The Journal of Open Source Software

The Walrus: a library for the calculation of hafnians, Hermite polynomials and Gaussian boson sampling

Brajesh Gupt[1], **Josh Izaac**[1], **and Nicolás Quesada**[1]

1 Xanadu, Toronto, Canada

DOI: 10.21105/joss.01705

Software

- Review
- Repository
- Archive

Editor: Kathryn Huff

Reviewers:

- @amitkumarj441
- @poulson

Submitted: 19 August 2019
Published: 19 December 2019

In The Walrus, we provide a highly optimized implementation of the best known algorithms for hafnians, loop hafnians, multidimensional Hermite polynomials, and torontonians of generic real and complex matrices. We also provide access to recently proposed methods to generate samples of a Gaussian boson sampler. These methods have exponential time complexity in the number of bosons measured. For ease of use, a Python interface to the library's low-level C++ implementations is also provided, as well as pre-compiled static libraries installable via the Python package manager `pip`. This short paper provides a high-level description of the library and its rationale; in-depth information on the algorithms, implementations and interface can be found in its documentation.

The hafnian matrix function was introduced by Caianiello (1953) as a generalization of the permanent while studying problems in bosonic quantum field theory. For a symmetric matrix $\mathbf{A} = \mathbf{A}^T$ of size $2n$, the hafnian (haf) is defined as

$$\text{haf}\,(\mathbf{A}) = \sum_{\sigma \in \text{PMP}(2n)} \prod_{i=1}^{n} A_{\sigma(2i-1),\sigma(2i)},$$

where $\text{PMP}(2n)$ is the set of perfect matching permutations of $2n$ elements, i.e., permutations $\sigma : [2n] \to [2n]$ such that $\sigma(2i-1) < \sigma(2i)$ and $\sigma(2i-1) < \sigma(2i+1)$ for all i (Barvinok, 2016).

While the permanent counts the number of perfect matchings of a *bipartite* graph encoded in an adjacency matrix $\mathbf{B}$, the hafnian counts the number of perfect matchings of an *arbitrary undirected graph*, and thus the permanent is a special case of the hafnian; this relation is encapsulated in the identity $\text{perm}(\mathbf{B}) = \text{haf}\left(\left[\begin{smallmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{smallmatrix}\right]\right)$.

The permanent has received a significant amount of attention, especially after Valiant (1979) proved that it is #P-hard to compute, giving one of the first examples of a problem in this complexity class. This important complexity-theoretic observation was predated by Ryser (1963), who provided an algorithm to calculate the permanent of an arbitrary matrix of size $n \times n$ in $O(2^n)$ time, which is still to date the fastest algorithm for calculating permanents.

Surprisingly, it took almost half a century to derive a formula for hafnians that matched the complexity of the one for permanents. Indeed, it was only Björklund (2012) who derived an algorithm that computed the hafnian of a $2n \times 2n$ matrix in time $O(2^n)$.

The interest in hafnians was recently reignited by findings in quantum computing. Gaussian Boson Sampling (GBS) (Hamilton et al., 2017; Kruse et al., 2019) is a non-universal model of quantum computation in which it is possible to show that there are computations a quantum computer can do in polynomial time that a classical computer cannot. Experimentally, GBS is based on the idea that a certain subset of quantum states, so-called Gaussian states, can be easily prepared in physical devices, and then those states can be measured with particle-number resolving detectors. Because these are quantum mechanical particles, the outcomes of the measurements are stochastic in nature and it is precisely the simulation of these random "samples" that requires superpolynomial time to simulate on a classical computer.

The relation between GBS and hafnians stems from the fact that the probability of a given experimental outcome is proportional to the hafnian of a matrix constructed from the covariance matrix of the Gaussian state. This observation requires the Gaussian state to have zero mean and the detector to be able to resolve any number of particles. More generally, one can consider Gaussian states with finite mean (N Quesada, 2019; N. Quesada et al., 2019), in which the probability is given by a loop hafnian, a matrix function that counts the number of perfect matchings of a graph that has loops (Björklund, Gupt, & Quesada, 2019). Moreover, if the particle detectors can only decide whether there were zero or more than zero particles – so-called threshold detectors – then the probability is given by the torontonian, a matrix function that acts as a generating function for the hafnian (Quesada, Arrazola, & Killoran, 2018). One can also show that the probabilities of a Gaussian state probed in the number basis are related to multidimensional Hermite polynomials (Dodonov, Man'ko, & Man'ko, 1994). Calculating the probabilities of a GBS experiment in terms of multidimensional Hermite polynomials (Kok & Braunstein, 2001) is often suboptimal since they have worse space and time scaling than the corresponding calculation in terms of hafnians.

