Thermodynamic matrix exponentials and thermodynamic parallelism

Samuel Duffield,* Maxwell Aifer,* Gavin Crooks, Thomas Ahle, and Patrick J. Coles
Normal Computing Corporation, New York, New York 10011, USA

(Received 5 January 2024; accepted 22 January 2025; published 10 February 2025)

Thermodynamic computing exploits fluctuations and dissipation in physical systems to efficiently solve various mathematical problems. It was recently shown that certain linear algebra problems can be solved thermodynamically, leading to a speedup scaling with the matrix dimension. Here, we provide a thermodynamic algorithm for exponentiating a real matrix. We describe a simple electrical circuit involving coupled oscillators, which can implement our algorithm. We also show that this algorithm provides an asymptotic speedup that is linear in the dimension. Finally, we introduce the concept of thermodynamic parallelism to explain this speedup, stating that thermodynamic noise provides a resource leading to effective parallelization of computations.

DOI: 10.1103/PhysRevResearch.7.013147

I. INTRODUCTION

The exponential of a matrix plays a central role in the study of linear differential equations. A real matrix $A \in \mathbb{R}^{d\times d}$ specifies a homogeneous linear differential equation $\mathrm{d}x = -Ax\,\mathrm{d}t$, whose solution may be written

$$x(t) = e^{-At}x(0), \tag{1}$$

where e^{-At} can be defined in many equivalent ways (see Table I). Examples occur in simulating classical and quantum physical systems, processing audio and video signals [1], analyzing economic time series data [2], and more recently in machine learning (e.g., reinforcement learning [3] and large language models [4]). Linear differential equations also appear in the analysis of Markov processes, whose applications include queuing theory [5], biology [6], and rating systems [7].

Due to their important and diverse applications, there has been intense research activity on the efficient computation of matrix exponentials in the last century [8,9], including physics-inspired digital algorithms that utilize the structure of the input matrix in the context of classical [10] or quantum Monte Carlo [11,12]. The most popular numerical methods are diagonalization and Padé approximation [13,14], which have time complexity $O(d^3)$ or $O(d^\omega)$ for general matrices.[1] The matrix exponential can also be found by solving the initial value problem (IVP) $\mathrm{d}x = -Ax\,\mathrm{d}t$, $x(0) = \hat{e}_i$ for each standard basis vector $\hat{e}_i$. This approach also requires $O(d^3)$ operations, as each IVP can be solved in $O(d^2)$ operations, but it can be parallelized by allocating one thread for each IVP, thus taking $O(d^2)$ time if you have d processors. However, parallelizing to d processors can be challenging. A modern A100 GPU has 6912 cores [16] which allows highly parallel algorithms [17,18], but adding more cores is difficult due to heat dissipation and the complexity of intercore communication.

A variety of recent work has investigated the use of physical processes as a resource for computation. This includes Ising machines [19], probabilistic bits [20], coupled oscillators [21], neuromorphic circuits [22], and memristor crossbars [23,24].

More specifically, an analog device such as an electrical circuit [25] can encode the (deterministic) differential equation in its voltages. However, analog devices face serious challenges due to their susceptibility to errors and the difficulty of scaling the necessary hardware to large problem sizes. For example, as illustrated in Fig. 1(b), implementing the matrix exponentiation method described above in an analog way would require d separate analog devices, each capable of solving a d-dimensional linear differential equation (each device would therefore need to store the d^2 elements of A). Moreover, each device would be influenced by various types of errors, including noise caused by thermal fluctuations. See the Supplemental Material [26] for a full analysis of such an analog device under natural noise assumptions.

TABLE I. Some ways to define and compute e^A [27].

Power series	Limit
$I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots$	$\lim_{s\to\infty}(I + \frac{A}{s})^s$
Differential system	Schur form, $A = QTQ^T$
$X'(t) = AX(t), X(0) = I$	$Q\,\mathrm{diag}(e^T)Q^*$
Cauchy integral	Padé approximation
$\frac{1}{2\pi i}\oint_\Gamma e^z(zI - A)^{-1}dz$	$(I + \frac{A}{2} + \cdots)(I - \frac{A}{2} + \cdots)^{-1}$

*These authors contributed equally to this work.

