

<https://doi.org/10.1038/s44335-024-00014-0>

Thermodynamic linear algebra

Check for updates

Maxwell Aifer[✉], Kaelan Donatella, Max Hunter Gordon, Samuel Duffield, Thomas Ahle, Daniel Simpson, Gavin Crooks & Patrick J. Coles

Linear algebra is central to many algorithms in engineering, science, and machine learning; hence, accelerating it would have tremendous economic impact. Quantum computing has been proposed for this purpose, although the resource requirements are far beyond current technological capabilities. We consider an alternative physics-based computing paradigm based on classical thermodynamics, to provide a near-term approach to accelerating linear algebra. At first sight, thermodynamics and linear algebra seem to be unrelated fields. Here, we connect solving linear algebra problems to sampling from the thermodynamic equilibrium distribution of a system of coupled harmonic oscillators. We present simple thermodynamic algorithms for solving linear systems of equations, computing matrix inverses, and computing matrix determinants. Under reasonable assumptions, we rigorously establish asymptotic speedups for our algorithms, relative to digital methods, that scale linearly in matrix dimension. Our algorithms exploit thermodynamic principles like ergodicity, entropy, and equilibration, highlighting the deep connection between these two seemingly distinct fields, and opening up algebraic applications for thermodynamic computers.

Basic linear algebra primitives like solving linear systems and inverting matrices are present in many modern algorithms. Such primitives are relevant to a multitude of applications, for example optimal control of dynamic systems and resource allocation. They are also a common sub-routine of many artificial intelligence (AI) algorithms, and account for a substantial portion of the time and energy costs in some cases. The most common method to perform these primitives is LU decomposition, whose time-complexity scales as $O(d^3)$. Many proposals have been made to accelerate such primitives, for example using iterative methods such as the conjugate gradient method. In the last decade, these primitives have been accelerated by hardware improvements, notably by graphical processing units (GPUs), fueling massive parallelization. However, the scaling of these methods is still a prohibitive factor, and obtaining a good approximate solution to a dense matrix of more than a few tens of thousand dimensions remains challenging.

Exploiting physics to solve mathematical problems is a deep idea, with much focus on solving optimization problems^{1–3}. In the context of linear algebra, much attention has been paid to quantum computers⁴, since the mathematics of discrete-variable quantum mechanics matches that of linear algebra. A quantum algorithm⁵ to solve linear systems has been proposed, which for sparse and well-conditioned matrices scales as $\log d$. However, the resource requirements⁶ for this algorithm are far beyond current hardware capabilities. More generally building large-scale quantum hardware has remained difficult⁷, and variational quantum algorithms for linear algebra^{8–10} have battled with vanishing gradient issues^{11–13}.

Therefore, the search for alternative hardware proposals that can exploit physical dynamics to accelerate linear algebra primitives has been

ongoing. Notably, memristor crossbar arrays have been of interest for accelerating matrix-vector multiplications^{14,15}. Solving linear systems has also been the subject of analog computing approaches¹⁶.

Recently, we defined a new class of hardware, built from stochastic, analog building blocks, which is ultimately thermodynamic in nature¹⁷. (See also probabilistic-bit computers^{18–20} and thermodynamic neural networks^{21–25} for alternative approaches to thermodynamic computing²⁶). AI applications like generative modeling are a natural fit for this thermodynamic hardware, where stochastic fluctuations are exploited to generate novel samples.

In this work, we surprisingly show that the same thermodynamic hardware from Ref. 17 can also be used to accelerate key primitives in linear algebra. Thermodynamics is not typically associated with linear algebra, and connecting these two fields is therefore non-trivial. Here, we exploit the fact that the mathematics of harmonic oscillator systems is inherently affine (i.e., linear), and hence we can map linear algebraic primitives onto such systems. (See also Ref. 27 for a discussion of harmonic oscillators in the context of quantum computing speedups.) We show that simply by sampling from the thermal equilibrium distribution of coupled harmonic oscillators, one can solve a variety of linear algebra problems.

Specifically we develop thermodynamic algorithms for the following linear algebraic primitives: (i) solving a linear system $Ax = b$, (ii) estimating a matrix inverse A^{-1} , (iii) solving Lyapunov equations²⁸ of the form $A\Sigma + \Sigma A^T = \mathbb{1}$ and (iv) estimating the determinant of a symmetric positive definite matrix A . We show that if implemented on thermodynamic hardware, these methods scale favorably with problem size compared to digital algorithms. Our numerical simulations corroborate our analytical

scaling results and also provide evidence of the fast convergence of these primitives with the wall-clock time, with the speedup relative to digital methods getting more pronounced with increasing dimension and condition number.

We remark that there is a connection between our thermodynamic algorithms and digital Monte-Carlo (MC) algorithms that were developed for linear algebra^{29–33}. Namely, our algorithms can be viewed as a continuous-time version of these digital MC algorithms. However, we emphasize that the continuous time (i.e., physics-based rather than physics-inspired) nature of our algorithms is crucial for obtaining our predicted asymptotic speedup. Additionally, thermodynamic algorithms can be run on a single device³⁴ whereas efficient digital MC linear algebra requires extensive parallelization³⁰.

Results

Algorithmic scaling

In Table 1, we summarize the asymptotic scaling results for our thermodynamic algorithms as compared to the best state-of-the-art (SOTA) digital methods for dense symmetric positive-definite matrices. The derivations of these results can be found in the Supplementary Information (Supplementary Notes 1–5), and are based on bounds obtained for physical thermodynamic quantities, including correlation times, equilibration times, and free energy differences. As one can see from Table 1, an asymptotic speedup is predicted for our thermodynamic algorithms relative to the digital SOTA algorithms. Specifically, a speedup that is linear in d is expected for each of the linear algebraic primitives (ignoring a possible dependence of κ on d). We remark that the complexity of analog algorithms is subtle³⁵ and depends, e.g., on assumptions of how the hardware size grows with problem size. The assumptions made to obtain our scaling results are detailed in the Methods section. In what follows, we systematically present our thermodynamic algorithms for various linear algebraic primitives.

Table 1 | Comparison of asymptotic complexities of linear algebra algorithms

Problem	Digital SOTA	Thermodynamic
Linear system	$O(\min\{d^\omega, d^2\sqrt{\kappa}\})$	$O(d\kappa^2\varepsilon^{-2})$
Matrix inverse	$O(d^3)$	$O(d^2\kappa\varepsilon^{-2})$
Lyapunov equation	$O(d^6)$	$O(d^2\kappa\varepsilon^{-2})$
Matrix determinant	$O(d^6)$	$O(d\kappa \ln(\kappa)^3\varepsilon^{-2})$

Here, d is the matrix dimension, κ is the condition number, and ε is the error. For our thermodynamic algorithms, the complexity depends on the dynamical regime. Here we display the overdamped dynamics which have marginally better complexity than the underdamped equivalents. For the digital SOTA, the complexity of solving symmetric, positive definite linear systems, matrix inverse, Lyapunov equation, and matrix determinant problems are respectively for algorithms based on: conjugate gradient method⁶³, fast matrix multiplication/inverse⁶⁴, Bartels-Stewart algorithm⁶⁵, and LUP decomposition⁶⁶. $\omega \approx 2.3$ denotes the matrix multiplication constant.

