Simulating Quantum Drude Oscillators on a photonic quantum computer

Matthieu Sarkis*
Department of Physics and Materials Science
University of Luxembourg, L-1511,
Luxembourg City, Luxembourg.

Adel Sohbi†
ORCA

(Dated: December 18, 2022)

I. REMARKS AND QUESTIONS

Remarks and questions that should be addressed:

- Are alg. (1) or (3) the best since one needs to prepare the state twice to measure both the position quadratures and the photon numbers. Maybe a single measure in the coherent states basis would be possible? This can be achieved by a heterodyne measurement. With strawbery fields it can only be performed in the Gaussian or Bosonic backends, but not the tensorflow backend...
- What is the role of the parameter M in alg. (4)? Could one safely set $M = 1$ and still hope for convergence, a bit like in reinforcement learning? Setting $M = 1$ is of course a very rough estimate of the expected value, but since it is embedded in the training loop, maybe this estimate is actually enough?
- In strawberryfields the training procedure is managed by the tensorflow backend. Is it actually implementing the shift rule? Double check that.
- Maybe as a toy model of the toy model we can start by a single 1d harmonic oscillator

  $$H = \frac{1}{2}\left(p^2 + x^2\right) = a^\dagger a + \frac{1}{2} \tag{1}$$

  We can then test measuring separately both quadratures and compare with a single measurement of the photon number. The correct statevector is of course the Fock vacuum in that case. For me the problem is more related to the strawberryfields API than conceptual. Ideally I'd rather perform a single heterodyne measurement on each channel, the cost function is then simply given by (21).
- It seems that with the tf backend, one has to perform multiple shots manually, i.e. the 'shots' argument of the 'run' method is not implemented.
- I had problem related to the fact that the measurement process is non-differentiable. Then I checked at the paper of Xanadu on state preparation: in some sense they are cheating. They are computing the loss directly from the statevector, and they have access to the statevector because they are working on the simulator. On a real photonic device, one doesn't have access to the statevector directly, but only to the outcome of measurements. I propose that we bypass the issue by cheating the same way they did, unless some smart ORCA employee comes up with a nice solution haha. The only solution I could imagine for now is the following: getting inspiration from the 'reparameterization trick' using in the context of autoencoders. In that context one also has to backpropagate through a sampling operation from a probability distribution. In that case, the problem is solved because we know that the distribution is actually Gaussian, and therefore we can parameterize it in terms of its mean and variance, and differentiate with respect to those parameters. In our case however, the probability distribution defined by the ansatz state and the observable is of course not known generally, so it makes it difficult to use this trick...

Things to be done:

- Generalization to 3d: just more harmonic oscillators basically
- Higher orders in the multipolar expansion

II. INTRODUCTION

III. DEFINITION OF THE MODEL

The Hamiltonian describing a system of N QDOs in 3d is given by:

$$H = \sum_{i=1}^{N}\left[\frac{\boldsymbol{p}_i^2}{2m_i} + \frac{1}{2}m_i\omega_i^2\boldsymbol{x}_i^2\right] + \sum_{i<j} V_{\text{Coul}}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) , \tag{2}$$

with the Coulomb interaction receiving contributions from every pair of constituents (centers and point parti-

* matthieu.sarkis@uni.lu
† sohbi@kias.re.kr

cles):

$$\frac{V_{\text{Coul}}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)}{q_i q_j (4\pi\epsilon_0)^{-1}} = \frac{1}{r} - \frac{1}{|\boldsymbol{r}+\boldsymbol{x}_i|} - \frac{1}{|\boldsymbol{r}-\boldsymbol{x}_j|} + \frac{1}{|\boldsymbol{r}-\boldsymbol{x}_j+\boldsymbol{x}_i|} \tag{3}$$

In the multipolar expansion, this can be expressed as a power series in the inverse distance separating the two centers:

$$V_{\text{Coul}}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \sum_{n\geq 0} V_n\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \,, \tag{4}$$

with the following scaling behavior in terms of the distance between the centers:

$$V_n\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \propto r_{ij}^{-n-3} \,. \tag{5}$$

A. One-dimensional case

We consider the one dimensional system in which the electrons are constrained to move either in the direction parallel to the axis separating the two nuclei, or perpendicular to the latter.

$$V_0(x_i, x_j) = \frac{q_i q_j}{4\pi\epsilon_0} \frac{x_i x_j}{r_{ij}^3} \times \begin{cases} -2, & \text{parallel case} \\ 1, & \text{perpendicular case} \end{cases} \tag{6}$$

We define

$$\gamma_{ij} := \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}^3} \times \begin{cases} -2, & \text{parallel case} \\ 1, & \text{perpendicular case} \end{cases} \tag{7}$$