[1] Here, $\omega \approx 2.371\,552$ is the fast matrix multiplication constant [15].

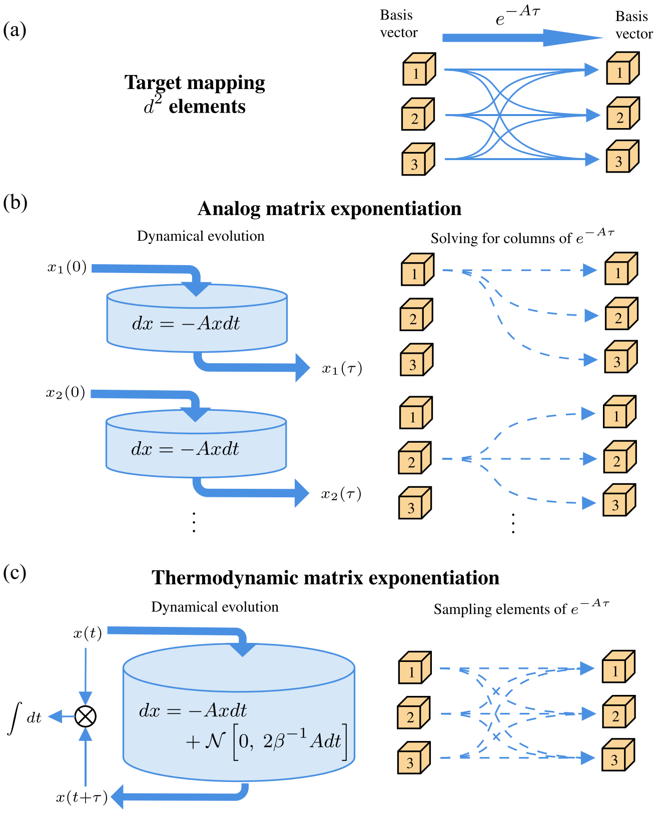

FIG. 1. Analog and thermodynamic approaches. (a) The target matrix $e^{-A\tau}$ has d^2 elements that must be experimentally characterized. An individual experiment can be viewed as sampling these elements from a distribution, due to inherent randomness in the experiment (either unintentional in the analog case or intentional in the thermodynamic case). (b) A naive analog approach would sample d elements at a time, while (c) the thermodynamic approach samples all d^2 elements at once by using noise as a resource, an advantage we call thermodynamic parallelism.

In this paper, we show how to evaluate matrix exponentials on a thermodynamic computing [28–36] device with d cells, using a *stochastic* differential equation (SDE). Our algorithm is not susceptible to thermal noise, and in fact relies on noise as a resource, an advantage over existing analog methods. Moreover, the stochastic dynamics of the thermodynamic device can be put into an exact correspondence with the deterministic evolution that an ensemble of d copies of the device would undergo in the absence of noise. We respectively refer to these properties of our algorithm as *noise resilience* and *thermodynamic parallelism*, and they result in significant theoretical advantages over existing methods, both analog and digital.

Our algorithm provides a polynomial speedup over known methods, which is due to thermodynamic parallelism. While the time complexity of $O(d^2)$ can also be obtained using a parallel digital or analog method (as discussed earlier), this requires physical parallelism, meaning the hardware must be physically duplicated in order to solve the d initial value problems simultaneously. Our approach allows for the computation of a matrix exponential using a single device which would otherwise be capable of solving only one IVP at a time, and the total time required scales as $O(d^2)$. As a result, a matrix exponential can be computed with similar hardware and time requirements as would be necessary to solve a single IVP using a deterministic analog device, as shown in Fig. 1(c). To summarize, digital and analog methods allow for a time-space tradeoff, achieving either $O(d^3)$ time with a single thread or $O(d^2)$ time with $O(d)$ threads. Our thermodynamic matrix exponential algorithm runs in $O(d^2)$ time with a single device encoding d^2 quantities.

In what follows we give a basic description of the hardware necessary for this algorithm and the steps involved in its execution. We also present numerical simulation results, confirming that the time required scales no worse than $O(d^2)$. An analytical derivation of the time required for evaluating the matrix exponential can be found in the Supplemental Material [26].