Solving linear systems of equations

The celebrated linear systems problem is to find $x \in \mathbb{R}^d$ such that

$$Ax = b, \tag{1}$$

given some invertible matrix $A \in \mathbb{R}^{d \times d}$ and nonzero $b \in \mathbb{R}^d$. We may assume without loss of generality that the matrix A in Eq. (1) is symmetric and positive definite (SPD); if A is not SPD, then we may consider the system $A^T Ax = A^T b$, whose solution $x = A^{-1}b$ is also the solution of $Ax = b$. This will affect the total runtime, but still allows for asymptotic scaling improvements with respect to digital methods, in some cases. Also note that constructing an SPD system from a generic one in this way results in the squaring of the condition number, which influences performance. In what follows, we will therefore assume that A is SPD. Now let us connect this problem to thermodynamics. We consider a macroscopic device with d degrees of freedom, described by classical physics. Suppose the device has potential energy function:

$$U(x) = \frac{1}{2}x^T Ax - b^T x, \tag{2}$$

where $A \in \text{SPD}_d(\mathbb{R})$. Note that this is a quadratic potential that can be physically realized with a system of harmonic oscillators, where the coupling between the oscillators is determined by the matrix A , and the b vector describes a constant force on each individual oscillator. (We remark that while Fig. 1 depicts mechanical oscillators, from a practical perspective, one can build the device from electrical oscillators such as RLC circuits.)

Suppose that we allow this device to come to thermal equilibrium with its environment, whose inverse temperature is $\beta = 1/k_B T$. At thermal equilibrium, the Boltzmann distribution describes the probability for the oscillators to have a given spatial coordinate: $f(x) \propto \exp(-\beta U(x))$. Because $U(x)$ is a quadratic form, $f(x)$ corresponds to a multivariate Gaussian distribution. Thus at thermal equilibrium, the spatial coordinate x is a Gaussian random variable

$$x \sim \mathcal{N}[A^{-1}b, \beta^{-1}A^{-1}]. \tag{3}$$

The key observation is that the unique minimum of $U(x)$ occurs where $Ax - b = 0$, which also corresponds to the unique maximum of $f(x)$. For a Gaussian distribution, the maximum of $f(x)$ is also the first moment $\langle x \rangle$. Thus, we have that, at thermal equilibrium, the first moment is the solution to the linear system of equations:

$$\langle x \rangle = A^{-1}b. \tag{4}$$

From this analysis, we can construct a simple thermodynamic protocol for solving linear systems, which is depicted in Fig. 1. Namely, the protocol

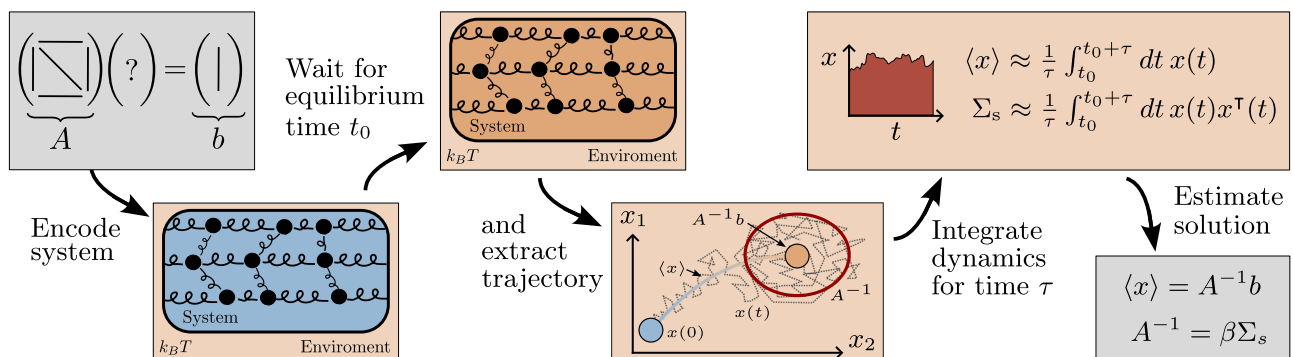


Fig. 1 | Diagram of our thermodynamic algorithm for solving linear systems and inverse estimation. The system of linear equations, or the matrix A , is encoded into the thermodynamic hardware, the system is then allowed to evolve until the

stationary distribution has been reached, when the trajectory is then integrated to estimate the sample mean or covariance. This gives estimates of the solution of the linear system or the inverse of A respectively.

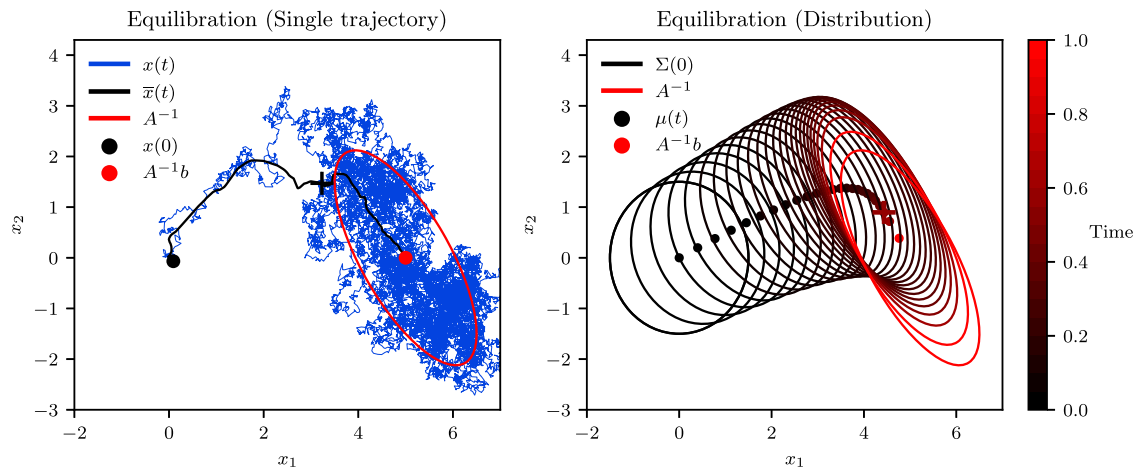


Fig. 2 | Equilibration of the thermodynamic system. The process of equilibration is depicted on the single-trajectory level (left) and on the distribution level (right). The trajectory dynamics are described by the overdamped Langevin equation and the distributional dynamics by the Fokker-Planck equation⁶⁷. The system displays

ergodicity, as the time average of a single trajectory (blue curve, left) approaches the ensemble average (dots, right) in the long-time limit. Time and the coordinate vector (x_1, x_2) are in arbitrary units.

involves realizing the potential in Eq. (2), waiting for the system to come to equilibrium, and then sampling x to estimate the mean $\langle x \rangle$ of the distribution. This mean can be approximated using a time-average, defined as

$$\langle x \rangle \approx \bar{x}(\tau) = \frac{1}{\tau} \int_{t_0}^{t_0+\tau} dt' x(t'), \quad (5)$$

where t_0 must be sufficiently large to allow for equilibration and τ must be sufficiently large for the average to converge to a desired degree of precision. The eventual convergence of this time average to the mean is the content of the ergodic hypothesis^{36,37}, which is often assumed for quite generic thermodynamic systems. It should be mentioned that the mean could also be approximated as the average of a sequence of samples; however the integration approach has the advantage that it can conveniently be implemented in a completely analog way (for example, using an integrator electrical circuit), which obviates the need for transferring data from the physical device until the end of the protocol.

Figure 2 shows the equilibration process for both a single trajectory (left panel) and the overall distribution (right panel). One can see the ergodic principle illustrated in this figure, since the time dynamics of a single trajectory at thermal equilibrium are representative of the overall ensemble.

The overall protocol can be summarized as follows.

Linear system protocol.

- Given a linear system $Ax = b$, set the potential of the device to

$$U(x) = \frac{1}{2} x^T A x - b^T x \quad (6)$$

at time $t = 0$.

- Choose equilibration tolerance parameters $\epsilon_{\mu 0}, \epsilon_{\Sigma 0} \in \mathbb{R}^+$, and choose the equilibration time

$$t_0 \geq \hat{t}_0, \quad (7)$$

where \hat{t}_0 is computed from the system's physical properties or using heuristic methods based on Eqs. (28), (30). Allow the system to evolve under its dynamics until $t = t_0$, which ensures that $\| \langle x \rangle - A^{-1} b \| / \| A^{-1} b \| \leq \epsilon_{\mu 0}$ and $\| \Sigma - \beta^{-1} A^{-1} \| / \| \beta^{-1} A^{-1} \| \leq \epsilon_{\Sigma 0}$.

