Notes for Alessandro Tammaro's thesis

July 17, 2021

Contents

1 Classical algorithms for quantum chemistry **2**

1.1 The Born-Oppenheimer approximation 2

1.2 Second quantization hamiltonian 2

1.3 Basis sets in quantum chemistry 3

1.4 An exercise, and some resources 4

1.5 Slater determinants 5

1.6 The Hartree-Fock method 6

1.7 Solutions of the Hartree-Fock equations 7

1.8 An exercise, and a first investigation 8

1.9 Molecular symmetries 9

1.10 An exercise 10

1.11 Exact diagonalization (FCI) 11

1.12 Exact diagonalization in an active space (CASCI) 12

1.13 Two exercises 13

2 Quantum algorithms for quantum chemistry **14**

2.1 Mapping fermions to qubits 14

3 An exercise **15**

3.1 Variational quantum algorithms 16

3.2 Ansatz states for variational quantum simulations 17

3.3 An exercise 18

3.4 Decoherence and error correction 20

1 Classical algorithms for quantum chemistry

1.1 The Born-Oppenheimer approximation

The basic task of electronic structure theory is to solve the Schrödinger equation

$$\hat{H}|\Psi\rangle = E|\Psi\rangle \quad , \tag{1}$$

to determine the ground and low-lying excited states of a Hamiltonian $\hat{H}$. Often nuclear motion can be neglected or treated classically, in which case $\hat{H}$ has the Born-Oppenheimer form,

$$\hat{H} = \sum_{a<b} \frac{Z_a Z_b}{|\mathbf{R}_a - \mathbf{R}_b|} + \sum_i \left[-\frac{1}{2}\nabla_i - \sum_a \frac{Z_a}{|\mathbf{R}_a - \mathbf{x}_i|} \right] + \sum_{i<j} \frac{1}{|\mathbf{x}_i - \mathbf{x}_j|} \quad , \tag{2}$$

where we consider a molecule with nuclei having atomic numbers and positions

$$(Z_1, \mathbf{R}_1) \dots (Z_n, \mathbf{R}_n) \quad , \tag{3}$$

and we use atomic units, i.e. Bohr and Hartree for lengths and energies,

$$\mathrm{a_B} = \frac{\hbar^2}{2m_e e^2} = 5.29\, 10^{-11}\,\mathrm{m} \quad , \quad \mathrm{Ha} = \frac{\hbar^2}{m {\mathrm{a_B}}^2} = 27.21\,\mathrm{eV} \quad . \tag{4}$$

1.2 Second quantization hamiltonian

The first step to address the electronic structure problem by computational means is typically to represent the Born-Oppenheimer Hamiltonian in second quantization, which requires to define a basis $\{\psi_p\}_{p=1}^M$ of orthonormal orbitals. Given such a basis, the Born-Oppenheimer Hamiltonian takes the form

$$\hat{H} = E_0 + \sum_{pq,\sigma} h_{pq} \hat{a}^\dagger_{p\sigma} \hat{a}_{q\sigma} + \sum_{prqs,\sigma\tau} \frac{(pr|qs)}{2} \hat{a}^\dagger_{p\sigma} \hat{a}^\dagger_{q\tau} \hat{a}_{s\tau} \hat{a}_{r\sigma} \quad . \tag{5}$$

where

- E_0 is the nuclear repulsion $E_0 = \sum_{a<b} \frac{Z_a Z_b}{|\mathbf{R}_a - \mathbf{R}_b|}$,
- h is the one-body part of the Hamiltonian,

$$h_{pq} = \int d\mathbf{x}\, \psi^*_p(\mathbf{x}) \left(-\frac{1}{2}\nabla - \sum_a \frac{Z_a}{|\mathbf{R}_a - \mathbf{x}|} \right) \psi_q(\mathbf{x}) \quad , \tag{6}$$

- $\sigma, \tau \in \{\uparrow, \downarrow\}$ are spin indices
- $(pr|qs)$ is the electron-repulsion integral

$$(pr|qs) = \int d\mathbf{x} \int d\mathbf{y}\, \psi^*_p(\mathbf{x}) \psi_r(\mathbf{x}) \frac{1}{|\mathbf{x} - \mathbf{y}|} \psi^*_q(\mathbf{y}) \psi_s(\mathbf{y}) \quad . \tag{7}$$

1.3 Basis sets in quantum chemistry

The orbitals $\{\psi_p\}_{p=1}^M$ are typically produced by orthonormalization of a "basis set" of atom centered Gaussian orbitals. A natural choice is represented by Slater-type orbitals (STOs),

$$\psi_{nlm}(r,\theta,\phi) \propto r^{n-1}\, e^{-\alpha r}\, Y_l^m(\theta,\phi) \equiv f_n(r)\, Y_l^m(\theta,\phi) \quad . \tag{8}$$

However, the computation of the second-quantization Hamiltonian is very difficult when STOs are used. Because it is easier to calculate overlap and other integrals with Gaussian basis functions, STOs are typically approximated with linear combinations, or contractions, of primitive Gaussian functions,

$$f_n(r) = \sum_{j=1}^n c_j e^{-\alpha_j r^2} \quad . \tag{9}$$

Some widely used basis sets are listed below

- Minimal basis sets (STO-nG) – a minimal basis set includes only those orbitals that would be occupied by electrons in the atoms forming the molecule. In an STO-nG basis, n Gaussians are used to approximate each STO. Minimal bases are very compact, but often inaccurate: their main limitations are the inability to capture polarization of electronic orbitals in response to the presence of surrounding charges, and transition to high-angular momentum orbitals due to electron-electron interaction.

- split-valence basis sets (n-abG) – include orbitals with higher angular momentum, to ensure more accurate calculations. In a n-abG basis, core atomic orbitals are contractions of n primitive Gaussian functions, and valence orbitals are composed of two basis functions, which are linear combinations of a, b primitive Gaussian functions respectively. A n-abG** basis contains polarization functions, allowing the electronic density to respond to the presence of surrounding charges. A n-ab++G basis contains diffuse functions, allowing to describe intermolecular bonding. Split-valence basis sets are more flexible than minimal bases, but their accuracy is limited.