II. THERMODYNAMIC MATRIX EXPONENTIALS

Let $A \in \mathbb{R}^{d\times d}$ be a real symmetric positive definite matrix (note that we will relax this assumption later in the paper). Suppose we would like to find the matrix exponential e^{-A}. We assume that the thermodynamic device evolves a system dictated by the overdamped Langevin equation

$$\mathrm{d}x = -Ax\,\mathrm{d}t + \mathcal{N}[0, 2\beta^{-1}B\,\mathrm{d}t], \tag{2}$$

for scalar $\beta \in \mathbb{R}^+$ and symmetric positive definite B. These dynamics represent an Ornstein-Uhlenbeck (OU) process with correlation function (or correlation matrix)

$$C(t,s) = \langle [x(t) - \langle x(t)\rangle][x(s) - \langle x(s)\rangle]^\intercal \rangle\,. \tag{3}$$

We will assume that $\langle x(0)\rangle = 0$, which implies that $\langle x(t)\rangle = 0$ at all times. The correlation matrix becomes

$$C(t,s) = \langle x(t)x^\intercal(s)\rangle\,.$$

At equilibrium, $x \sim \mathcal{N}[0, \Sigma]$ is normally distributed, with a correlation function given by [37]

$$C(t+\tau, t) = \langle x(t+\tau)x(t)^\intercal\rangle = e^{-A\tau}\Sigma.$$

The covariance matrix Σ satisfies the Lyapunov equation

$$A\Sigma + \Sigma A^\intercal = 2\beta^{-1}B. \tag{4}$$

Setting $B = A$ results in the solution $\Sigma = \beta^{-1}\mathbb{I}$. Therefore the SDE

$$\mathrm{d}x = -Ax\,\mathrm{d}t + \mathcal{N}[0, 2\beta^{-1}A\,\mathrm{d}t] \tag{5}$$

has a correlation function

$$C(t+\tau, t) = \beta^{-1}e^{-A\tau}. \tag{6}$$

Importantly, according to Eq. (6) the correlation function evolves in the same way as would a collection of d deterministic analog devices, which underlies the parallelism of the thermodynamic algorithm.

However, the correlation function is not directly accessible by measurement, and must be estimated via an average. Due to the ergodicity of this system [30,37], we can use a time average instead of an ensemble average, and obtain

$$e^{-A\tau} \approx \frac{\beta}{T}\int_0^T \mathrm{d}t\, x(t+\tau)x^\intercal(t), \tag{7}$$

which can be accumulated in continuous-time on an analog device (see Ref. [30]).

The parameters β and τ are tuning parameters and the matrix exponential e^{-A} can be extracted by rescaling the input A by τ^{-1}.

III. ARBITRARY MATRICES

We can relax the constraint that A needs to be symmetric positive definite. Suppose we want e^{-M} for arbitrary M. Then consider $A = c\mathbb{I} + M$ and $B = \frac{1}{2}(A + A^{\intercal})$ [which similarly results in $\Sigma = \beta^{-1}\mathbb{I}$ solving the Lyapunov equation in (4)]. The parameter $c \in \mathbb{R}^{+}$ must be large enough that the eigenvalues of A have a positive real part, which will also ensure that B is positive definite and a valid diffusion matrix. The SDE becomes

$$\mathrm{d}x = -Ax\,\mathrm{d}t + \mathcal{N}[0, \beta^{-1}(A + A^{\intercal})\mathrm{d}t]. \tag{8}$$

Applying the previous procedure will produce an approximation to

$$e^{-A} = e^{-c\mathbb{I}-M} = e^{-c}e^{-M}.$$

Therefore a scalar rescaling by e^{c} provides the desired e^{-M}. These arguments show that our thermodynamic algorithm can be used to obtain the exponential of any matrix in principle. In practice, one would have to be careful with precision if c is required to be too large. We find that the time required to evaluate the matrix exponential depends on the condition number, and the condition number of A differs from that of M. Also, if c is large then there will be a large loss in precision because the elements of e^{-A} will be very small, so it may be necessary to rescale M as well. A task for future work is to carefully take these considerations into account to describe how to select c in practice and the subsequent complexity of the matrix exponential algorithm in the case of arbitrary matrices.