- Choose error tolerance parameter ϵ_x and success probability P_s , and choose the integration time

$$\tau \geq \hat{\tau}, \quad (8)$$

where $\hat{\tau}$ is computed from the system's physical properties, Eq. (28) or (30). Use an analog integrator to measure the time average

$$\bar{x} = \frac{1}{\tau} \int_{t_0}^{t_0+\tau} dt x(t), \quad (9)$$

which satisfies $\| A \bar{x} - b \| / \| b \| \leq \epsilon_x$ with probability at least P_s .

In order to implement the protocol above, the necessary values of \hat{t}_0 and $\hat{\tau}$ must be identified, which requires a more quantitative description of equilibration and ergodicity. To obtain such a description, a model of the system's microscopic dynamics may be introduced. Given that the system under consideration is composed of harmonic oscillators in contact with a heat bath, it is natural to allow for damping (i.e., energy loss to the bath) and stochastic thermal noise, which always accompanies damping due to the fluctuation-dissipation theorem^{38,39}. The Langevin equation accounts for these effects, and specifically we consider two common formulations, the overdamped Langevin (ODL) equation and the underdamped Langevin (UDL) equations. In the Methods section, we provide additional details on ODL and UDL dynamics, and we provide explicit formulas for \hat{t}_0 and $\hat{\tau}$ for the overdamped and underdamped regimes.

Estimating the inverse of a matrix

The results of the previous section rely on estimating the mean of x , but make no use of the fluctuations in x at equilibrium. By using the second moments of the equilibrium distribution, we can go beyond solving linear systems. For example it is possible to find the inverse of an SPD matrix A . As mentioned, the stationary distribution of x is $\mathcal{N}[A^{-1} b, \beta^{-1} A^{-1}]$, meaning the inverse of A can be obtained by evaluating the covariance matrix of x . This can be accomplished in an entirely analog way, using a combination of analog multipliers and integrators. By setting $b = 0$ for this protocol, we ensure that $\langle x \rangle = 0$, so the stationary covariance matrix is, by definition

$$\Sigma_s = \lim_{t \rightarrow \infty} \langle x(t) x^T(t) \rangle. \quad (10)$$

In order to estimate this, we again perform time averages after allowing the system to come to equilibrium

$$\Sigma_s \approx \overline{xx^T} = \frac{1}{\tau} \int_{t_0}^{t_0+\tau} dt x(t) x^T(t). \quad (11)$$

It is therefore necessary to have an analog component which evaluates the product $x_i(t)x_j(t)$ for each pair (i, j) , resulting in d^2 analog multiplier

components. Each of these products is then fed into an analog integrator component, which computes one element of the time-averaged covariance matrix

$$\Sigma_{s,ij} \approx \frac{1}{\tau} \int_{t_0}^{t_0+\tau} dt x_i(t)x_j(t). \tag{12}$$

While the equilibration time is the same as for the linear system protocol, the integration time is different, because in general the covariance matrix is slower to converge than the mean. We now give a detailed description of the inverse estimation protocol, assuming ODL dynamics (the corresponding results for underdamped dynamics can be found in Supplementary Note 3). In the Methods section, we provide explicit formulas for \hat{t}_0 and $\hat{\tau}$ for the Inverse Estimation Protocol. We remark that our matrix inversion algorithm is a special case of our general algorithm for solving Lyapunov equations; the latter is presented in Supplementary Note 5.

Inverse estimation protocol.

- Given a positive definite matrix A , set the potential of the device to

$$U(x) = \frac{1}{2} x^T A x \tag{13}$$

at time $t = 0$.

- Choose equilibration tolerance parameter $\epsilon_{\Sigma 0} \in \mathbb{R}^+$, and choose the equilibration time

$$t_0 \geq \hat{t}_0, \tag{14}$$

where \hat{t}_0 is computed from the system’s physical properties, Eq. (33) or (34). Allow the system to evolve under its dynamics until $t = t_0$, which ensures that $\|\Sigma - \beta^{-1} A^{-1} b\| / \|\beta^{-1} A^{-1}\| \leq \epsilon_{\Sigma}$.

- Choose error tolerance parameter ϵ_{Σ} and success probability P_{ϵ} , and choose the integration time

$$\tau \geq \hat{\tau}, \tag{15}$$

where $\hat{\tau}$ is computed from the system’s physical properties, Eq. (33) or (34). Use analog multipliers and integrators to measure the time averages

$$\overline{x_i x_j} = \frac{1}{\tau^2} \int_{t_0}^{\tau} dt x_i(t)x_j(t), \tag{16}$$

which satisfies $\|\overline{xx^T} - \beta^{-1} A^{-1}\|_F / \|\beta^{-1} A^{-1}\|_F \leq \epsilon_A$ with probability at least P_{ϵ} .

Estimating the determinant of a matrix

The determinant of the covariance matrix appears in the normalization factor of a multivariate normal distribution, whose density function is

$$f_{\mu;\Sigma}(x) = (2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right), \tag{17}$$

and it is therefore natural to wonder whether hardware which is capable of preparing a Gaussian distribution may be used to somehow estimate the determinant of a matrix. This can in fact be done, as the problem is equivalent to the estimation of free energy differences, an important application of stochastic thermodynamics. Recall that the difference in free energy between equilibrium states of potentials U_1 and U_2 is⁴⁰

$$\Delta F = F_2 - F_1 = -\beta^{-1} \ln\left(\frac{\int dx e^{-\beta U_2(x)}}{\int dx e^{-\beta U_1(x)}}\right). \tag{18}$$

Suppose the potentials are quadratic, with $U_1(x) = x^T A_1 x$ and $U_2(x) = x^T A_2 x$. Then each integral simplifies to the inverse of a Gaussian normalization factor,

$$\int dx e^{-\beta V_j(x)} = (2\pi)^{d/2} \sqrt{\beta^{-1} |A_j^{-1}|}, \tag{19}$$

so

$$\Delta F = -\beta^{-1} \ln\left(\sqrt{\frac{|A_2^{-1}|}{|A_1^{-1}|}}\right) = -\beta^{-1} \ln\left(\sqrt{\frac{|A_1|}{|A_2|}}\right). \tag{20}$$

This suggests that the determinant of a matrix A_1 can be found by comparing the free energies of the equilibrium states with potentials U_1 and U_2 (where A_2 has known determinant), and then computing

$$|A_1| = e^{-2\beta\Delta F} |A_2|. \tag{21}$$

Fortunately, the free energy difference ΔF can be found, assuming we have the ability to measure the work which is done on the system as the potential $U(x)$ is changed from U_1 to U_2 . According to the Jarzynski equality⁴¹, the free energy difference between the (equilibrium) states in the initial and final potential is

$$e^{-\beta\Delta F} = \langle e^{-\beta W} \rangle, \tag{22}$$

where $\langle \cdot \rangle$ denotes an average over all possible trajectories of the system between time $t = 0$ and time $t = \tau$, weighed by their respective probabilities. This may be approximated by an average over N repeated trials,

$$e^{-\beta\Delta F} \approx \overline{e^{-\beta W}} \equiv \frac{1}{N} \sum_{j=1}^N e^{-\beta W_j}. \tag{23}$$

However, while Jarzynski’s relation may be applied directly to estimate the free energy difference, this estimator has large bias and is slow to converge. Far more well-behaved estimators have been found based on work measurements. For simplicity, we here provide the expression based on Jarzynski’s estimator, while in the Methods section and in Supplementary Note 4 we refer to more suitable estimators. In summary, the determinant of A_1 is approximated by

$$|A_1| \approx \left(\overline{e^{-\beta W}}\right)^2 |A_2|. \tag{24}$$

In practice we will generally be interested in the log determinant to avoid computational overflow. This is

$$\ln(|A_1|) \approx 2 \ln\left(\overline{e^{-\beta W}}\right) + \ln(|A_2|). \tag{25}$$

It is shown in Supplementary Note 4 that to estimate the log determinant to within (absolute) error δ_{LD} with probability at least P_{δ} , the total amount of time required is roughly

$$\tau \approx \frac{d \ln(\kappa)^2}{\delta_{LD}^2 (1 - P_{\delta})} \ln\left(\chi^2 \kappa^{3/2} \epsilon_{\Sigma 0}^{-1} \left[\frac{1}{4\zeta_{\max}^2} + 1\right]\right) \tau_{\text{r(UD)}} = O(d \ln(\kappa)^3). \tag{26}$$

We also present numerical simulations of a protocol for determinant estimation that does not include directly measuring the work in Supplementary Note 4.