- Correlation-consistent basis sets (cc-pVxZ) – designed for converging post Hartree-Fock calculations systematically to the complete basis set limit using empirical extrapolation techniques. In a cc-pVxZ basis, "cc-p" stands for "correlation-consistent polarized", "V" indicates they are valence-only basis sets, and "xZ" indicates the shells of polarization functions added to the basis (x=D,T,Q,5). Augmented aug-cc-pVxZ bases contain diffuse basis functions. Core-valence cc-pCVxZ bases allow to correlate core and valence orbitals. Douglas-Kroll cc-pVxZ-dk bases contain relativistic corrections, important for transition metals and rare earths. Correlation-consistent basis sets are more expensive, but represent the current state of the art for post Hartree-Fock calculations.

1.4 An exercise, and some resources

```
import numpy as np
from   pyscf import gto,tools

# setting up a molecule

mol          = gto.Mole()
mol.verbose  = 5
mol.atom     = [['O',(0, 0.0000, 0.1173)],
                ['H',(0, 0.7572,-0.4692)],
                ['H',(0,-0.7572,-0.4692)]]
mol.charge   = 0
mol.spin     = 0
mol.basis    = 'sto-6g'
mol.build()

# printing the AO labels

for x in mol.ao_labels():
    print(x)

# overlap, kinetic energy, potential energy

s = mol.intor('int1e_ovlp')
t = mol.intor('int1e_kin')
v = mol.intor('int1e_nuc')

# electron repulsion integral

eri = mol.intor('int2e',aosym='s1')

# plot the AOs

nao = mol.nao_nr()
aos = np.eye(nao)
for i in range(nao):
    tools.cubegen.orbital(mol,'ao_%d.cube'%i,aos[:,i])
```

- a database of molecular properties: https://cccbdb.nist.gov/
- PySCF, a program for quantum chemistry calculations: https://sunqm.github.io/pyscf/
- VESTA, a program to visualize atomic and molecular orbitals: https://jp-minerals.org/vesta/en/
- two important references on chemistry [1, 2]
- a comprehensive collection of PySCF exercies https://github.com/pyscf/pyscf/tree/master/examples
- references describing basis sets are listed in https://github.com/pyscf/pyscf/tree/master/pyscf/gto/basis/

1.5 Slater determinants

Slater determinants are many-body states defined by the occupation of orbitals,

$$\begin{array}{ll} \text{restricted} & |\Psi\rangle = \hat{a}^\dagger_{c_1\uparrow} \dots \hat{a}^\dagger_{c_n\uparrow} \hat{a}^\dagger_{c_1\downarrow} \dots \hat{a}^\dagger_{c_m\downarrow} |\emptyset\rangle \\ \text{unrestricted} & |\Psi\rangle = \hat{a}^\dagger_{c_1\uparrow} \dots \hat{a}^\dagger_{c_n\uparrow} \hat{a}^\dagger_{c'_1\downarrow} \dots \hat{a}^\dagger_{c'_m\downarrow} |\emptyset\rangle \\ \text{generalized} & |\Psi\rangle = \hat{a}^\dagger_{c_1} \dots \hat{a}^\dagger_{c_{n+m}} |\emptyset\rangle \end{array} \qquad (10)$$

Here n, m denotes the number of spin-up and spin-down particles respectively. When $n = m$ ($n \neq m$) we talk about closed-shell (open-shell) systems. The occupied orbitals are linear combination of basis functions,

$$\begin{array}{ll} \text{restricted} & |c_i \sigma\rangle = \sum_p C_{pi} |\psi_p \sigma\rangle \\ \text{unrestricted} & |c_i \uparrow\rangle = \sum_p C_{pi} |\psi_p \uparrow\rangle, \; |c'_i \downarrow\rangle = \sum_p C'_{pi} |\psi_p \downarrow\rangle \\ \text{generalized} & |c_i\rangle = \sum_{p\sigma} C_{p\sigma,i} |\psi_p \sigma\rangle \to \sum_p C_{pi} |\chi_p\rangle \end{array} \qquad (11)$$

For simplicity, in these notes we will consider generalized determinants only. The following properties will be useful in the study of Hartree-Fock calculations and orbital relaxation.

- orthonormality of orbitals – a Slater determinant is invariant under invertible transformations of the orbitals,

  $$|\Psi'\rangle = \hat{a}^\dagger_{c'_1} \dots \hat{a}^\dagger_{c'_n} |\emptyset\rangle = \det(T) \, \hat{a}^\dagger_{c_1} \dots \hat{a}^\dagger_{c_n} |\emptyset\rangle = \det(T) \, |\Psi\rangle \quad , \qquad (12)$$

  with $|c'_k\rangle = \sum_m T_{km} |c_m\rangle$

- Thouless theorem – given a one-body operator $\hat{X} = \sum_{pq} X_{pq} \, \hat{a}^\dagger_p \hat{a}_q$ and a Slater determinant $|\Psi\rangle$, $|\Psi'\rangle = e^{\hat{X}} |\Psi\rangle$ is also a Slater determinant, with

  $$|c_i\rangle = \sum_{pi} C_{pi} \, |\psi_p\rangle \quad , \quad |c'_i\rangle = \sum_{pq} \left(e^X\right)_{pq} C_{qi} \, |\psi_p\rangle \qquad (13)$$

- Wick's theorem – all the properties of a Slater determinant are contained in its one-body density matrix,

  $$\langle\Psi| \hat{a}^\dagger_p \hat{a}_r |\Psi\rangle = \rho_{pr} \quad , \quad \rho_{pr} = \sum_{i=1}^n C_{pi} \, C^*_{ri} \quad . \qquad (14)$$

  In particular, the two-body density matrix has the form

  $$\langle\Psi| \hat{a}^\dagger_p \hat{a}^\dagger_q \hat{a}_s \hat{a}_r |\Psi\rangle = \rho_{pr} \, \rho_{qs} - \rho_{ps} \, \rho_{qr} \qquad (15)$$

1.6 The Hartree-Fock method

The Hartree-Fock method approximates the exact wavefunction of a molecule by a single Slater determinant. It constructs a set of n coupled nonlinear equations for the n spin-orbitals occupied in the Slater determinant by invoking the variational principle. A solution of these equations yields a stationary point, but not necessary a local or a global minimum, of the energy.