We therefore reach the following one dimensional model for N QDOs in the dipolar approximation:

$$\begin{aligned} H &= \sum_{i=1}^{N} \left(\frac{p_i^2}{2m_i} + \frac{1}{2} m_i \omega_i^2 x_i^2 \right) + \sum_{i<j} \gamma_{ij} x_i x_j \\ &= \sum_{i=1}^{N} \hbar\omega_i \left(a_i^\dagger a_i + \frac{1}{2} \right) + \sum_{i<j} \gamma_{ij} x_i x_j \\ &= \sum_{i=1}^{N} \hbar\omega_i \left(a_i^\dagger a_i + \frac{1}{2} \right) \\ &\quad + \frac{\hbar}{2} \sum_{i<j} \frac{\gamma_{ij}}{\sqrt{m_i m_j \omega_i \omega_j}} \left(a_i + a_i^\dagger \right) \left(a_j + a_j^\dagger \right) \end{aligned} \tag{8}$$

with the annihilation and creation operators:

$$\begin{aligned} a_i &= \sqrt{\frac{m_i\omega_i}{2\hbar}}\, x_i + \frac{i}{\sqrt{2\hbar m_i \omega_i}}\, p_i \,, \\ a_i^\dagger &= \sqrt{\frac{m_i\omega_i}{2\hbar}}\, x_i - \frac{i}{\sqrt{2\hbar m_i \omega_i}}\, p_i \,, \end{aligned} \tag{9}$$

and where the from the second line we have symmetrized the Hamiltonian and used the canonical commutation relation. Just in case, let me write down the next terms in the multipolar expansion:

$$V_1(x_i, x_j) = \frac{q_i q_j}{4\pi\epsilon_0} \frac{x_i x_j (x_i - x_j)}{r_{ij}^4} \times \begin{cases} 3, & \text{parallel case} \\ 0, & \text{perpendicular case} \end{cases} \tag{10}$$

$$\begin{aligned} V_2(x_i, x_j) = & \frac{q_i q_j}{4\pi\epsilon_0} \frac{x_i x_j (2x_i^2 - 3x_i x_j + 2x_j^2)}{r_{ij}^5} \\ & \times \begin{cases} -2, & \text{parallel case} \\ -\frac{3}{4}, & \text{perpendicular case} \end{cases} \end{aligned} \tag{11}$$

even though we will probably focus on the dipolar approximation.

For simplicity in the following, we will perform the following simplifications:

- All the masses are equal to a common value m, that we set equal to 1.
- All the frequences are equal to a common value ω, that we set equal to 1.
- All the charges are equal to a common value q, that we set equal to 1.
- We also set $\hbar = 1$ and $\frac{1}{4\pi\epsilon_0} = 1$.

The model therefore reads:

$$H = \sum_{i=1}^{N} \left(a_i^\dagger a_i + \frac{1}{2} \right) + \frac{1}{2} \sum_{i<j} \gamma_{ij} \left(a_i + a_i^\dagger \right) \left(a_j + a_j^\dagger \right) \,. \tag{12}$$

In the case of two QDOs we have:

$$H = a_1^\dagger a_1 + a_2^\dagger a_2 + \frac{\gamma}{2} \left(a_1 + a_1^\dagger \right) \left(a_2 + a_2^\dagger \right) + 1 \,, \tag{13}$$

with

$$\gamma := \frac{1}{r^3} \times \begin{cases} -2, & \text{parallel case} \\ 1, & \text{perpendicular case} \end{cases} \tag{14}$$

where r is the distance between the two nuclei.

B. Three-dimensional case

1. *Multipolar expansion*

The dipole-dipole potential in 3d reads:

$$V_0(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{\boldsymbol{x}_i \cdot \boldsymbol{x}_j - 3(\boldsymbol{x}_i \cdot \boldsymbol{n}_{ij})(\boldsymbol{x}_j \cdot \boldsymbol{n}_{ij})}{r_{ij}^3} = \boldsymbol{x}_i \mathbb{T}_{ij} \boldsymbol{x}_j \,, \tag{15}$$

where $r_{ij}\boldsymbol{n}_{ij}$ is the vector connecting nuclei j to nuclei i, and we defined the tensor

$$\mathbb{T}_{ij} = \frac{\mathbb{I}_3 - \boldsymbol{n}_{ij} \otimes \boldsymbol{n}_{ij}}{r_{ij}^3} \,. \tag{16}$$

In the case of 2 QDOs sitting along the z-axis and separated by a distance r:

$$V_0(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{x_1 x_2 + y_1 y_2 - z_1 z_2}{r^3} \,. \tag{17}$$

We therefore need free channels per QDO, so a total of 6 channels.

C. Full Coulomb potential

We still consider the case of 2 QDOs, but this time we treat the full Coulomb potential (3). We take the two nuclei to be aligned along the z-axis and separated by a distance r. The potential can be written as

$$\begin{aligned} V_{\mathrm{Coul}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{1}{r} - \frac{1}{(r^2 + x_1^2 + y_1^2 + z_1^2 + 2rz_1)^{\frac{1}{2}}} \\ - \frac{1}{(r^2 + x_2^2 + y_2^2 + z_2^2 - 2rz_2)^{\frac{1}{2}}} \\ + \frac{1}{(r^2 + (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 - 2r(z_2 - z_1))^{\frac{1}{2}}} \end{aligned} \tag{18}$$

IV. PHOTONIC CIRCUIT

The circuit implements a unitary $U(\theta)$ acting on an input reference state (the Fock vacuum for instance) that we simply take to be the vacuum state $|0\rangle$. The state prepared by the circuit is therefore given by

$$|\psi(\theta)\rangle = U(\theta)|0\rangle \,. \tag{19}$$

Since we are working in the dipolar approximation, we expect that using a Gaussian state would be enough. The circuit is therefore composed of at most quadratic optical components (squeezing operations for our ansatz). A non-Gaussian operation can optionally be added in the end of each layer in the ansatz circuit.