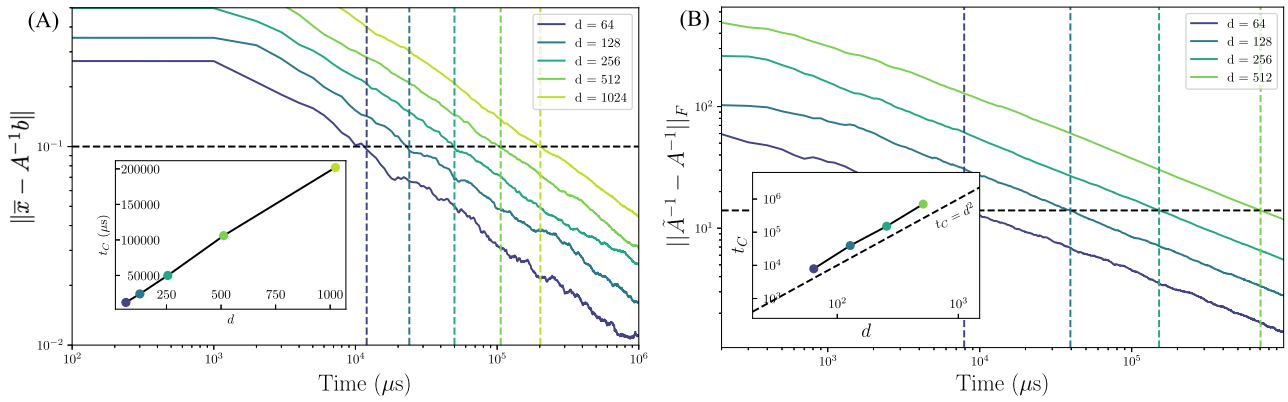


Fig. 3 | Error of our thermodynamic algorithms as a function of the analog integration time for different dimensions. Matrices A are drawn from a Wishart distribution with $2d$ degrees of freedom. Vertical dashed lines are the times t_C at which error goes below a threshold (horizontal dashed line). Inset: Crossing time t_C

as a function of dimension d . **A** For the linear systems algorithm, a linear relationship between dimension and the analog dynamics runtime is observed. **B** For the matrix inversion algorithm, a quadratic relationship between dimension and the analog dynamics runtime is observed.

Convergence and comparison to digital algorithms

Convergence. We now present several numerical experiments to corroborate our analytical results. Figure 3A displays the convergence of the absolute error, $\|\tilde{x} - A^{-1}b\|$ where $\|\cdot\|$ denotes the 2-norm, as a function of time for our thermodynamic linear systems algorithm. This plot shows that the expected convergence time to reach a given error is linearly proportional to the dimension of the system, which is in agreement with the analytical bounds that we presented above.

Similarly, let us examine the performance of the inverse estimation protocol. We employ the absolute error on the inverse, $\|\tilde{A}^{-1} - A^{-1}\|_F$ where $\|\cdot\|_F$ denotes the Frobenius norm. Figure 3B shows the convergence of the error as a function of the analog dynamics time for our thermodynamic inverse estimation algorithm. We see that the expected convergence time to reach a given error is quadratic ($\propto d^2$) in the dimension, in agreement with the analytical bounds presented above.

Comparison to digital algorithms. Another question of key importance is how the thermodynamic algorithm is expected to perform in practical scenarios, i.e., when being run on real thermodynamic hardware. Due to the hardware being analog in nature, this involves additional digital-to-analog compilation steps. To investigate this question, we consider a timing model for the thermodynamic algorithm, based on the hardware proposal described Ref. 17 (See Supplementary Note 2 for a brief summary of this hardware, whose dynamics correspond to the overdamped regime as in Eq. (27)). This model includes all the digital, digital-to-analog and analog operations needed to solve the problem, starting with a matrix A stored on a digital device, and sending back the solution x from the thermodynamic system to the digital device. Note that this includes a compilation step that scales as $O(d^2)$, which is absent for the digital methods; Cholesky decomposition and the conjugate gradient method are run on a digital computer, and the initial matrix is stored on that same computer, hence there is no transfer cost, unlike for the thermodynamic algorithm. Assumptions about this model are detailed in the Methods section. Note that analog imprecision is not taken into account in these experiments, and is the subject of further investigations⁴².

Figure 4 plots the absolute error for solving linear systems as a function of time for the thermodynamic algorithm (TA), the conjugate gradient (CG) method, and the Cholesky decomposition (which is exact). In panels (A–C) we explore how the methods converge with varying κ and d . While at low dimensions our method performs poorly with respect to the Cholesky decomposition and only slightly better than CG, it becomes very competitive for dimensions $d = 1000$ and $d = 5000$. Panels (d)–(f) show the error as a function of time for different condition numbers, at fixed dimension. One can see that as κ grows (as conditioning is worse) our method becomes more

competitive with CG. This suggests that, even in practical scenarios where we account for realistic computational overhead issues, our thermodynamic linear systems algorithm can outperform SOTA digital methods, especially for large d and large κ .

Figure 4 also shows that the thermodynamic algorithm performs significantly better than the CG method at early times, although the CG method ultimately achieves a higher quality result for later times. This suggests that the thermodynamic algorithm is ideally suited to providing an approximate solution in a short amount of time. Nevertheless, we note that the effective temperature of the thermodynamic hardware is an important parameter, and one can lower this temperature to achieve higher precision solutions from the thermodynamic hardware, as can be seen from the curves in Fig. 4.

Using a timing model similar to that employed for the linear systems protocol, we performed a runtime comparison to Cholesky decomposition for the task of matrix inversion. The results are shown in Fig. 5, where the error is plotted as a function of physical time for dimensions 100, 1000, and 5000. The dashed lines represent the corresponding times for Cholesky decomposition, for given dimensions. We see that as the dimension grows, the advantage with respect to the Cholesky decomposition also grows, thus highlighting a practical thermodynamic advantage. Our method for the inverse estimation therefore has the advantage of having well-defined convergence properties as a function of dimension and condition number (compared to other approximate methods for inverting dense matrices, which do not have well defined convergence properties), as well as leading to reasonable error values in practical settings.

Overall, these numerical experiments highlight the potential utility of thermodynamic hardware by showing the opportunity for speedup over SOTA digital methods, based on a simulated timing model of the thermodynamic device.

Discussion

Various types of physics-based computers have been devised, which are supposed to expedite calculations by using physical processes to evaluate expensive functions^{4,7,43–45}. These devices (which include quantum computers and a number of distinct analog architectures) have been shown to offer theoretical advantages for solving certain problems, including linear systems of equations, but they have not found common use commercially. A key obstacle to harnessing the power of physical computing is that fluctuations in the system’s state tend to cause errors that compound over time, and which cannot be corrected in a straightforward way⁴⁶ (as can be done for digital computers).