Due to Wick's theorem, the expectation value of the Born-Oppenheimer Hamiltonian over a Slater determinant is

$$E[C] = \sum_{pq} h_{pq} \rho_{pq} + \sum_{prqs} \frac{(pr|qs)}{2} \left(\rho_{pr} \rho_{qs} - \rho_{ps} \rho_{qr} \right) \tag{16}$$

To minimize the functional $E(C)$ under the constraint of orthonormality of the orbitals, we define the Lagrangian

$$\begin{aligned} L[C] &= E[C] + \sum_{ij} \lambda_{ij} \left(C^\dagger C - \mathbb{1} \right)_{ij} \\ &= E[C] + N[C] \end{aligned} \tag{17}$$

The Hartree-Fock equations are obtained imposing

$$\frac{\partial L}{\partial C^*_{ck}} = \sum_{ab} \frac{\partial \rho_{ab}}{\partial C^*_{ck}} \frac{\partial E}{\partial \rho_{ab}} + \frac{\partial N}{\partial C^*_{ck}} = 0 \quad . \tag{18}$$

To explicit the Hartree-Fock equations, we assume real-valued matrix elements for simplicity, and we use the following results,

$$\begin{aligned} \frac{\partial E}{\partial \rho_{ab}} &= h_{ab} + \sum_{qs} (ab|qs) \rho_{qs} - \sum_{qr} (ar|qb) \rho_{qr} = F[\rho]_{ab} \quad , \\ \frac{\partial \rho_{ab}}{\partial C^*_{ck}} &= C_{ak} \delta_{bc} \quad , \\ \frac{\partial N}{\partial C^*_{ck}} &= C_{cj} \delta_{ki} \end{aligned} \tag{19}$$

leading to

$$\frac{\partial L}{\partial C^*_{ck}} = \left(F[\rho] \, C \right)_{ck} - \left(C \lambda \right)_{ck} = 0 \quad . \tag{20}$$

Introducing a unitary transformation that diagonalizes the λ matrix, we obtain the "Hartree-Fock-Roothan" equations

$$\left(F[\rho] \, C \right)_{ck} = C_{ck} \, \varepsilon_k \quad . \tag{21}$$

Here, we formulated the Hartree-Fock-Roothan equations in a basis of orthonormal orbitals. These can be reformulated in the basis of atomic orbitals, where they take the form

$$\left(F[\rho] \, C \right)_{ck} = \left(SC \right)_{ck} \, \varepsilon_k \quad . \tag{22}$$

The HF orbitals, eigenvectors of the Fock matrix in HF theory, are called molecular orbitals (MOs), as opposed to atomic orbitals (AOs).

1.7 Solutions of the Hartree-Fock equations

The Hartree-Fock equations are solved iteratively, following this procedure

1. Specify molecule, basis, and electronic state of interest
2. Guess initial coefficients C
3. Form Fock matrix F
4. Solve $FC = SC\varepsilon$
5. Use new coefficients C to build new Fock matrix F
6. repeat until convergence of the energy and/or of ρ

The initial coefficients C are important to help or allow convergence of the Hartree-Fock equations. The initial coefficients are typically

- the ground-state of the one-electron Hamiltonian, or
- a density matrix constructed summing over single-atom densities evaluated at ANO level of theory and then projected on the basis set used for the current calculation, or
- a user-defined guess, or
- the result of a previous calculation

Some practical considerations are the following:

- Hartree-Fock self-consistent-field (SCF) usually converges fairly well with a good initial guess, at least for compressed or equilibrium molecular geometries.
- Stretched bonds, diradicals, transition metals, high-spin states and other chemical situations can lead to problematic convergence
- molecular point-group symmetries should be enforced in the initial guess and throughout the algorithm iterations
- while the HF-SCF is guaranteed to find a stationary point of the energy, they are not guaranteed to find a local or global minimum; this has to be checked after the calculation has finished
- users have to make sure that the orbital occupations and spin states are correct. There is no algorithmic tool to verify that.
- In cases of problematic convergence, the algorithm can be stabilized by second-order (Newton) solvers [3] or the DIIS (direct inversion in the iterative subspace) technique [4]

1.8 An exercise, and a first investigation

```
import numpy as np
from   pyscf import gto,scf

mol          = gto.Mole()
mol.verbose  = 4
mol.atom    = [['O',(0, 0.0000, 0.1173)],
               ['H',(0, 0.7572,-0.4692)],
               ['H',(0,-0.7572,-0.4692)]]
mol.charge   = 0
mol.spin     = 0
mol.symmetry = True
mol.basis    = '6-31g'
mol.build()

mf = scf.RHF(mol)
E  = mf.kernel()
if(not mf.converged):
   mf = scf.newton(mf)
   E  = mf.kernel(mf.make_rdm1())
print("SCF energy, original    ",E)

rho  = mf.make_rdm1()
h1   = mf.get_hcore()
h2   = mol.intor('int2e',aosym='s1')
E0   = mol.energy_nuc()
E1   = np.einsum('pq,pq',h1,rho)
E2   = 0.50*np.einsum('prqs,pr,qs',h2,rho,rho)
E2  -= 0.25*np.einsum('prqs,ps,qr',h2,rho,rho)
print("SCF energy, recomputed ",E0+E1+E2)

C    = mf.mo_coeff
eps = mf.mo_energy
occ = mf.mo_occ
print("Hartree-Fock energies and occupations")
for i in range(mol.nao_nr()):
    print('%2d %.6f %d ' % (i,eps[i],int(occ[i])))
```

- The folder ”Esercizi_PySCF” contains calculations for the hydroxyl radical $OH^{\cdot}$ and the hydroxyl anion OH^- How do the SCF energies for $OH^{\cdot}$ and OH^- depend on the basis set, e.g. using cc-pVxZ and aug-cc-pVxZ? Which species is more stable, i.e. has the lowest energy?

1.9 Molecular symmetries

Molecular symmetries, in chemistry, describe the symmetry properties present in molecules. Molecular symmetries allow to classify electronic wavefunctions according to their symmetry properties, and to understand several molecular properties, such as multipole moments and allowed electronic transitions. Appropriate treatment of molecular symmetries economizes Hartree-Fock and post Hartree-Fock calculations, and accelerates and stabilizes convergence. It is an especially important ingredient in orbital relaxation as well.

Symmetry operations are geometrical operations S, applied to the positions $\mathbf{R}_i$ of the nuclei constituting a molecule, that leave the molecular geometry unchanged. Examples of symmetry operations are

- C_n (rotations by a multiple of $2\pi/n$ around an axis)
- i (inversion through a center of symmetry)

The symmetries of a molecular geometry form a group, called the symmetry group of the molecule. Examples of symmetry groups are

- $C_{\infty v}$ (rotational symmetry around an axis, found in linear hetero-nuclear molecules like CuH)
- $D_{\infty h}$ (rotational symmetry around an axis and inversion symmetry, found in linear homo-nuclear molecules like H_2 or atoms like Cu).