IV. THERMODYNAMIC PARALLELISM

Let us now reflect on the mechanism of speedup. The advantage of the thermodynamic algorithm is that it allows for the collection of a sample of the matrix exponential in constant time (that is, time not scaling with dimension) using a device with d nodes, and d^2 couplings between them. Using a deterministic analog algorithm, the same device could be used to solve the d initial value problems in serial to obtain a single sample of the matrix exponential, resulting in an $O(d^3)$ time cost to collect the $O(d^2)$ samples.

The ability to solve d initial value problems simultaneously, using a device that ostensibly only solves one, can be attributed to the probabilistic nature of the thermodynamic algorithm. As discussed above, the correlation function is given by $C(t+\tau, t) = \beta^{-1}e^{-A\tau}$. An analogy can be drawn to an ensemble of d separate deterministic analog devices, each of which solves the differential equation $\mathrm{d}x = -Ax\,\mathrm{d}t$. We describe the state of this system by a matrix X, each of whose columns is the state of a single device in the ensemble. Initially we set the state of the ith device to the standard basis vector $\hat{e}_i$, so $X(0) = \mathbb{I}$. It is then apparent that $X(\tau) = e^{-A\tau}$, which is (up to a constant) the same as the time evolution of the correlation matrix of the thermodynamic device. While the state of the system $x(t)$ is a vector with only d components, the correlation matrix (which is a property of the *distribution*, rather than the state) has d^2 elements, representing the desired matrix exponential.

Thermodynamic parallelism stems from the fact that the probability distribution of the system is more informative than the instantaneous state; this aligns with the intuition of the system exploring multiple trajectories at the same time, which is why we refer to this advantage as a form of parallelism. Without noise, the above reasoning would break down. The dynamics of the differential equation $\mathrm{d}x = -Ax\,\mathrm{d}t$ are contractive and the Shannon entropy of the system tends to zero over time, meaning the correlations will eventually be too small to measure. Thus, noise can be viewed as the key resource that enables thermodynamic parallelism.

While we focus here on the matrix exponential algorithm, similar arguments based on thermodynamic parallelism can be used to explain the asymptotic advantage achieved by the thermodynamic matrix inversion algorithm in Ref. [30]. In that work, the matrix inverse is found by estimating the covariance matrix of the thermodynamic device, $\Sigma = \langle xx^{\intercal}\rangle$. For a symmetric positive definite matrix A, the SDE $\mathrm{d}x = -Ax + \mathcal{N}[0, 2\beta^{-1}\mathbb{I}\,\mathrm{d}t]$ eventually reaches a stationary distribution $x \sim \mathcal{N}[0, \beta^{-1}A^{-1}]$, so $\beta\Sigma = A^{-1}$ at equilibrium [30]. The matrix inverse could also be found by solving d linear systems of equations of the form $Ax = \hat{e}_i$, where $\hat{e}_i$ is the ith standard basis vector. An analog device could solve a linear system of this form by evolving the ordinary differential equation (ODE) $\mathrm{d}x = -(Ax - \hat{e}_i)\,\mathrm{d}t$ for sufficiently long. A collection of d such devices could solve the d linear systems in parallel via the matrix ODE

$$\mathrm{d}X = -AX\,\mathrm{d}t + \mathbb{I}\,\mathrm{d}t. \tag{9}$$

In the thermodynamic matrix inverse algorithm, the covariance matrix evolves under the ODE

$$d\Sigma = -A\Sigma\,\mathrm{d}t - \Sigma A\,\mathrm{d}t + 2\mathbb{I}\,\mathrm{d}t, \tag{10}$$

where we set $\beta = 1$. This is a symmetrized version of Eq. (9), and A and Σ are symmetric, Equations (9) and (10) have the same stationary solution $X = \Sigma = A^{-1}$. This means that the thermodynamic matrix inverse algorithm allows a single device similar to one in Fig. 2 to emulate a collection of d deterministic analog devices. Moreover, this suggests that thermodynamic parallelism, arising from stochastic noise, may be a broader mechanism to explain the potential advantage of thermodynamic computers.