For this reason, we have considered *thermodynamic* algorithms, which treat the naturally-present fluctuations as a resource, or at the very

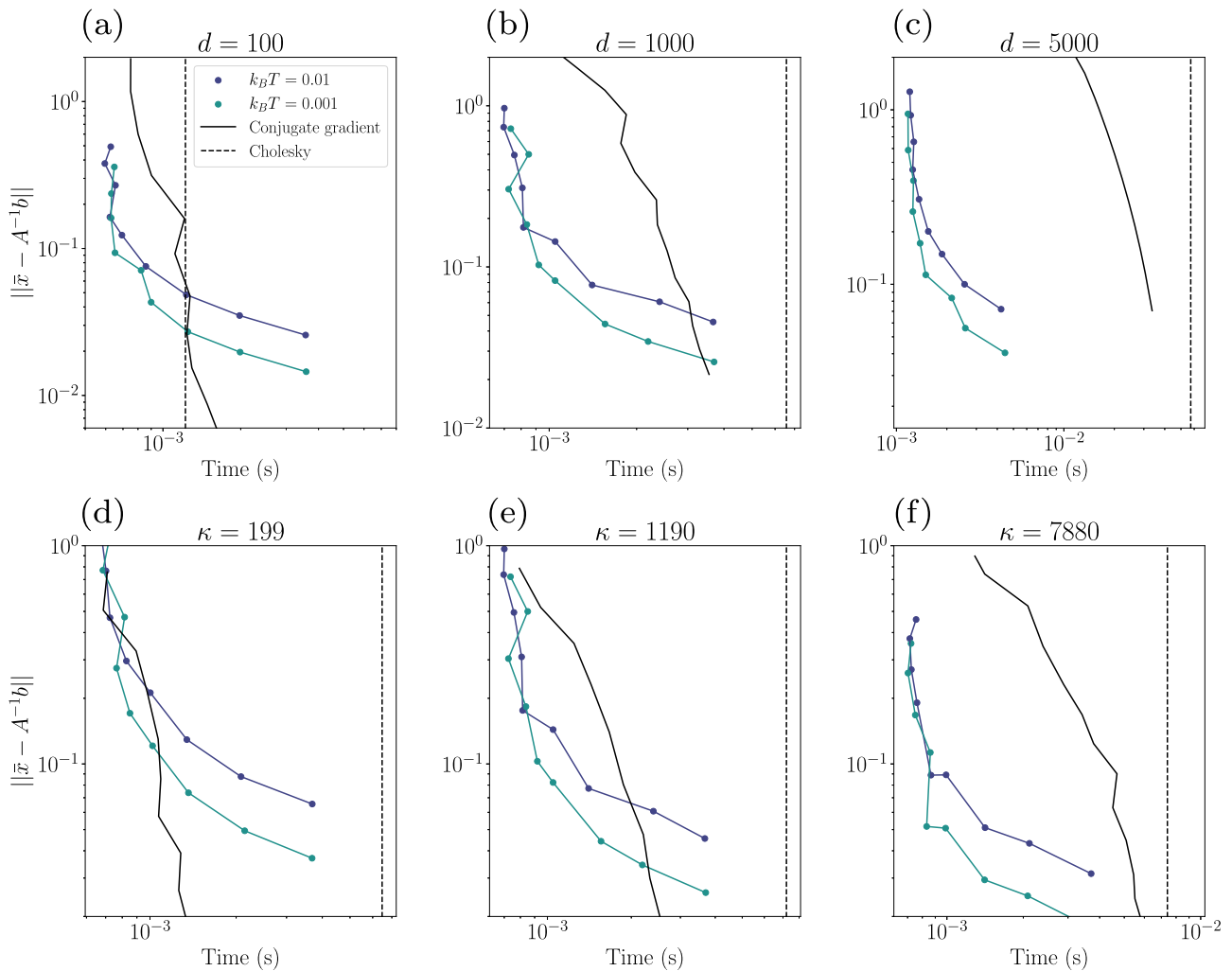


Fig. 4 | Comparison of the error $\|\bar{x} - A^{-1}b\|$ of the thermodynamic algorithm (TA) to solve linear systems with the conjugate gradient method and Cholesky decomposition as a function of total runtime. a–c The TA is shown for different values of $k_B T$ (units of $1/\gamma$) for each dimension in $\{100, 1000, 5000\}$. Random matrices are drawn from the Wishart distribution and then mixed with the

identity such that their condition numbers are respectively 120, 1189, 5995. d–f Same quantities with a fixed condition number κ , respectively 199, 1190, and 7880 for fixed dimension $d = 1000$. Calculations were performed on an Nvidia RTX 6000 GPU.

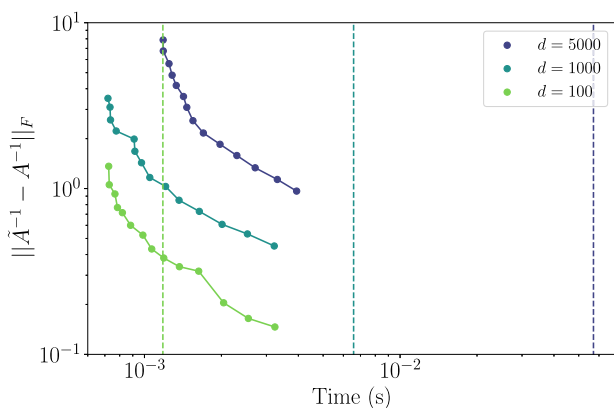


Fig. 5 | Comparison of the error of the thermodynamic algorithm (TA) to invert matrices with the Cholesky decomposition as a function of total runtime. Dimensions $d = 100, 1000, 5000$, respectively in light green, light blue, and purple, are shown for the thermodynamic algorithm (solid lines) and the Cholesky decomposition (dashed lines). Here the condition numbers are respectively $\{120, 1189, 5995\}$. Calculations were performed on an Nvidia RTX A600 GPU.

least are indifferent to them. In fact, we have introduced three distinct classes of thermodynamic algorithms: first-moment based, second-moment based, and all-moment based algorithms. Other thermodynamic algorithms will likely be discovered making use of third and higher moments, implying that such methods form a hierarchy. In some sense, using higher moments allows us to solve “harder” problems, for example inverting a matrix (which uses the second moments) is harder than solving a linear system of equations (which uses the first moment). Whether a precise relationship can be found between computational hardness and the hierarchy of thermodynamic algorithms is currently an open question.

A key property of the system we have studied here that makes it amenable to thermodynamic computing is that the system reaches equilibrium quickly, and the moments of its equilibrium distribution can be efficiently approximated using time averages due to its ergodicity. In extending these methods to other types of dynamics, it is desirable to identify other classes of systems that also equilibrate quickly and for which time averages converge quickly to equilibrium moments. As there is some evidence that this is the case for certain classes of classical chaotic systems as well as quantum systems^{47,48}, we anticipate that thermodynamic algorithms may be developed in these settings as well.

Another open question concerns the optimality of these new thermodynamic algorithms. Our analysis implies that, while the time and energy costs of linear-algebraic primitives are negotiable, the product of time and energy necessary for a computation is fundamentally constrained (see Methods). It is therefore of interest to search for thermodynamic algorithms which achieve lower values of the energy-time product for these computations, and also to see whether such constraints may apply to other problems as well. We anticipate that non-equilibrium thermodynamics will be a crucial tool in exploring such resource tradeoffs for computation. For example, we have used the fact that a thermodynamic distance may be defined between equilibrium configurations of a system, and this distance determines the minimal amount of dissipated energy necessary to transition from one configuration to another in a finite time^{49–53}; the shorter the time of transition, the more dissipation must occur. Perhaps, then, the search for algorithms which have minimal energy-time product may be framed as a variational problem of minimizing length on the thermodynamic manifold. Although proofs of optimal algorithmic performance are notoriously hard to find in digital computing paradigms⁵⁴, unavoidable resource tradeoffs are relatively mundane in thermodynamic analyses^{55,56}, suggesting that computational cost may be fruitfully studied within the thermodynamic computing paradigm.