In The Walrus, we provide a highly optimized implementation of the best known algorithms for hafnians, loop hafnians, Hermite polynomials, and torontonians of generic real and complex matrices. We also implement algorithms that specialize to certain matrices with structure, for example having repeated rows and columns (Kan, 2008) or non-negative entries (Barvinok, 1999). For increased efficiency, these algorithms are implemented in C++ as a templated header-only library, allowing them to be applied to arbitrary numeric types and are also parallelized via OpenMP. Common linear algebra algorithms are applied using the Eigen C++ template library, which may also act as a frontend to an optimized BLAS/LAPACKE library installation if the user so chooses. For ease of use, a Python interface to the low-level C++ algorithms is also provided, as well as pre-compiled static libraries installable via the Python package manager `pip` for Windows, MacOS, and Linux users. We also provide implementations of multidimensional Hermite polynomials, that are, to the best of our knowledge, the first ones implemented in a fast open source library. With this underlying machinery we also provide two extra Python-only modules. The first one, *quantum*, allows one to calculate in a straightforward manner the probabilities or probability amplitudes of Gaussian states in the particle representation. The second one, *samples*, allows one to generate GBS samples. This module implements state-of-the-art algorithms that have been recently developed (Nicolas Quesada & Arrazola, 2019). Of course, given the promise that GBS should be a hard problem for classical computers, the complexity of the algorithm we provide for GBS is, like the complexity of the hafnian, still exponential in the size of the number of particles generated.

Our package has already been used in several research efforts to understand how to generate resource states for universal quantum computing (N. Quesada et al., 2019; Tzitrin, Bourassa, Menicucci, & Sabapathy, 2019), study the dynamics of vibrational quanta in molecules (N Quesada, 2019; Valson Jacob, Kaur, Roga, & Takeoka, 2019), and develop the applications of GBS (Bromley et al., 2019) to molecular docking (Banchi, Fingerhuth, Babej, & Arrazola, 2019), graph theory (Schuld, Brádler, Israel, Su, & Gupt, 2019), and point processes (Jahangiri, Arrazola, Quesada, & Killoran, 2019). More importantly, it has been useful in delineating when quantum computation can be simulated by classical computing resources and when it cannot (Gupt, Arrazola, Quesada, & Bromley, 2018; Killoran et al., 2019; Nicolas Quesada & Arrazola, 2019; Wu, Cheng, Zhang, Yung, & Sun, 2019).

Acknowledgements

The authors thank our colleagues at Xanadu, especially, J.M. Arrazola, T.R. Bromley, S. Jahangiri and N. Killoran, for valuable feedback.

N.Q. thanks A. Björklund and W.R. Clements for insightful correspondence and C. Ducharme and L.G. Helt for comments on the manuscript.

References

Banchi, L., Fingerhuth, M., Babej, T., & Arrazola, J. M. (2019). Molecular docking with Gaussian boson sampling. *arXiv preprint arXiv:1902.00462*.

Barvinok, A. (1999). Polynomial time algorithms to approximate permanents and mixed discriminants within a simply exponential factor. *Random Structures & Algorithms*, *14*(1), 29–61. doi:10.1002/(SICI)1098-2418(1999010)14:1<29::AID-RSA2>3.0.CO;2-X

Barvinok, A. (2016). *Combinatorics and complexity of partition functions* (Vol. 274). Springer. doi:10.1007/978-3-319-51829-9

Björklund, A. (2012). Counting perfect matchings as fast as Ryser. In *Proceedings of the twenty-third annual acm-siam symposium on discrete algorithms* (pp. 914–921). doi:10.1137/1.9781611973099.73

Björklund, A., Gupt, B., & Quesada, N. (2019). A faster hafnian formula for complex matrices and its benchmarking on a supercomputer. *Journal of Experimental Algorithmics (JEA)*, *24*(1), 11. doi:10.1145/3325111

Bromley, T. R., Arrazola, J. M., Jahangiri, S., Izaac, J., Quesada, N., Delgado Gran, A., Schuld, M., et al. (2019). Applications of near-term photonic quantum computers: Software and algorithms. *arXiv preprint arXiv:1912.07634*.