V. PHYSICAL DEVICE CONSTRUCTION

Let us discuss how one could construct a device to implement the thermodynamic matrix exponential algorithm. Essentially, we need a device that can implement the SDE in Eq. (5). Implementing this OU process with electrical circuits has, to some extent, been discussed in Refs. [29,30] where stochastic units (s-units) were proposed as the basic building block of the hardware, with each s-unit composed of an *RC* circuit with a stochastic voltage source. Indeed, an array of s-units, capacitively coupled to one another, is shown in the

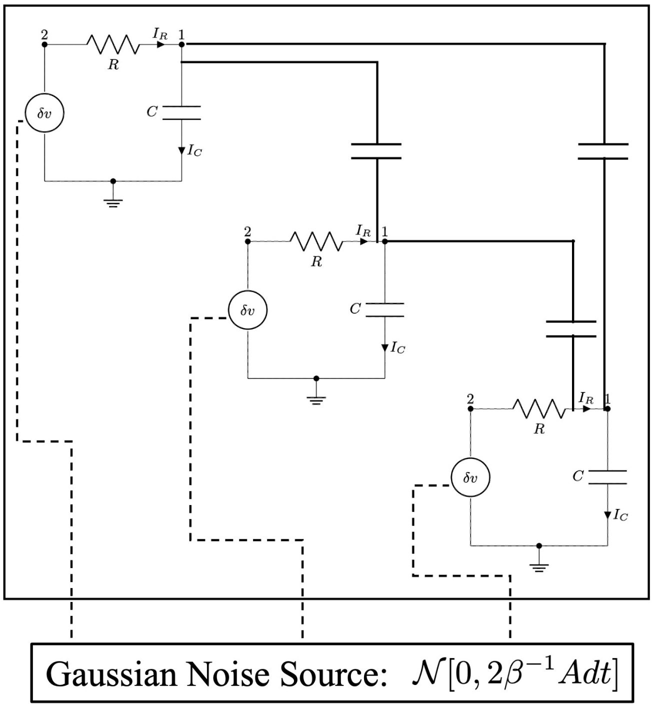

FIG. 2. Possible hardware architecture for matrix exponentiation. The overall system is composed of two subunits. One subunit (bottom) outputs, as a vector of voltages, Gaussian noise with zero mean and covariance matrix proportional to A. The other subunit (top) takes this output as its noise source and directly simulates the stochastic differential equation in (5).

top of Fig. 2, and in theory this s-unit array can simulate Eq. (5).

However, an additional subtlety about Eq. (5) is that the noise source is correlated. Hence, one needs a Gaussian noise source that outputs the vector of correlated noise values to provide the noise term in Eq. (5). This is shown simply as a black box at the bottom of Fig. 2, while we give a detailed discussion on how to construct this noise source in the Supplemental Material [26]. Namely, this noise source can consist of an (additional) array of s-units that are capacitively coupled, and the voltage vector outputted by the noise source is the output of integrators that integrate the current flow through the resistors in the s-units (see Supplemental Material [26] for details).

VI. COMPLEXITY

As outlined earlier, the thermodynamic matrix exponential algorithm has two steps: (1) Sample from the stationary distribution $x(0) \sim \mathcal{N}[0, \beta^{-1}\mathbb{I}]$ (digitally or otherwise). (2) Evaluate Eq. (7) by integrating to time T determined using Eq. (11) below.

We denote the time average appearing in Eq. (7) by $\overline{C}$, and define the root mean square (RMS) error $\mathcal{E} = \sqrt{\langle \|e^{-A\tau} - \beta\overline{C}\|_F^2\rangle}$, where $\|\cdot\|_F$ is the Frobenius norm. As shown in the Supplemental Material [26], we can achieve an RMS error of $\mathcal{E}$ or lower by setting the integration time larger than the following lower bound,

$$T \geqslant \frac{2d(d+1)e^{2\alpha_{\min}\tau}\kappa}{\|A\|\mathcal{E}^2}, \tag{11}$$

where κ and $\alpha_{\min}$ are respectively the condition number and the smallest singular value of A. Therefore the time complexity scaling is $O(d^2\kappa\mathcal{E}^{-2})$ which provides a speedup for well-conditioned matrices, although we note that the condition number is not guaranteed to remain constant and may well increase with dimension. However, many practical scenarios involve matrices where the condition number scales favorably, allowing the algorithm to retain a computational advantage.