Aside from the theoretical questions mentioned, clearly the task of actually implementing our algorithms remains an important one. Recently, we created an electrical thermodynamic computing device⁵⁷ on a printed circuit board and used it to demonstrate the thermodynamic matrix inversion algorithm presented here, inverting 8 by 8 matrices using 8 coupled electrical oscillators. A potential next step would be to experimentally verify our predicted scaling of integration time with dimension (e.g., linear scaling for linear systems and quadratic scaling for matrix inversion), thus confirming our predicted speedup over digital methods. We anticipate that other researchers may independently seek to verify our results experimentally, leading to a rapid development of thermodynamic hardware. As a result, we predict that these methods will become appealing alternatives to digital algorithms, particularly in settings where it is desirable to trade some accuracy for better time and energy scaling.

In addition to our work's direct impact, the broader impact is laying the theoretical, mathematical foundations for the emerging paradigm of thermodynamic computing²⁶. Our work provides the first mathematical analysis, as well as the first numerical benchmarks, of potential speedups for thermodynamic hardware. Thus we have taken the somewhat vague notion of thermodynamic computing and made it concrete and precise, with a clear set of applications. Moving forward, we expect new applications to be discovered, beyond linear algebra, since one can simply modify the potential energy function $U(x)$ to solve, e.g., non-linear algebraic problems. There is also the exciting prospect of running multiple applications, such as the linear algebra ones here and the probabilistic AI ones discussed in Ref. 17, on the same thermodynamic hardware, providing the user with a flexible programming experience. One can envision that much of the amazing technological developments (compilers, simulators, programming languages, etc.) that have happened in quantum computing will likely happen for thermodynamic computing in the near future.

Methods

Timing parameters for overdamped and underdamped regimes

Linear systems. For our linear systems algorithm, the overdamped Langevin (ODL) equation takes the form:

$$dx = -\frac{1}{\gamma}(Ax - b)dt + \mathcal{N}[0, 2\gamma^{-1}\beta^{-1} dt], \quad (27)$$

where $\gamma > 0$ is called the damping constant and $\beta = 1/k_B T$ is the inverse temperature of the environment. The system has a physical timescale (which is clear from dimensional analysis) that we call the *relaxation timer*, $\tau_r = \gamma/\|A\|$. The condition number of A is $\kappa = \alpha_{\max}/\alpha_{\min}$, where $\alpha_1 \dots \alpha_d$ are the eigenvalues of A . With these definitions, we arrive (see Supplementary Note 2 for derivation) at the following formulas for \hat{t}_0 and $\hat{\tau}$ in the overdamped

case, which can be used in the linear systems algorithm:

$$\hat{t}_0 = \max\left\{\kappa\tau_r \ln\left(\kappa\epsilon_{\mu 0}^{-1}\right), \frac{1}{2}\kappa\tau_r \ln\left(2\kappa\epsilon_{\Sigma 0}^{-1}\right)\right\}, \quad \hat{\tau} = \frac{2\kappa^2 d \|A\|}{\beta \|b\|^2 \epsilon_x^2 (1 - P_\epsilon)} \tau_r. \quad (28)$$

The underdamped model is instead described by the UDL equations,

$$dx = \frac{1}{M}p dt, \quad dp = -(Ax - b) dt - \frac{\gamma}{M}p dt + \mathcal{N}[0, 2\gamma\beta^{-1} \mathbb{I} dt]. \quad (29)$$

We define $\xi = \gamma/2M$, $\omega_j = \sqrt{\alpha_j/M}$, and $\zeta_j = \xi/\omega_j$. Moreover, a timescale $\tau_{r(\text{UD})}$ can be identified for the underdamped system which is analogous to the quantity τ_r associated with the overdamped system. In particular, we define $\tau_{r(\text{UD})} = \xi^{-1}$. We introduce a dimensionless quantity χ as well, which is $\chi = (1 + \xi/\omega_{\min})^{1/2} (1 - \xi/\omega_{\min})^{-1/2}$. With these definitions, we arrive (see Supplementary Note 3 for derivation) at the following formulas for the timing parameters in the underdamped case:

$$\hat{t}_0 = \max\left\{\tau_{r(\text{UD})} \ln\left(\kappa^{1/2} \chi \epsilon_{\mu 0}^{-1}\right), \frac{1}{2} \tau_{r(\text{UD})} \ln\left(\chi^2 \kappa^{3/2} \epsilon_{\Sigma 0}^{-1} \left[\frac{1}{4\zeta_{\max}^2} + 1\right]\right)\right\},$$

$$\hat{\tau} = \frac{2\sqrt{\kappa} \chi d \|A\|}{\beta \|b\|^2 \epsilon_x^2 (1 - P_\epsilon)} \tau_{r(\text{UD})}. \quad (30)$$

An important distinction between the ODL and UDL regimes is that the random variable x undergoes a Markov stochastic process in the ODL case, but is non-Markovian in the UDL case^{58,59}. The simple interpretation of this non-Markovianity is that the underdamped system exhibits inertia, which is a form of memory-dependence. This inertia has a non-trivial (and sometimes beneficial) effect on the algorithm's overall performance, which is apparent from the scaling results in Table 1.

Remark on dependence on the norm of A . Note that the norm $\|A\|$ appears in Eq. (28) explicitly in the expression for $\hat{\tau}$ and also enters through the definition of τ_r . We see that $\hat{\tau}_r$ does not depend on $\|A\|$ because the two factors cancel, but \hat{t}_0 is inversely proportional to $\|A\|$. In order to eliminate the dependence on the norm of $\|A\|$, we could simply divide A by its norm before applying the algorithm. Then, upon obtaining the solution to the linear system $x = A^{-1}b$, the vector x would be divided by $\|A\|$ to recover the solution to the original problem. However, the computation of $\|A\|$ would have a larger polynomial time complexity than our algorithm, so this would eliminate our polynomial speedup. Instead, we can divide A by its largest diagonal element, defining $A' = a_0 A / \max\{\text{diag}(A)\}$, where a_0 is a constant with units of γ/t . Now note that

$$\|A'\| \geq \frac{1}{d} \text{tr}\{A'\} \geq \frac{1}{d} \max\{\text{diag}(A')\} = \frac{a_0}{d}. \quad (31)$$

As a lower bound on $\|A\|$ gives an upper bound on \hat{t}_0 , we may substitute $\|A'\|$ into the above equation for \hat{t}_0 , yielding the following:

$$\hat{t}_0 = \max\left\{\kappa \frac{\gamma d}{a_0} \ln\left(\kappa \epsilon_{\mu 0}^{-1}\right), \frac{1}{2} \kappa \frac{\gamma d}{a_0} \ln\left(2\kappa \epsilon_{\Sigma 0}^{-1}\right)\right\},$$

$$\hat{\tau} = \frac{2\kappa^2 d}{\beta \|b\|^2 \epsilon_x^2 (1 - P_\epsilon)} \gamma. \quad (32)$$

The linear system resulting from normalizing A by its largest diagonal element can therefore be solved in linear time. When the solution x' is found, it can then be divided by the largest diagonal element of A to obtain the solution to the original problem. A similar argument applies to Eq. (30) for the underdamped case.

