule under investigation resembles those the functionals were parameterised to get right. Else …

(Important, but shrinking, class of problems for which DFT fails)

\[
\begin{align*}
N &= 16000 \\
m &= 1000000 \\
n &= 50000 \\
1000 \text{ time steps} \\
\text{Blue Gene/Q}
\end{align*}
\]
Summary

For a wide class of molecules, the electronic structure of the undistorted ground state is relatively easy. Weakly correlated. DFT and CCSD(T) hit different sweet spots of accuracy vs cost.

An important class of systems have difficult electronic structure, usually characterised by many degenerate or near degenerate states and a poor mean field solution. Strongly correlated.

We don’t know how to solve these problems efficiently and reliably.
Quantum Chemistry on a Quantum Computer

Exploit the mapping of Fermionic creation and annihilation operators onto qubit operations

\[ \hat{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \sum_{pqrs} g_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \]

Unitary-type operations can be used to prepare a state, perform QFT, and evolve a state according to a Hamiltonian

\[ |\psi\rangle = U |0\rangle \quad e^{i\hat{H}t} |\psi\rangle \]

Trotter expansion makes it possible to decompose general angle unitaries from \( e^{i\hat{H}t} \) into a sequence of local angle unitaries.
Quantum Algorithm 1: Phase Estimation

\[
\text{Prepare}\quad |\psi\rangle = U|0\rangle
\]

QFT \quad E

Requires that \( |\psi\rangle \) has a large overlap with the true eigenstate. For the easy cases, where CC works, this is probably possible.

Open question: How to prepare good states for hard cases?

\[
\text{Prepare}\quad |0\rangle\quad \text{Adiabatic map}\quad e^{i((1-t)\hat{F}+t\hat{H})}|0\rangle\quad \text{QFT}\quad E
\]
Quantum Algorithm 2: Variational Approach

Decompose the Hamiltonian into a sum of unitary operations

\[ \hat{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \sum_{pqrs} g_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \]

\[ = \sum_i c_i U_i \]

\[ |\psi\rangle = U |0\rangle \]

Measure \( E \)

Refine U
Quantum Algorithm 2: Variational Approach

We will only have access to a limited space of unitaries

\[ U = e^{\hat{T} - \hat{T}^\dagger} \]

Questions:

Is unitary truncated coupled cluster better than regular?
How easy or hard is the refinement of \( U \)?
Quantum Algorithm 2: Variational Approach

Numerical experiments for a 1-d periodic Hubbard Hamiltonian

\[
\hat{H} = -t \sum_{\langle i,j \rangle \sigma} a_{i\sigma}^\dagger a_{j\sigma} + U \sum_i n_{i\uparrow} n_{i\downarrow}
\]

4 1-particle states for each spin

Half-filled case:

2 up spin particles, two down spin particles: 36 states
Quantum Algorithm 2: Variational Approach

Numerical experiments for a 1-d periodic Hubbard Hamiltonian

\[ |\psi\rangle = e^{ \hat{T} - \hat{T}^\dagger } |0\rangle \]

\[ T = \sum_{ai} t_i^a a_i^\dagger a_i + \sum_{abij} t_{ij}^{a b} a_i^\dagger a_j^\dagger a_j a_i \]
Summary

Classical algorithms are efficient when the electronic structure is well approximated by one occupation number state.

Classical algorithms struggle when many occupation number states are required for a qualitatively correct ground state. This is where quantum algorithms will probably have the biggest impact.

This situation occurs in e.g. superconducting materials and clusters of transition metal atoms in the body.

Many important topics have not been mentioned:

- Excited states for Fermionic systems
- Bosonic Hamiltonians for QM of nuclei