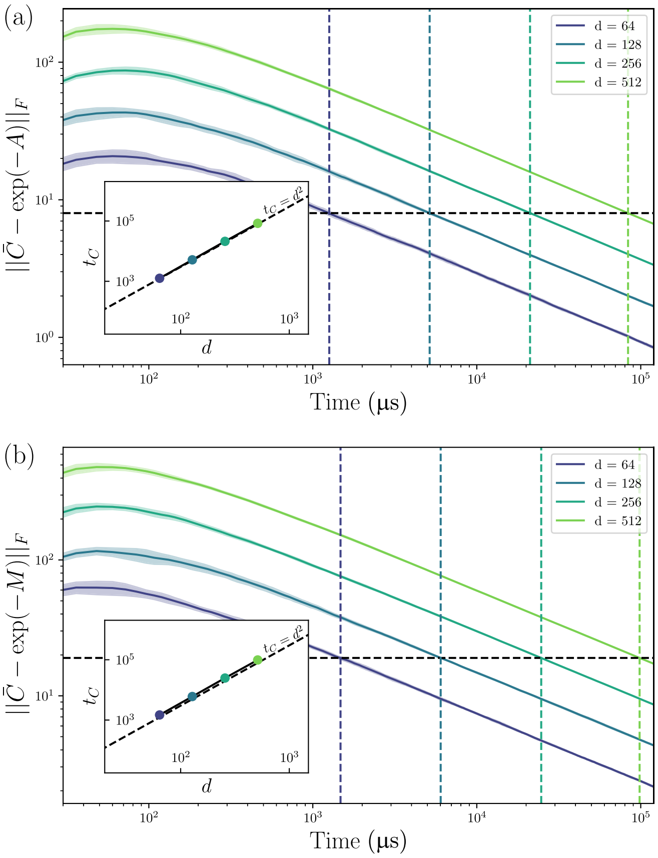

FIG. 3. Error as a function of analog integration time for varying matrix dimension. The error is the Frobenius norm of the sampled matrix minus the true matrix exponential. (a) Matrices are positive definite and drawn from a Wishart distribution (with $2d$ degrees of freedom). (b) Matrices are asymmetric, with elements drawn from the Haar distribution over orthogonal matrices. To make the eigenvalues have positive real part, a term $c\mathbb{I}$ is added as described in the text, with $c = 1.1$. Displayed are average and one standard deviation over ten random seeds. Vertical lines represent the times t_C when the error falls below a threshold. Inset: Crossing time t_C vs matrix dimension d.

VII. NUMERICS

Figure 3 shows the numerical simulation results for our thermodynamic matrix exponentiation algorithm. The Frobenius error is plotted versus time for various dimensions, with the insets showing the time to reach a given error threshold versus dimension. For both positive-definite and more general matrices, respectively shown in Figs. 3(a) and 3(b), the runtime does not grow faster than $O(d^2)$, as expected (as we have τ and $\alpha_{\min} = \frac{\kappa}{\|A\|}$ constant). This result holds for matrices drawn from a Wishart distribution and from an orthogonal Haar distribution, which provides evidence that our bounds on polynomial scaling are widely applicable. Reproducible code for all simulations can be found in the open-source thermodynamic simulator THERMOX [38].

VIII. DISCUSSION

We have shown that matrix exponentiation can be solved in an amount of time proportional to the square of the dimension, whereas previously the best known upper estimates of asymptotic time complexity had higher-order polynomial scaling with dimension. However, in this work we have not investigated the scaling of the energy cost of our algorithm, and describing this energy scaling is a crucially important task. In other work, some evidence has been found of energetic advantage for similar thermodynamic algorithms [30,39].

The promise of quantum computing has often been attributed to the (still controversial) idea of quantum parallelism [40] and quantum matrix exponentiation has been proposed [41], but the practicality and commercial impact of quantum computing remains long term in timescale. We have found that a computationally useful form of parallelism can be achieved within a classical probabilistic setting with near-term hardware, and its origin is explained unambiguously. It should be noted that, while the presence of this parallelism may yield insight into the theoretical speedup we have identified, the complexity of our algorithm can be analyzed without involving the idea of parallelism. Moreover, at present it is not clear how the type of parallelism we have identified compares with the more conventional forms of parallelism commonly used in digital computing, for example, parallelized arithmetic on a GPU. Elucidating this relationship is an area of ongoing research.