Matrix inversion. The timing parameters for the inverse estimation protocol (as derived in Supplementary Notes 2 and 3) are, for the

overdamped case,

$$\hat{t}_0 = \frac{1}{2} \kappa \tau_r \ln(2\kappa \epsilon_{\Sigma 0}^{-1}), \quad \hat{\tau} = \frac{4\kappa d(d+1)}{(1-P_\epsilon) \epsilon_\Sigma^2} \tau_r, \quad (33)$$

and for the underdamped case

$$\hat{t}_0 = \frac{1}{2} \tau_{r(\text{UD})} \ln\left(\chi^2 \kappa^{3/2} \epsilon_{\Sigma 0}^{-1} \left[\frac{1}{4\zeta_{\max}^2} + 1\right]\right), \quad \hat{\tau} = \frac{4\kappa d(d+1)}{(1-P_\epsilon) \epsilon_\Sigma^2} \tau_{r(\text{UD})}. \quad (34)$$

Energy-time tradeoff

Note the appearance of the ratio $\|A\|/\|b\|^2$ in the time required to solve a linear system given by Eqs. (28) and (30). It is tempting to imagine that one might solve the system faster simply by multiplying b by some constant c . Then, the time required to solve the system is apparently reduced by a factor of c^2 , and the solution to original problem is obtained (up to a factor of c). A similar approach would be to multiply A by a small number; of course, in practice it is not possible to solve linear systems of equations in vanishingly short periods of time, which is reflected in an energy-time tradeoff. As we explain in Supplementary Note 2, there is an energy cost associated with initializing the system proportional to $b^T A^{-1} b$, and this results in a re-formulation of Eqs. (28) and (30) as lower bounds on the product of energy and time. If \mathcal{E} is the energy required to solve the system $Ax = b$, and τ is the necessary time, then we have, in the overdamped case:

$$\mathcal{E} \tau \geq \frac{2\kappa^2 d}{\epsilon_x^2 (1-P_\epsilon)} \beta^{-1} \tau_r \quad (35)$$

and in the underdamped case:

$$\mathcal{E} \tau \geq \frac{2\sqrt{\kappa} \chi d}{\epsilon_x^2 (1-P_\epsilon)} \beta^{-1} \tau_{r(\text{UD})}. \quad (36)$$

This fundamental energy-time tradeoff appears naturally within this computational model. While digital computations can often be accelerated by investing more energy (for example, via parallelization), it is generally less obvious what exact form the relationship between time and energy cost takes, suggesting that thermodynamic algorithms may offer a new and useful perspective on algorithmic complexity.

Detailed algorithmic scaling

In the main text, we presented a simplified version of the detailed table shown in Table 2. Table 2 breaks down the scaling into the overdamped and underdamped regimes, whereas Table 1 just takes the best scalings of our algorithms. The latter is essentially the scaling associated with the underdamped regime. However, in practice, there can be some engineering advantages to working in the overdamped regime, and hence it is useful to see the complexities of both regimes.

Table 2 | Asymptotic complexities of linear algebra algorithms, including overdamped and underdamped regimes

Problem	Digital SOTA	This work (Overdamped)	This work (Underdamped)
Linear System	$O(\min\{d^w, d^2 \sqrt{\kappa}\})$	$O(d\kappa^2 \epsilon^{-2})$	$O(\gamma d \kappa^{3/2} \epsilon^{-2})$
Matrix Inverse	$O(d^w)$	$O(d^2 \kappa \epsilon^{-2})$	$O(\gamma d^2 \kappa^2 \epsilon^{-2})$
Lyapunov Equation	$O(d^3)$	$O(d^2 \kappa \epsilon^{-2})$	$O(\gamma d^2 \kappa^2 \epsilon^{-2})$
Matrix Determinant	$O(d^w)$	$O(d \kappa \ln(\kappa)^3 \epsilon^{-2})$	$O(\gamma d \kappa \ln(\kappa)^3 \epsilon^{-2})$

For our thermodynamic algorithms, the complexity depends on the dynamical regime, i.e., whether the dynamics are overdamped and underdamped, as shown in this table.

Assumptions

A number of assumptions are made in the analytical derivations of the findings presented in the Results section. Certain aspects of the problem have been idealized in order to reveal the fundamental performance characteristics of the thermodynamic algorithms. The main assumptions are the following

- The dynamics of the system may be described by the ODL equation or the UDL equations.
- The potential function $U(x)$, and in particular the matrix A and vector b , can be switched instantaneously between different values.
- The potential energy function $U(x)$ can be implemented to arbitrary accuracy.
- Before a protocol begins, the system may be taken to be in an equilibrium distribution $\mathcal{N}[0, \beta^{-1} \|A\|^{-1} \mathbb{I}]$.

Numerical simulations

We outline the methods used for our numerical simulations of the thermodynamic algorithms. In general, we simulated the overdamped system dynamics because the performance is similar to the underdamped case, and the overdamped system is more numerically stable. The ODL equation, $dx = -Ax + \mathcal{N}[0, Bdt]$, is often written using an Itô integral⁶⁰:

$$x(t) = e^{-At} x_0 + \int_0^t dt' e^{-A(t-t')} L dW_{t'}, \quad (37)$$

where $LL^T = B$, and in this form it is apparent that the deterministic and stochastic parts of the evolution (the first and second terms above) can be evaluated separately. As this is a Gaussian process, the corresponding Fokker-Planck dynamics are fully captured by the behavior of the first and second moments, which can be evaluated directly using the well-known solution to the Ornstein-Uhlenbeck (OU) process. The JAX⁶¹ library was used to simulate the system at high dimensions, leveraging efficient implementations of matrix exponentials, diagonalization, and convolution to evaluate the various terms in the solution of the OU process. In what follows, we describe the timing model employed for benchmarking our thermodynamic algorithm, assuming an implementation using electrical circuits.

Timing model. To obtain the comparisons to other digital methods, we considered the following procedure to run the TA on electrical hardware. For more details on our model for the hardware implementation we refer the reader to Supplementary Note 2. We take the $RC = 1/\gamma = 1\mu\text{s}$, which sets the characteristic timescale of the thermodynamic device. The determinant estimation procedure is excluded here for clarity, as it involves directly measuring work, which may involve a more complicated hardware proposal.

1. Compute the values of the resistors $\{R_{ij}, R_i'\}$ entering the matrix \mathcal{J} that encodes the A matrix.
2. Digital-to-analog (DAC) conversion of the \mathcal{J} matrix and the b vector with a given bit-precision.
3. Let the dynamics run for t_0 (the equilibration time). Note that for simulations this time was chosen heuristically by exploring convergence in the solutions of the problem of interest.
4. Switch on the integrators (and multipliers for the inverse estimation) and let the system evolve for time τ .
5. Analog-to-digital (ADC) conversion of the solution outputted from the integrators sent back to the digital device. For step 1, we measured the time for the digital operation to be performed, and for the other steps we estimated the time, based on the following assumptions:
 - 16 bit-precision
 - 5000 ADC/DAC channels with a sampling rate: 250 Msamples/s.
 - $R = 10^3 \Omega$, $C = 1 \text{ nF}$, which means $RC = 1 \mu\text{s}$ is the characteristic timescale of the system. Finally, note that in all cases that were investigated, the dominant contribution to the total runtime was the digital compilation step. This step includes $O(d^2)$ operations and involves

conversion of the matrix A to \mathcal{J} , which is detailed in Supplementary Note 2. Hence some assumptions about the DAC/ADC may be relaxed and the total thermodynamic runtime would be similar. The RC time constant may also be reduced to make the algorithm faster.

Also note that the polynomial complexity of transferring data to and from the thermodynamic computing device should be considered. For the matrix inversion, Lyapunov equation, and determinant estimation algorithms this does not cause any issue, as the amount of data to be transferred scales with d^2 , which is the same as the time-complexity of the algorithms. For the linear system algorithm, in order to preserve the $O(d)$ time complexity it is necessary for the uploading of the matrix to hardware to be parallelized across $O(d)$ channels, in which case the read/write operations do not affect the polynomial time complexity. An instructive example of a thermodynamic algorithm working as part of a hybrid digital-analog algorithm is presented in Ref. 62, where the linear systems algorithm is used as a subroutine in a second-order optimization algorithm.