When bases of atom-centered Gaussian orbitals are used, symmetries are represented by a group of invertible transformations of the atomic orbitals, $\hat{S}|\varphi_p\rangle = \sum_r S_{pr}|\varphi_r\rangle$. Symmetries help labeling molecular orbitals according to irreducible representations of the molecular symmetry group, which in turn are connected to the behavior of the molecular orbitals under symmetry transformation. For the groups $C_{\infty v}$ and $D_{\infty h}$, the nomenclature of irreducible representations is based on the following conventions

- A – denotes symmetric under rotations around the principal axis)
- E, T – denotes doubly and triply degenerate representations
- g, u – when the point group has an inversion center (i.e. $D_{\infty h}$) , these subscripts indicate even, odd (gerade, ungerade) behavior under inversion

Therefore, irreducible representations of the symmetry groups $C_{\infty v}$ and $D_{\infty h}$ are

- for $C_{\infty v}$ – A_1, E_{1x}, E_{1y}, E_{2x}, E_{2y} and so on
- for $D_{\infty h}$ – A_{1g}, A_{1u}, E_{1gx}, E_{1gy}, E_{1ux}, E_{1uy}, and so on

1.10 An exercise

```
from pyscf import gto,scf,symm

mol = gto.Mole()
mol.build(verbose  = 0,
          atom     = [[’O’,(0.0000, 0.0000, 0.1173)],
                       [’H’,(0.0000, 0.7572,-0.4692)],
                       [’H’,(0.0000,-0.7572,-0.4692)]],
          basis    = ’cc-pvdz’,
          spin     = 0,
          symmetry = True)

mf = scf.RHF(mol)
e  = mf.kernel()
print("energy      ",e)
print("occupancies ",mf.get_irrep_nelec())

occ = [int(x) for x in mf.mo_occ]
irr = symm.label_orb_symm(mol,mol.irrep_name,mol.symm_orb,mf.
                                  mo_coeff)
eps = mf.mo_energy

for k,(ok,ik,ek) in enumerate(zip(occ,irr,eps)):
    print(’%2d  %d  %s  %.6f’ % (k,ok,ik+’ ’*(4-len(ik)),ek))
```

- Plot a few molecular orbitals of H_2O (one or two per irreducible representation) and connect their graphical representation with the corresponding irreducible representation.

- A few references on group theory and chemistry are [5, 6]

1.11 Exact diagonalization (FCI)

The Fock space of electrons in M spatial orbitals is spanned by the following quantum states, called "configurations" or "occupation number states",

$$|\mathbf{x}\rangle = (\hat{a}^\dagger_{0,\uparrow})^{x_0} \dots (\hat{a}^\dagger_{M-1,\uparrow})^{x_{M-1}} (\hat{a}^\dagger_{0,\downarrow})^{x_M} \dots (\hat{a}^\dagger_{M-1,\downarrow})^{x_{2M-1}} |\emptyset\rangle \quad , \tag{23}$$

where $\mathbf{x}$ is a binary string of length $2M$; there are 4^M such states. The Hilbert space of $N_\uparrow$ and $N_\downarrow$ electrons in M spatial orbitals is spanned by states with $\sum_{i=0}^{M-1} x_i = N_\uparrow$, $\sum_{i=M}^{2M-1} x_i = N_\downarrow$; there are $\binom{M}{N_\uparrow} \times \binom{M}{N_\downarrow}$ such states. Examples for $M = 2$ spatial orbitals are listed in the following table.

$\mathbf{x}$	$(N_\uparrow, N_\downarrow)$	$\lvert\mathbf{x}\rangle$	$\mathbf{x}$	$(N_\uparrow, N_\downarrow)$	$\lvert\mathbf{x}\rangle$
0000	(0,0)	$\lvert\emptyset\rangle$	0110	(1,1)	$\hat{a}^\dagger_{0\downarrow}\hat{a}^\dagger_{1\uparrow}\lvert\emptyset\rangle$
0001	(1,0)	$\hat{a}^\dagger_{0\uparrow}\lvert\emptyset\rangle$	1010	(1,1)	$\hat{a}^\dagger_{1\downarrow}\hat{a}^\dagger_{1\uparrow}\lvert\emptyset\rangle$
0010	(1,0)	$\hat{a}^\dagger_{1\uparrow}\lvert\emptyset\rangle$	1100	(0,2)	$\hat{a}^\dagger_{1\downarrow}\hat{a}^\dagger_{0\downarrow}\lvert\emptyset\rangle$
0100	(0,1)	$\hat{a}^\dagger_{0\downarrow}\lvert\emptyset\rangle$	1110	(1,2)	$\hat{a}^\dagger_{1\downarrow}\hat{a}^\dagger_{0\downarrow}\hat{a}^\dagger_{1\uparrow}\lvert\emptyset\rangle$
1000	(0,1)	$\hat{a}^\dagger_{1\downarrow}\lvert\emptyset\rangle$	1101	(1,2)	$\hat{a}^\dagger_{1\downarrow}\hat{a}^\dagger_{0\downarrow}\hat{a}^\dagger_{1\uparrow}\lvert\emptyset\rangle$
0011	(2,0)	$\hat{a}^\dagger_{1\uparrow}\hat{a}^\dagger_{0\uparrow}\lvert\emptyset\rangle$	1011	(2,1)	$\hat{a}^\dagger_{0\downarrow}\hat{a}^\dagger_{1\uparrow}\hat{a}^\dagger_{0\uparrow}\lvert\emptyset\rangle$
0101	(1,1)	$\hat{a}^\dagger_{0\downarrow}\hat{a}^\dagger_{0\uparrow}\lvert\emptyset\rangle$	0111	(2,1)	$\hat{a}^\dagger_{1\downarrow}\hat{a}^\dagger_{1\uparrow}\hat{a}^\dagger_{0\uparrow}\lvert\emptyset\rangle$
1001	(1,1)	$\hat{a}^\dagger_{1\downarrow}\hat{a}^\dagger_{0\uparrow}\lvert\emptyset\rangle$	1111	(2,2)	$\hat{a}^\dagger_{1\downarrow}\hat{a}^\dagger_{0\downarrow}\hat{a}^\dagger_{1\uparrow}\hat{a}^\dagger_{0\uparrow}\lvert\emptyset\rangle$