Matrix exponentiation is relevant to virtually all time-dependent processes that have linear feedback. These appear, of course, in physics, but also in disparate fields such as machine learning, biology, and economics. The future impact of these results, then, is limited only by the scale of thermodynamic hardware that can be built. These results also reveal a potential polynomial separation between the digital computing and thermodynamic computing paradigms, in terms of time complexity.

Demonstrating the utility of thermodynamic computation over standard digital hardware can only be convincingly proven through experimentation, so a physical demonstration is a key future direction. We anticipate that, as a result of the potential advantages we have found, thermodynamic computing will become a rich and competitive ground for both theoretical and experimental work in the near future.

[1] A. V. Oppenheim, *Discrete-Time Signal Processing* (Pearson Education India, Chennai, 1999).

[2] J. D. Hamilton, *Time Series Analysis* (Princeton University Press, Princeton, NJ, 1994).

[3] C. Lu, Y. Schroecker, A. Gu, E. Parisotto, J. Foerster, S. Singh, and F. Behbahani, Structured state space models for in-context reinforcement learning, Adv. Neural Info. Proc. Systems, **36** (2024).

[4] A. Gu, K. Goel, and C. Ré, Efficiently Modeling Long Sequences with Structured State Spaces, in *Proceedings of International Conference on Learning Representations* (ICLR, 2022).

[5] J. F. Shortle, J. M. Thompson, D. Gross, and C. M. Harris, *Fundamentals of Queueing Theory*, Wiley Series in Probability and Statistics Vol. 399 (Wiley, Hobokem, NJ, 2018).

[6] L. J. S. Allen, *An Introduction to Stochastic Processes with Applications to Biology* (CRC Press, Boca Raton, FL, 2010).

[7] S. Duffield, S. Power, and L. Rimella, A state-space perspective on modelling and inference for online skill rating, J. R. Stat. Soc. Ser. C: Appl. Stat. **73**, 1262 (2024).

[8] C. Moler and C. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, SIAM Rev. **20**, 801 (1978).

[9] C. Moler and C. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, SIAM Rev. **45**, 3 (2003).

[10] J. A. Acebron, A Monte Carlo method for computing the action of a matrix exponential on a vector, Appl. Math. Comput. **362**, 124545 (2019).

[11] J. A. Barker, A quantum-statistical Monte Carlo method; path integrals with boundary conditions, J. Chem. Phys. **70**, 2914 (1979).

[12] Z. Bai, W. Chen, R. Scalettar, and I. Yamazaki, Numerical methods for quantum Monte Carlo simulations of the Hubbard model, in *Multi-Scale Phenomena in Complex Fluids: Modeling, Analysis and Numerical Simulation* (World Scientific, Singapore, 2009), pp. 1–110.

[13] H. Padé, Sur la représentation approchée d'une fonction par des fractions rationnelles, Ann. Sci. Éc. Norm. Supér. **9**, 3 (1892).

[14] A. H. Al-Mohy and N. J. Higham, A new scaling and squaring algorithm for the matrix exponential, SIAM J. Matrix Anal. Appl. **31**, 970 (2010).

[15] V. V. Williams, Y. Xu, Z. Xu, and R. Zhou, New bounds for matrix multiplication: from alpha to omega, in *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (SIAM, 2024), pp. 3792–3835.

[16] PNY Technologies, NVIDIA A100 - PNY Technologies, https://www.pny.com/nvidia-a100, 2023, accessed 20 November 2023.

[17] R. Chowdhury, F. Silvestri, and F. Vella, A computational model for tensor core units, in *Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures* (Association for Computing Machinery, New York, 2020), pp. 519–521.

[18] T. D. Ahle and F. Silvestri, Similarity search with tensor core units, in *International Conference on Similarity Search and Applications* (Springer, Berlin, 2020), pp. 76–84.

[19] N. Mohseni, P. L. McMahon, and T. Byrnes, Ising machines as hardware solvers of combinatorial optimization problems, Nat. Rev. Phys. **4**, 363 (2022).