Caianiello, E. R. (1953). On quantum field theory—i: Explicit solution of Dyson's equation in electrodynamics without use of Feynman graphs. *Il Nuovo Cimento (1943-1954)*, *10*(12), 1634–1652. doi:10.1007/BF02781659

Dodonov, V., Man'ko, O., & Man'ko, V. (1994). Multidimensional Hermite polynomials and photon distribution for polymode mixed light. *Physical Review A*, *50*(1), 813. doi:10.1103/PhysRevA.50.813

Gupt, B., Arrazola, J. M., Quesada, N., & Bromley, T. R. (2018). Classical benchmarking of Gaussian boson sampling on the Titan supercomputer. *arXiv preprint arXiv:1810.00900*.

Hamilton, C. S., Kruse, R., Sansoni, L., Barkhofen, S., Silberhorn, C., & Jex, I. (2017). Gaussian boson sampling. *Physical Review Letters*, *119*(17), 170501. doi:10.1103/PhysRevLett.119.170501

Jahangiri, S., Arrazola, J. M., Quesada, N., & Killoran, N. (2019). Point processes with Gaussian boson sampling. *arXiv preprint arXiv:1906.11972*.

Kan, R. (2008). From moments of sum to moments of product. *Journal of Multivariate Analysis*, *99*(3), 542–554. doi:10.1016/j.jmva.2007.01.013

Killoran, N., Izaac, J., Quesada, N., Bergholm, V., Amy, M., & Weedbrook, C. (2019). Strawberry fields: A software platform for photonic quantum computing. *Quantum*, *3*, 129. doi:10.22331/q-2019-03-11-129

Kok, P., & Braunstein, S. L. (2001). Multi-dimensional Hermite polynomials in quantum optics. *Journal of Physics A: Mathematical and General*, *34*(31), 6185. doi:10.1088/0305-4470/34/31/312

Kruse, R., Hamilton, C. S., Sansoni, L., Barkhofen, S., Silberhorn, C., & Jex, I. (2019). Detailed study of Gaussian boson sampling. *Physical Review A*, *100*(3), 032326. doi:10.1103/PhysRevA.100.032326

Quesada, N. (2019). Franck-Condon factors by counting perfect matchings of graphs with loops. *The Journal of Chemical Physics*, *150*(16), 164113. doi:10.1063/1.5086387

Quesada, N., & Arrazola, J. M. (2019). The classical complexity of Gaussian boson sampling. *arXiv preprint arXiv:1908.08068*.

Quesada, N., Arrazola, J. M., & Killoran, N. (2018). Gaussian boson sampling using threshold detectors. *Physical Review A*, *98*(6), 062322. doi:10.1103/PhysRevA.98.062322

Quesada, N., Helt, L. G., Izaac, J., Arrazola, J. M., Shahrokhshahi, R., Myers, C. R., & Sabapathy, K. K. (2019). Simulating realistic non-Gaussian state preparation. *Physical Review A*, *100*(2), 022341. doi:10.1103/PhysRevA.100.022341

Ryser, H. J. (1963). *Combinatorial mathematics*. Mathematical Association of America; distributed by Wiley [New York]. doi:10.1017/S0013091500011299

Schuld, M., Brádler, K., Israel, R., Su, D., & Gupt, B. (2019). A quantum hardware-induced graph kernel based on Gaussian boson sampling. *arXiv preprint arXiv:1905.12646*.

Tzitrin, I., Bourassa, J. E., Menicucci, N. C., & Sabapathy, K. K. (2019). Towards practical qubit computation using approximate error-correcting grid states. *arXiv preprint arXiv:1910.03673*.

Valiant, L. G. (1979). The complexity of computing the permanent. *Theoretical computer science*, *8*(2), 189–201. doi:10.1016/0304-3975(79)90044-6

Valson Jacob, K., Kaur, E., Roga, W., & Takeoka, M. (2019). Franck-Condon factors via compressive sensing. *arXiv preprint arXiv:1909.02935*.

Wu, B., Cheng, B., Zhang, J., Yung, M.-H., & Sun, X. (2019). Speedup in classical simulation of Gaussian boson sampling. *arXiv preprint arXiv:1908.10070*.