A generic state can be written as a linear combination of configurations from a set S,

$$|\psi\rangle = \sum_{\mathbf{x} \in S} c_{\mathbf{x}} |\mathbf{x}\rangle \quad , \tag{24}$$

and the Hamiltonian can be represented as a matrix acting on the space of linear combinations of configurations,

$$H = \sum_{\mathbf{x},\mathbf{y} \in S} |\mathbf{x}\rangle \langle \mathbf{x}|H|\mathbf{y}\rangle \langle \mathbf{y}| = \sum_{\mathbf{x},\mathbf{y} \in S} h_{\mathbf{x},\mathbf{y}} \, |\mathbf{x}\rangle\langle \mathbf{y}| \quad . \tag{25}$$

The matrix elements $h_{\mathbf{x},\mathbf{y}}$ can be efficiently computed using the Slater-Condon rules, and the matrix h can in principle be diagonalized to obtain Hamiltonian eigenfuctions, either in the Fock space or in the Hilbert space of $N_\uparrow$ and $N_\downarrow$ electrons in M spatial orbitals. In practice, matrix diagonalization is extremely expensive and excessively informative, since it is often interesting to access the ground and low-energy excited states only. Therefore, chemistry calculations often resort to approximate methods targeting only a few eigenstates based on a subroutine that applies h to a vector $c_{\mathbf{x}}$, e.g. Davidson's algorithm [7]. This kind of exact diagonalization is often referred to as full configuration interaction (FCI) in chemistry literature.

Due to its combinatorial cost with M, $N_\uparrow$ and $N_\downarrow$, FCI is affordable only for up to $M \simeq 15$ orbitals. Methods capturing approximately electronic correlation are thus needed for all but the smallest instances of the electronic structure problem.

1.12 Exact diagonalization in an active space (CASCI)

Often, a reasonable description of a molecule can be obtained focusing on certain "chemically active" electrons and orbitals. Chemically active orbitals are orbitals from which and to which leading electronic excitations go. Chemically active electrons and orbitals form an active space. Orbitals and electrons that are not chemically active can be discarded as follows,

- if low-energy, frozen, i.e. kept doubly occupied during the calculation
- if high-energy, downfolded, i.e. kept empty during the calculation

Core orbitals (those corresponding to the electronic configuration of the noble gas preceding the element studied) always have to be frozen, unless cc-pCVxZ bases are used.

Freezing orbitals – let us consider the electronic structure, written in a basis of orthonormal molecular spin orbitals,

$$H = E_0 + \sum_{pq=1}^{M} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \sum_{prqs=1}^{M} \frac{(pr|qs)}{2} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r \quad . \tag{26}$$

Assume the F lowest-energy molecular spin-orbitals, denoted with uppercase letters PQ, are frozen. The Hamiltonian of the remaining electrons and orbitals is

$$\begin{aligned} H = E_0 &+ \sum_{P=1}^{F} h_{PP} + \sum_{PQ=1}^{F} \frac{(PP|QQ) - (PQ|QP)}{2} \\ &+ \sum_{pq>F} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \sum_{P=1}^{F} \sum_{qs>F} (PP|qs) \hat{a}_q^\dagger \hat{a}_s - \sum_{P=1}^{F} \sum_{qs>F} (Pr|qP) \hat{a}_q^\dagger \hat{a}_r \\ &+ \sum_{prqs>F} \frac{(pr|qs)}{2} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r \end{aligned} \tag{27}$$

- the first row describes a shift in the constant E_0 (energy offset) due to the frozen MOs
- the third row describes the electron-electron repulsion for unfrozen MOs
- the second row is the sum of the one-body part of the Hamiltonian for unfrozen MOs, and of a one-body operator describing an external potential applied by frozen orbitals to unfrozen orbitals

The last K virtual orbitals, on the other hand, can be downfolded by just restricting the summations in the previous equation to indices $F < p < M - K$.

The frozen-core and downfolding approximations do not affect the Hartree-Fock energy and wavefunction, but economizes correlated calculations by removing F particles and $F + K$ molecular spin-orbitals. Care has to be put in removing chemically irrelevant degrees of freedom, or accuracy would be compromised. A FCI calculation in an active space is called a CASCI calculation.

1.13 Two exercises

```
from pyscf import gto,scf,fci,mcscf

mol = gto.Mole()
mol.build(verbose  = 4,
          atom     = [['O',(0.0000, 0.0000, 0.1173)],
                       ['H',(0.0000, 0.7572,-0.4692)],
                       ['H',(0.0000,-0.7572,-0.4692)]],
          basis    = 'sto-6g',
          symmetry = True)

mf = scf.RHF(mol)
mf.kernel()

# FCI,   (5,5) electrons in 7 orbitals
ci = fci.FCI(mol,mf.mo_coeff)
print('E(FCI)   ',ci.kernel()[0])
# CASCI, (4,4) electrons in 6 orbitals
ci = mcscf.CASCI(mf,nelecas=(4,4),ncas=6)
print('E(CASCI) ',ci.kernel()[0])
```

```
rom pyscf import gto,scf,fci,mcscf

mol = gto.Mole()
mol.build(verbose  = 4,
          atom     = [['O',(0.0000, 0.0000, 0.1173)],
                       ['H',(0.0000, 0.7572,-0.4692)],
                       ['H',(0.0000,-0.7572,-0.4692)]],
          basis    = 'cc-pvdz',
          charge   = 0,
          spin     = 0,
          symmetry = 'C2v')

myhf = mol.RHF()

myhf.kernel()
myhf.analyze()

# active space of H 1s and O 2s, 2p orbitals
mymc = myhf.CASCI(nelecas=(4,4),ncas=6)
cas_space_symmetry = {'A1':3,'B1':2,'B2':1}
mo = mcscf.sort_mo_by_irrep(mymc,myhf.mo_coeff,cas_space_symmetry)
mymc.fcisolver.wfnsym = 'A1'

mymc.kernel(mo)
mymc.analyze()
```