[20] J. Kaiser, S. Datta, and B. Behin-Aein, Life is probabilistic—why should all our computers be deterministic? Computing with p-bits: Ising solvers and beyond, in *2022 International Electron Devices Meeting (IEDM)* (IEEE, Piscataway, NJ, 2022), pp. 21–24.

[21] J. Chou, S. Bramhavar, S. Ghosh, and W. Herzog, Analog coupled oscillator based weighted ising machine, Sci. Rep. **9**, 14786 (2019).

[22] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud *et al.*, Neuromorphic silicon neuron circuits, Front. Neurosci. **5**, 73 (2011) .

[23] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves, Z. Li, J. P. Strachan, P. Lin, Z. Wang, M. Barnell, Q. Wu, R. S. Williams, and J. J. Yang, Analogue signal and image processing with large memristor crossbars, Nat. Electron. **1**, 52 (2018).

[24] S.-i. Yi, J. D. Kendall, R. S. Williams, and S. Kumar, Activity-difference training of deep neural networks using memristor crossbars, Nat. Electron. **6**, 45 (2023).

[25] P. E. Allen, R. Dobkin, and D. R. Holberg, *CMOS Analog Circuit Design* (Elsevier, Amsterdam, 2011).

[26] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevResearch.7.013147 for further details on thermodynamic matrix exponential convergence, complexity, noise resilience, error mitigation and thermodynamic parallelism.

[27] N. J. Higham, *Functions of Matrices: Theory and Computation* (SIAM, Philadelphia, 2008).

[28] T. Conte, E. DeBenedictis, N. Ganesh, T. Hylton, J. P. Strachan, R. S. Williams, A. Alemi, L. Altenberg, G. E. Crooks, J. Crutchfield *et al.*, Thermodynamic computing, arXiv:1911.01968.

[29] P. J. Coles, C. Szczepanski, D. Melanson, K. Donatella, A. J. Martinez, and F. Sbahi, Thermodynamic AI and the fluctuation frontier, arXiv:2302.06584.

[30] M. Aifer, K. Donatella, M. H. Gordon, S. Duffield, T. Ahle, D. Simpson, G. E. Crooks, and P. J. Coles, Thermodynamic linear algebra, npj Unconv. Comput. **1**, 13 (2024).

[31] S. Chowdhury, A. Grimaldi, N. A. Aadit, S. Niazi, M. Mohseni, S. Kanai, H. Ohno, S. Fukami, L. Theogarajan, G. Finocchio *et al.*, A full-stack view of probabilistic computing with p-bits: devices, architectures and algorithms, IEEE J. Explor. Solid-State Comput. Devices **9**, 1 (2023).

[32] T. Hylton, Thermodynamic neural network, Entropy **22**, 256 (2020).

[33] N. Ganesh, A thermodynamic treatment of intelligent systems, in *2017 IEEE International Conference on Rebooting Computing (ICRC)* (IEEE, Piscataway, NJ, 2017), pp. 1–4.

[34] A. A. Alemi and I. Fischer, TherML: Thermodynamics of machine learning, arXiv:1807.04162.

[35] A. B. Boyd, J. P. Crutchfield, and M. Gu, Thermodynamic machine learning through maximum work production, New J. Phys. **24**, 083040 (2022).

[36] P. Lipka-Bartosik, M. Perarnau-Llobet, and N. Brunner, Thermodynamic computing via autonomous quantum thermal machines, Sci. Adv. **10**, eadm8792 (2024).

[37] C. W. Gardiner, *Handbook of Stochastic Methods* (Springer, Berlin, 1985), Vol. 3.

[38] S. Duffield, K. Donatella, and D. Melanson, THERMOX: Exact OU processes with JAX (2024), https://github.com/normal-computing/thermox.

[39] M. Aifer, S. Duffield, K. Donatella, D. Melanson, P. Klett, Z. Belateche, G. Crooks, A. J. Martinez, and P. J. Coles, Thermodynamic Bayesian inference, arXiv:2410.01793.

[40] R. Jozsa, Characterizing classes of functions computable by quantum parallelism, Proc. R. Soc. London, Ser. A **435**, 563 (1991).

[41] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nat. Phys. **10**, 631 (2014).