V. VARIATIONAL ALGORITHM

We define the following loss function:

$$\mathcal{C}(\theta) := \langle\psi(\theta)|H|\psi(\theta)\rangle \tag{20}$$

with the Hamiltonian defined in eq. (12). In order to compute this loss, one therefore has to measure both the photon number operator on each channel, as well as the position quadrature operator on each channel.

Algorithm 1: Computation of the energy using photon numbers and quadratures

```
Parameters: reference statevector $|0\rangle$, circuit $U$
Result: Value of the energy $E$

Prepare statevector $|\psi\rangle = U|0\rangle$;
Measure the position quadratures $x_i$;
Prepare statevector $|\psi\rangle = U|0\rangle$;
Measure the photon numbers $n_i$;
Compute the energy $E$ with eq. (12);
return $E$.
```

Algorithm 2: Computation of the energy using coherent state basis

```
Parameters: reference statevector $|0\rangle$, circuit $U$
Result: Value of the energy $E$

Prepare statevector $|\psi\rangle = U|0\rangle$;
Perform heterodyne measurement on each channel to
  get $\alpha_i$;
Compute the energy $E$ with eq. (12);
return $E$.
```

Algorithm 3: Computation of the energy using quadratures

```
Parameters: reference statevector $|0\rangle$, circuit $U$
Result: Value of the energy $E$

Prepare statevector $|\psi\rangle = U|0\rangle$;
Perform homodyne measurement on each channel to
  get the position quadratures $x_i$;
Prepare statevector $|\psi\rangle = U|0\rangle$;
Perform homodyne measurement on each channel to
  get the momentum quadratures $p_i$;
Compute the energy $E$ with eq. (12);
return $E$.
```

After measuring the amplitude α_i on each channel with heterodyne detections, the measured energy reads

$$E = \sum_{i=1}^{N} \left(|\alpha_i|^2 + \frac{1}{2} \right) + 2 \sum_{i<j} \gamma_{ij} \mathrm{Re}(\alpha_i) \mathrm{Re}(\alpha_j) \tag{21}$$

The expected value of the energy in state $|\psi\rangle$ is obtained by averaging the result of $M \in \mathbb{N}$ such measurements:

$$\langle\psi|H|\psi\rangle = \frac{1}{M} \sum_{j=1}^{M} E_j + \mathcal{O}\left(\frac{1}{\sqrt{M}} \right) \tag{22}$$

A very rough estimate would consist in setting $M = 1$.

Algorithm 4: Computation of the loss

```
Parameters: $M \in \mathbb{N}$
Result: Value of the loss $\mathcal{C}$

Initialize $\mathcal{C} \leftarrow 0$;
for $j = 1$ to $M$ do
    Compute the energy $E$ with alg. (1), (2) or (3);
    Update the loss $\mathcal{C} \leftarrow \mathcal{C} + E$;
end for
$\mathcal{C} \leftarrow \mathcal{C}/M$;
return $\mathcal{C}$.
```

Algorithm 5: Training of the parameterized photonic circuit

```
Parameters: $N_{\text{steps}} \in \mathbb{N}$, initial parameters $\theta_0 \in \mathbb{R}^K$,
  learning rate $\eta \in \mathbb{R}_+$
Result: Optimized hyperparameters $\theta \in \mathbb{R}^K$

Initialize hyperparameters $\theta \leftarrow \theta_0$;
for $i = 1$ to $N_{steps}$ do
    Compute the loss $\mathcal{C}$ with alg. (4);
    Compute the gradient $\nabla_\theta \mathcal{C}$ with shift rule and alg.
      (4);
    Update the parameters $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{C}$;
end for
return $\theta$.
```

VI. RESULTS

VII. CONCLUSION

ACKNOWLEDGMENTS

CODE AVAILABILITY

The reader will find an open source python code accompanying this paper following this github repository.

[1] L. W. Anderson, M. Kiffner, P. K. Barkoutsos, I. Tavernelli, J. Crain, and D. Jaksch, "Coarse-grained intermolecular interactions on quantum processors," *Physical Review A*, vol. 105, no. 6, p. 062409, 2022.

[2] J. M. Arrazola, T. R. Bromley, J. Izaac, C. R. Myers, K. Brádler, and N. Killoran, "Machine learning method for state preparation and gate synthesis on photonic quantum computers," *Quantum Science and Technology*, vol. 4, no. 2, p. 024004, 2019.

[3] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, "Continuous-variable quantum neural networks," *Physical Review Research*, vol. 1, no. 3, p. 033063, 2019.