2 Quantum algorithms for quantum chemistry

2.1 Mapping fermions to qubits

It is possible to build an isomorphism between the Fock space of electrons in M spatial orbitals and the Hilbert space of $2M$ qubits. The first space is spanned by the 2^{2M} configurations

$$|\mathbf{x}\rangle_f = \left(\hat{a}_0^\dagger\right)^{x_0} \dots \left(\hat{a}_{2M-1}^\dagger\right)^{x_{2M-1}} |\emptyset\rangle \quad , \quad \mathbf{x} \in \{0,1\}^{2M} \quad , \tag{28}$$

and the latter by the computational basis states

$$|\mathbf{x}\rangle_q = |x_0\rangle \otimes \dots \otimes |x_{2M-1}\rangle \quad , \quad \mathbf{x} \in \{0,1\}^{2M} \quad . \tag{29}$$

It is thus very natural to define the isomorphism $|\mathbf{x}\rangle_f \mapsto |\mathbf{x}\rangle_q$ called the Jordan-Wigner transformation [9]. Under the Jordan-Wigner transformation, creation operators are mapped onto qubit operators

$$\hat{a}_p^\dagger \mapsto \sigma_{+,p}\, \sigma_{z,p-1} \, \dots \, \sigma_{z,0} \quad , \quad \sigma_+ = |1\rangle\langle 0| \quad , \tag{30}$$

which in turn permits to transform one-and two-body operators, for example

$$\hat{a}_p^\dagger \hat{a}_p \mapsto \frac{1-\sigma_{z,p}}{2} \quad , \tag{31}$$

and

$$\hat{a}_p^\dagger \hat{a}_q \mapsto \sigma_{+,p}\, \sigma_{z,p-1} \, \dots \, \sigma_{z,q+1}\, \sigma_{+,q} \quad , \quad q < p \quad . \tag{32}$$

In the Jordan-Wigner representation, a Slater determinant of MOs can be encoded with a circuit of $N_\uparrow + N_\downarrow$ X gates with depth 1,

$$|\Psi\rangle_f \mapsto \prod_{p\, occupied} \sigma_{x,p} \left(|0\rangle^{\otimes 2M}\right) \quad . \tag{33}$$

The Jordan-Wigner transformation is of course not the only possible isomorphism between fermions and qubits [10]. An alternative is the parity mapping $|\mathbf{x}\rangle_f \mapsto |\mathbf{p}\rangle_q$, with

$$|\mathbf{p}\rangle_q = |p_0\rangle \otimes \dots \otimes |p_{2M-1}\rangle \quad , \quad p_k = x_0 + \dots + x_k \mod 2 \quad . \tag{34}$$

The parity mapping has an appealing property: the qubits $M-1$ and $2M-1$ contain the parities

$$p_{M-1} = N_\uparrow \operatorname{mod} 2 \quad \text{and} \quad p_{M-1} = (N_\uparrow + N_\downarrow) \operatorname{mod} 2 \tag{35}$$

which, in a typical simulation where spin-z and particle number are constants of motion, do not change. Therefore, qubits $M-1$ and $2M-1$ can be removed from the simulation: this is the content of the ”two-qubit reduction” technique [11]. Molecular point group symmetries with groups isomorphic to $\mathbb{Z}_2 \times \dots \times \mathbb{Z}_2$ can also be used to further reduce the qubit count using the ”tapering” technique [11]. For our studies, it will be more important to observe that

$$\langle\Psi|\hat{a}_p^\dagger \hat{a}_r|\Psi\rangle = 0 \quad \text{unless } p, r \text{ are in the same irrep,} \tag{36}$$

and

$$\langle\Psi|\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_s \hat{a}_r|\Psi\rangle = 0 \quad \text{unless } \hat{a}_s \hat{a}_r|\Psi\rangle \text{ and } \hat{a}_q \hat{a}_p|\Psi\rangle \text{ are in the same irrep.} \tag{37}$$

3 An exercise

```
from qiskit.aqua.operators    import Z2Symmetries
from qiskit.chemistry.drivers import PySCFDriver,UnitsType
from qiskit.chemistry         import FermionicOperator

driver   = PySCFDriver(atom="Li 0.0 0.0 0.0; H 0.0 0.0 1.6",
                       unit=UnitsType.ANGSTROM,charge=0,spin=0,
                       basis='sto-3g')
molecule = driver.run()

repulsion_energy  = molecule.nuclear_repulsion_energy
num_particles     = molecule.num_alpha + molecule.num_beta
num_spin_orbitals = molecule.num_orbitals * 2

ferOp   = FermionicOperator(h1=molecule.one_body_integrals,
                            h2=molecule.two_body_integrals)
qubitOp = ferOp.mapping(map_type='parity', threshold=1e-8)
qubitOp = Z2Symmetries.two_qubit_reduction(qubitOp,num_particles)

print(qubitOp,repulsion_energy)

freeze_list  = [0]
remove_list  = [-3, -2]
remove_list  = [x % molecule.num_orbitals for x in remove_list]
freeze_list  = [x % molecule.num_orbitals for x in freeze_list]
remove_list  = [x - len(freeze_list)      for x in remove_list]
remove_list += [x + molecule.num_orbitals - len(freeze_list)
                for x in remove_list]
freeze_list += [x + molecule.num_orbitals for x in freeze_list]

# frozen core
ferOp  = FermionicOperator(h1=molecule.one_body_integrals,
                           h2=molecule.two_body_integrals)
ferOp,energy_shift = ferOp.fermion_mode_freezing(freeze_list)
num_spin_orbitals -= len(freeze_list)
num_particles     -= len(freeze_list)
shift              = energy_shift + repulsion_energy

qubitOp = ferOp.mapping(map_type='parity', threshold=1e-8)
qubitOp = Z2Symmetries.two_qubit_reduction(qubitOp,num_particles)

print(qubitOp,shift)

# virtual orbitals removal
ferOp              = ferOp.fermion_mode_elimination(remove_list)
num_spin_orbitals -= len(remove_list)
qubitOp = ferOp.mapping(map_type='parity', threshold=1e-8)
qubitOp = Z2Symmetries.two_qubit_reduction(qubitOp,num_particles)

print(qubitOp,shift)
```

3.1 Variational quantum algorithms

Having established a mapping between fermionic and qubit Hilbert spaces, we can encode fermionic wavefunction onto qubit wavefunctions.

In general, finding the ground state of a Hamiltonian is a problem in the complexity class QMA (quantum Merlin-Arthur).

Therefore, in general we cannot guarantee that the ground state of a Hamiltonian can be obtained applying a quantum circuit comprising a polynomial number of gates [12].

