Efficient evaluation of exponential and Gaussian functions on a quantum computer

Bill Poirier
Texas Tech University
Department of Chemistry & Biochemistry
Goals and Motivations

• Project grew out of my main interest in quantum computing, i.e. quantum computational chemistry.

• Most QCC efforts focus on second-quantized methods.
  – rely on Hartree-Fock-based Slater determinant representations.
  – zeroth-order picture: wavefunctions describing a collection of electrons are separable by particle.

• Limitations of second-quantized methods:
  – not always reliably accurate for strongly correlated systems.
  – generally not designed to compute many excited states (especially not wavefunctions).
  – not extendible to combined electron-nuclear problem!!
First-Quantized QCC

• In the long term, the most effective QCC algorithms are likely to be *first-quantized* strategies.*
  - Compute exact, full $3n$-dimensional electronic wavefunction, where $n$ is the number of electrons.
  - Electron correlation poses no special difficulty.
  - Combined electron-nuclear motion not a problem (in principle).

• However, qubits and quantum gate depths are still beyond present-day quantum hardware.
  - e.g. Li atom requires 100+ logical qubits;* whereas Honeywell ytterbium ion trap QC (currently) has only 50.

• Comparatively little first-quantized QCC development has happened to date.
  - lack of established framework. *creativity/new ideas are needed!*

Our First-Quantized Approach: “Cartesian-Separated” or Andromeda

• There are no Slater determinants.
  - antisymmetry dealt with through other means.
• Zeroth-order picture: representations are separable by *Cartesian component* [i.e, \((x,y,z)\)] rather than by *particle*.
• Usefulness in *bona fide* QCC:
  - all advantages of first-quantized approach realized.
  - number of qubits reduced, in comparison with other standard approaches.
  - recent QCC algorithm exhibits reduced quantum gate depth, in comparison with other standard approaches.
• Usefulness as a *classical* algorithm for *simulating* QCC:
  - can provide important benchmarks for QCC requirements *a priori*.
  - e.g., for Li atom calculation, using our representation only \(~60\) qubits needed.
Classical Publications


Quantum Publications


Andromeda Code

https://github.com/JonathanJerke/Andromeda

Provisional Patent

METHODS AND SYSTEMS FOR QUANTUM COMPUTATIONAL CHEMISTRY MODELING (4/15/20, Lionel W. Poirier, TTU System)
Cartesian-Separated Approach

Representing the two-body electron pair repulsion, $V_{ee}$

The basic “trick” is the following modified Laplace transform

$$\frac{1}{|\vec{r}_1 - \vec{r}_2|} = \int_0^\infty \frac{2}{\sqrt{\pi}} \prod_{d=1}^{3} e^{-\beta^2 (r_{d1} - r_{d2})^2} \, d\beta$$

which replaces the six-dimensional function $V_{ee}(x_1, y_1, z_1, x_2, y_2, z_2)$ with the two-dimensional Gaussian function, $\exp[-\beta^2(x_1-x_2)^2]$.

**The Gaussian function evaluation requires only two qubit wires, and is a much better behaved function.**
RB: “While there are possibly some advantages from computing the potential in this different fashion, as I've said it is not entirely clear how significant that alone will turn out to be since the essential bottleneck in current way of doing things...is computation of the inverse square root, and here the bottleneck will likely be the computation of the exponential.”

BP: “In your opinion, is this a fundamental limitation, or do you hold out hope that someone may **come up with a clever implementation** for the inverse square-root or exponential **that could radically improve matters**?”

RB: “I do expect that somebody could come up with better implementations for the exponential or inverse square root...that could potentially reverse the relative difficulty of these operations. However, I am fairly certain that those operations will remain dramatically more expensive than operations like squaring or addition...I am rather certain that the computation [of exponential or inverse square root] is always going to be orders of magnitude more expensive.

This is no longer the case!! Ryan is now also convinced.
Exponential Function Algorithm

• Consider a grid of \( k_{\text{max}} = 2^d \) uniformly spaced points in \( x \).
  \[ x_k = \Delta k, \text{ where } k = \{0,1,\ldots,2^d-1\}, \text{ and } \Delta \text{ is the grid spacing.} \]
  \( d \) is associated with resolution of the domain (we use \( d=7 \)).
  \textit{note:} shifting the coordinate interval is equivalent to rescaling the exponential function, so above can be used WLOG.

• We wish to evaluate the function \( f(x) = C \exp(-\alpha x) \)
  - at any/all of the grid points, \( x_k \), for arbitrary \( C \) and \( \alpha \).
  - let \( f_k = f(x_k) = C \exp(-\alpha \Delta k) = CA^k \), where \( A = \exp(-\alpha \Delta) \).

• Next, we apply a binary expansion of the integer \( k \):
  - i.e., \( k = \sum_{i=0}^{d-1} a_i 2^i \), with \( a_i = \{0,1\} \).
  - this yields \( f_k = CA\sum_{i=0}^{d-1} a_i 2^i = C \times (A^{2_0})^{a_0} \times (A^{2_1})^{a_1} \times \ldots \times (A^{2^{d-1}})^{a_{d-1}} \)
  - or \( f_k = C \times A_0^{a_0} \times A_1^{a_1} \times \ldots \times A_{d-1}^{a_{d-1}} \), with \( A_i = A^{2^i} \).
Exponential Function Algorithm

\[ f_k = C \times A_0^{a_0} \times A_1^{a_1} \times \cdots \times A_{d-1}^{a_{d-1}} \]

Now, because each factor, \( A_i^{a_i} \) has an exponent \( a_i \) whose value is either 0 or 1, this can be interpreted as an instruction:

“if \( a_i = 1 \), then multiply by \( A_i \); otherwise, do nothing.”

This suggests a quantum computing implementation as a sequence of conditional multiplications, with no more than \( d \) such conditional multiplications needed in all.
Exponent’l
Function
Algorithm

Comments:

1. Only $d=\log[k_{\text{max}}]$ total multiplications are required, with $d=7$ or so (i.e., $k_{\text{max}}>100$) sufficient for QCC. Greater domain resolution requires larger $d$, and therefore more ancilla bundles ($d$ total) and quantum gates. If “machine precision” in the domain is needed, $d=25-30$ is required.
Exponent’l Function Algorithm

Comments:

2. Greater *range* resolution requires more qubits in each ancilla bundle—with 25-30 again needed for typical “machine precision.”

3. The exponential function can easily span many orders of magnitude. Thus, not all $A_i$ might be needed, depending on the $x$ interval and range resolution.
4. The range of output values could be better represented by rescaling $C$. Conversely, *Floating point*-type representations of $A_i$ could also be used.

5. This would lead to a quite different multiplication routine, which in any event we prefer to leave a “black box” (e.g. to accommodate error correction, etc.)
6. Many strategies could potentially be used for reducing the qubits needed to store the $d$ ancilla bundles:

   a) use of QROM and/or “dirty ancillae.”

   b) exploit multiplicative relationships among $A_i$, so that only some need be stored, and the rest computed.

   c) “engineering trade-off” between # of qubits & quantum gates.