However, it is often the case that accurate approximations can be obtained on both classical and quantum computers by formulating an Ansatz and optimizing its structure [13].

The variational quantum eigensolver (VQE) algorithm introduces a family of parametrized wavefunctions of the form

$$|\Psi(\theta)\rangle = \hat{U}(\theta)|\Psi_0\rangle \quad , \quad \hat{U}(\theta) = \prod_g \hat{v}_g(\theta_g) \quad , \tag{38}$$

where $\hat{U}(\theta)$ is a quantum circuit, product of one- and two-qubit gates $\hat{v}_g$, each possibly featuring a set of parameters θ_g (the set of wavefunctions $S = \{\Psi(\theta)\}_\theta$ is called a ”variational form”). VQE determines the best approximation to the ground state of a Hamiltonian $\hat{H}$ by minimizing the energy

$$E(\theta) = \langle \Psi(\theta)|\hat{H}|\Psi(\theta)\rangle \quad , \tag{39}$$

which is done using an optimizer. There are multiple optimizers, for example

- gradient-free optimizers (e.g. cobyla, spsa, nelder-mead) making use of a subroutine to evaluate the energy $E(\theta)$ only

- gradient-based optimizers (e.g. slsqp, bfgs, conjugate gradient) making use of a subroutine to evaluate the energy gradient $\vec{g}(\theta) = \frac{\partial E}{\partial \theta}(\theta)$

Typically, access to derivatives makes algorithms more accurate and efficient. However, in presence of statistical uncertaintis (as it happens on quantum hardware), gradients need to be estimated analytically, which can be challenging for some Ansätze.

Once the optimal parameters are found, the wavefunction $\Psi(\theta)$ can be used to compute properties such as particle number, spin-z, spin-squared,

$$\hat{N} = \sum_{p\sigma} \hat{a}^\dagger_{p\sigma} \hat{a}_{p\sigma} \quad , \quad \hat{S}_i = \sum_{p\sigma\tau} [\sigma_i]_{\sigma\tau} \, \hat{a}^\dagger_{p\sigma} \hat{a}_{p\tau} \quad , \quad \hat{S}^2 = \sum_i \hat{S}_i^2 \quad , \tag{40}$$

or one- and two-body density matrices. The former are important for diagnosing symmetry breaking phenomena in the VQE state (e.g. spin contamination), the latter are used in orbital relaxation.

3.2 Ansatz states for variational quantum simulations

The structure of the Ansatz is important to obtain a balance between accuracy and computational cost in a VQE simulation. Examples of widely used Ansätze are

- quantum unitary coupled-cluster [14, 15, 16, 17] with single and double excitations (q-uCCSD),

  $$|\Psi(\theta)\rangle = e^{\hat{T}-\hat{T}^\dagger}|\Psi_{RHF}\rangle \quad , \quad \hat{T} = \sum_{ai} t_i^a \, \hat{a}_a^\dagger \hat{a}_i + \sum_{aibj} t_{ij}^{ab} \, \hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_j \hat{a}_i \quad , \tag{41}$$

  where sums run over occupied $(ijkl)$ and virtual $(abcd)$ orbitals in the mean-field reference state. VQE/q-UCCSD is typically implemented by a Trotter decomposition,

  $$\begin{aligned} e^{\hat{T}-\hat{T}^\dagger} &= \prod_{i=1}^{d} e^{\frac{\hat{T}-\hat{T}^\dagger}{d}} \\ &\simeq \prod_{i=1}^{d} \left[\prod_{ai} e^{\frac{t_i^a}{d} \, \hat{a}_a^\dagger \hat{a}_i - \mathrm{h.c.}} \prod_{abij} e^{\frac{t_{ij}^{ab}}{d} \, \hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_j \hat{a}_i - \mathrm{h.c.}} \right] \quad , \end{aligned} \tag{42}$$

  VQE/q-uCCSD is typically very accurate, but also very expensive, which has limited implementations on quantum hardware to very small problems (e.g. H_2/STO-6G).

- hardware-efficient Ansätze [18], such as the following R_y Ansatz,

  $$|\Psi(\theta)\rangle = \prod_{k=1}^{r} \left[\prod_{i} R_y(\theta_{k,i}) \prod_{ij \in E} \mathsf{cX}_{ij} \right] \prod_{i} R_y(\theta_{0,i}) |\Psi_{RHF}\rangle \quad . \tag{43}$$

  As the name implies, hardware-efficient Ansätze privilege matching the constraints – on qubit number, gate number, gate connectivity and circuit depth – imposed by decoherence on quantum hardware, rather than accuracy.

  As such, they have been demonstrated on quantum hardware in multiple occasions, but often deliver results of limited accuracy, compared against VQE/q-UCCSD.

3.3 An exercise

```
from qiskit import *
from qiskit.chemistry.drivers import UnitsType,HFMethodType
from qiskit.chemistry.core import Hamiltonian,TransformationType
from qiskit.chemistry.core import QubitMappingType
from qiskit.chemistry.components.initial_states import HartreeFock
from qiskit.chemistry.components.variational_forms import UCCSD
from qiskit.aqua.components.optimizers import L_BFGS_B
from qiskit.aqua.algorithms import VQE
from qiskit.aqua import QuantumInstance

# set up molecular object

import sys
sys.path.append('./pyscfd')
from pyscfdriver import *
from utils       import *

outfile   = open('h2o/qiskit_vqe.txt','w')
driver    = PySCFDriver(atom='''O 0.0000  0.0000  0.117;
                                H 0.0000  0.7570 -0.4690;
                                H 0.0000 -0.7570 -0.4690''',
                        unit=UnitsType.ANGSTROM,charge=0,spin=0,
                        basis='sto-6g',hf_method=HFMethodType.RHF,
                        symgroup='C2v',outfile=outfile)
molecule  = driver.run()

core      = Hamiltonian(transformation=TransformationType.FULL,
                        qubit_mapping=QubitMappingType.PARITY,
                        two_qubit_reduction=True,freeze_core=False,
                        orbital_reduction=[0,1])
H_op,A_op = core.run(molecule)

# tapering off qubits

z2syms,sqlist           = None,None
H_op,A_op,z2syms,sqlist = taper(molecule,core,H_op,A_op,outfile)

n_orb   = core._molecule_info['num_orbitals']
mapping = core._qubit_mapping
tqr     = core._two_qubit_reduction
n_part  = core._molecule_info['num_particles']

# Hartree-Fock state as a quantum circuit

init_state = HartreeFock(num_orbitals=n_orb,qubit_mapping=mapping,
                         two_qubit_reduction=tqr,
                         num_particles=n_part,sq_list=sqlist)
circuit    = init_state.construct_circuit()

outfile.write("\nHartree-Fock energy %f \n" % (molecule.hf_energy))
outfile.write("\nHartree-Fock circuit\n")
outfile.write(str(circuit.draw())+"\n")
```

```
# VQE q-UCCSD

var_form  = UCCSD(num_orbitals=n_orb,num_particles=n_part,
                  active_occupied=None,active_unoccupied=None,
                  initial_state=init_state,qubit_mapping=mapping,
                  two_qubit_reduction=tqr,num_time_slices=1,
                  z2_symmetries=z2syms)

optimizer = L_BFGS_B(maxiter=1000)
algo      = VQE(H_op,var_form,optimizer,aux_operators=A_op,
                                  include_custom=True)

backend          = Aer.get_backend('qasm_simulator')
quantum_instance = QuantumInstance(backend=backend)
algo_result      = algo.run(quantum_instance)
get_results(H_op,A_op,molecule,core,algo_result,outfile)
```

Ansatze and optimizers

- This script contains a calculation for H_2O at equilibrium geometry – how to do a HOMO-LUMO (highest-occupied, lowest-unoccupied molecular orbital) active space calculation?
- replace the q-UCCSD circuit with the R_y circuit and repeat the calculation
- in a recent publication (arXiv:1905.01534) McCaskey et al introduced a hardware-efficient Ansatz for 2-electrons in 2-orbitals applications. Try to implement it defining a custom variational form.
- explore the optimizer choice: do results change when e.g. COBYLA, SLSQP, CG are used in lieu of BFGS?

Other systems

- HeH^+ is a very small compound encountered in astrochemistry. Try computing its potential energy surface (i.e. energy versus bondlength) using the STO-6G basis set and the VQE Ansatz.
- LiH is a commonly studied compound. Try computing its potential energy surface (i.e. energy versus bondlength) using the STO-6G basis set and the VQE Ansatz, freezing the Li[1s] orbital and keeping orbitals in the A_1 irrep of the $C_{\infty v}$ symmetry group

Varia

- *warning* – Qiskit continues changing: it is therefore important to update programs so that they can run with recent versions and make sure deprecated objects are not used. The "Qiskit textbook"

  https://qiskit.org/textbook/preface.html

  is an important resource, and so are the "slack" channels (leading developers are Steve Wood and Manoel Marques)

References

[1] A. Szabo and N. S. Ostlund, *Modern quantum chemistry: introduction to advanced electronic structure theory.* Courier Corporation, 2012.

[2] T. Helgaker, P. Jorgensen, and J. Olsen, *Molecular electronic-structure theory.* John Wiley & Sons, 2014.

[3] P. E. Siegbahn, J. Almlöf, A. Heiberg, and B. O. Roos, "The complete active space SCF (CASSCF) method in a Newton–Raphson formulation with application to the HNO molecule," *The Journal of Chemical Physics*, vol. 74, no. 4, pp. 2384–2396, 1981.

[4] P. Pulay, "Convergence acceleration of iterative sequences. the case of SCF iteration," *Chemical Physics Letters*, vol. 73, no. 2, pp. 393–398, 1980.

[5] N. I. Levine, "Physical chemistry, 3^{rd} ed," *Chemistry Department Brooklyn College City University of New York*, pp. 371–373, 1988.

[6] M. Tinkham, *Group theory and quantum mechanics.* Courier Corporation, 2003.

[7] C. Davidson, "The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real symmetric matrices," *Journal of Computational Physics*, vol. 17, pp. 87–94, 1975.

[8] M. Head-Gordon and J. A. Pople, "Optimization of wave function and geometry in the finite basis Hartree-Fock method," *The Journal of Physical Chemistry*, vol. 92, no. 11, pp. 3063–3069, 1988.

[9] A. Tranter, S. Sofia, J. Seeley, M. Kaicher, J. McClean, R. Babbush, P. V. Coveney, F. Mintert, F. Wilhelm, and P. J. Love, "The bravyi–kitaev transformation: Properties and applications," *J. Chem. Phys*, vol. 115, no. 19, pp. 1431–1441, 2015.

[10] S. B. Bravyi and A. Y. Kitaev, "Fermionic quantum computation," *Annals of Physics*, vol. 298, no. 1, pp. 210–226, 2002.

[11] S. Bravyi, J. M. Gambetta, A. Mezzacapo, and K. Temme, "Tapering off qubits to simulate fermionic hamiltonians," *arXiv preprint arXiv:1701.08213*, 2017.

[12] J. Kempe, A. Kitaev, and O. Regev, "The complexity of the local hamiltonian problem," *SIAM Journal on Computing*, vol. 35, no. 5, pp. 1070–1097, 2006.

[13] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature communications*, vol. 5, no. 1, pp. 1–7, 2014.

[14] G. Harsha, T. Shiozaki, and G. E. Scuseria, "On the difference between variational and unitary coupled cluster theories," *The Journal of chemical physics*, vol. 148, no. 4, p. 044107, 2018.

[15] P. K. Barkoutsos, J. F. Gonthier, I. Sokolov, N. Moll, G. Salis, A. Fuhrer, M. Ganzhorn, D. J. Egger, M. Troyer, A. Mezzacapo, *et al.*, "Quantum algorithms for electronic structure calculations: Particle-hole hamiltonian and optimized wave-function expansions," *Physical Review A*, vol. 98, no. 2, p. 022322, 2018.

[16] F. A. Evangelista, G. K.-L. Chan, and G. E. Scuseria, "Exact parameterization of fermionic wave functions via unitary coupled cluster theory," *The Journal of chemical physics*, vol. 151, no. 24, p. 244112, 2019.

[17] I. O. Sokolov, P. K. Barkoutsos, P. J. Ollitrault, D. Greenberg, J. Rice, M. Pistoia, and I. Tavernelli, "Quantum orbital-optimized unitary coupled cluster methods in the strongly correlated regime: Can quantum algorithms outperform their classical equivalents?," *The Journal of chemical physics*, vol. 152, no. 12, p. 124107, 2020.

[18] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, no. 7671, pp. 242–246, 2017.