Noise resilience and error mitigation

To the extent that a circuit can be modeled by equilibrium thermodynamics, and the form and parameters of its Hamiltonian are known, the approach of thermodynamic computing is fully resilient to thermal fluctuations (for example, Johnson-Nyquist noise). However, there are various effects that can cause the continuous-variable Hamiltonian description to be invalid; for example at small currents the discretization of charge leads to significant shot noise which is not accounted for in our formalism. Circuit elements also do not behave in a completely idealized way, for example a capacitor will generally have a capacitance that depends on the applied voltage, leading to a Hamiltonian with higher order terms. Moreover, even if the form of the Hamiltonian is known, the parameters appearing in it cannot be known with certainty, and impedances of circuit components do not exactly match their nominal values, an issue called component mismatch. Finally, even if the electrical components behaved in a completely ideal way and their impedances were known exactly, we may only vary the parameters of a circuit with finite precision; for example, with B bits of precision, we can choose between 2^B possible values for a capacitance in the circuit. Although thermodynamic algorithms offer an appealing resilience to thermal noise, it is necessary to create strategies to mitigate these other sources of error.

In order to address the problems described above, we have begun a comprehensive program of identifying sources of error in thermodynamic computations and finding methods of mitigating these sources of error. Errors caused by imprecision of parameter specification can be significantly reduced using the method proposed in⁴², which is based on stochastically rounding component values between those allowed by the implementation. This method can be used to address imprecision errors, and indeed its experimental implementation resulted in 20% error reduction for matrix inversion⁴². However, it does not help reduce errors caused by component mismatch, so additional strategies are needed. We are presently researching approaches to mitigating component mismatch error, which will be covered in forthcoming work.

Nonlinear behavior of components may prove the most difficult source of error to address, as it involves the form of the Hamiltonian rather than just its parameters, and so cannot be fixed using a simple calibration procedure. To some extent, the degree of nonlinear behavior can be controlled by reducing the amplitude of signals, as nonlinearity typically becomes stronger as the amplitude increases. Unfortunately, reducing the signal amplitude comes with the drawback that a more sensitive measurement device would be needed to obtain the same accuracy. Strategies for reducing error caused by nonlinearities are therefore of great practical importance.

Data availability

All data reflected in our results was generated in a way that is clear from the text, and the results can be reproduced by simulating the algorithms as they are described. No external data was used. Results from specific simulations are available upon request.

Received: 12 March 2024; Accepted: 26 September 2024;

Published online: 05 November 2024

References

- Vadlamani, S. K., Xiao, T. P. & Yablonovitch, E. Physics successfully implements Lagrange multiplier optimization. *Proc. Natl Acad. USA* **117**, 26639–26650 (2020).
- Mohseni, N., McMahon, P. L. & Byrnes, T. Ising machines as hardware solvers of combinatorial optimization problems. *Nat. Rev. Phys.* **4**, 363–379 (2022).
- Inagaki, T. et al. A coherent ising machine for 2000-node optimization problems. *Science* **354**, 603–606 (2016).
- Feynman, R. P. Simulating physics with computers. *Int. J. Theor. Phys.* **21**, 467–488 (1982).
- Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009).
- Scherer, A. et al. Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2D target. *Quant. Inf. Process.* **16**, 1–65 (2017).
- Preskill, J. Quantum computing in the Nisq era and beyond. *Quantum* **2**, 79 (2018).
- Bravo-Prieto, C. et al. Variational quantum linear solver. *arXiv* <https://arxiv.org/abs/1909.05820> (2019).
- Xu, X. et al. Variational algorithms for linear algebra. *Sci. Bull.* **66**, 2181–2188 (2021).
- Cerezo, M. et al. Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625–644 (2021).
- McClellan, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **9**, 1–6 (2018).
- Cerezo, M., Sone, A., Volkoff, T., Cincio, L. & Coles, P. J. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nat. Commun.* **12**, 1–12 (2021).
- Wang, S. et al. Noise-induced barren plateaus in variational quantum algorithms. *Nat. Commun.* **12**, 1–11 (2021).
- Li, C. et al. Analogue signal and image processing with large memristor crossbars. *Nat. Electron.* **1**, 52–59 (2018).
- Yi, S.-i, Kendall, J. D., Williams, R. S. & Kumar, S. Activity-difference training of deep neural networks using memristor crossbars. *Nat. Electron.* **6**, 45–51 (2023).
- Huang, Y., Guo, N., Seok, M., Tsividis, Y. & Sethumadhavan, S. Evaluation of an analog accelerator for linear algebra. *Comput. Archit. N.* **44**, 570–582 (2016).
- Coles, P. J. et al. Thermodynamic AI and the fluctuation frontier. *arXiv* <https://doi.org/10.48550/arXiv.2302.0658> (2023).
- Aadit, N. A. et al. Massively parallel probabilistic computing with sparse Ising machines. *Nat. Electron.* **5**, 460–468 (2022).
- Camsari, K. Y., Sutton, B. M. & Datta, S. p-bits for probabilistic spin logic. *Appl. Phys. Rev.* **6**, 011305 (2019).
- Kaiser, J., Datta, S. & Behin-Aein, B. 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)*, 21–4 (IEEE, 2022).
- Hylton, T. Thermodynamic neural network. *Entropy* **22**, 256 (2020).
- Hylton, T. Thermodynamic state machine network. *Entropy* **24**, 744 (2022).
- Ganesh, N. Booting neuromorphic design-a complexity engineering approach. In *2020 International Conference on Rebooting Computing (ICRC)*, 80–89 (IEEE, 2020).
- Ganesh, N. A thermodynamic treatment of intelligent systems. In *2017 IEEE International Conference on Rebooting Computing (ICRC)* 1–4 (IEEE, 2017).
- Lipka-Bartosik, P., Perarnau-Llobet, M. & Brunner, N. Thermodynamic computing via autonomous quantum thermal machines. *Sci. Adv.* **10**,

- eadm8792 (American Association for the Advancement of Science, 2024).
26. Conte, T. et al. Thermodynamic computing. *arXiv* <https://arxiv.org/abs/1911.01968>. (2019).
 27. Babbush, R., Berry, D. W., Kothari, R., Somma, R. D. & Wiebe, N. Exponential quantum speedup in simulating coupled classical oscillators. *Phys. Rev. X*, **13**, 041041 (APS, 2023).
 28. Parks, P. C. Am Lyapunov's stability theory-100 years on. *IMA J. Math. Control Inf.* **9**, 275–303 (1992).
 29. Forsythe, G. E. & Leibler, R. A. Matrix inversion by a Monte Carlo method. *Math. Comput.* **4**, 127–129 (1950).
 30. Alexandrov, V. N. & Lakka, S. Comparison of three Monte Carlo methods for matrix inversion. In *Euro-Par'96 Parallel Processing: Second International Euro-Par Conference Lyon, France, August 26–29, 1996 Proceedings, Volume II 2*, 72–80 (Springer, 1996).
 31. Ökten, G. Solving linear equations by Monte Carlo simulation. *SIAM J. Sci. Comput.* **27**, 511–531 (2005).
 32. Rosca, N. *Monte Carlo Methods for Systems of Linear Equations*. <https://www.cs.ubbcluj.ro/~studia-m/2006-1/rosca.pdf> (2006).
 33. Dimov, I., Maire, S. & Sellier, J. M. A new walk on equations Monte Carlo method for solving systems of linear algebraic equations. *Appl. Math. Model.* **39**, 4494–4510 (2015).
 34. Duffield, S., Aifer, M., Crooks, G., Ahle, T. & Coles, P. J. Thermodynamic matrix exponentials and thermodynamic parallelism. *arXiv* <https://arxiv.org/abs/2311.12759> (2023).
 35. Valiant, G. Matrix multiplication in quadratic time and energy? towards a fine-grained energy-centric church-turing thesis. *arXiv* <https://arxiv.org/abs/2311.16342> (2023).
 36. Gallavotti, G. Ergodicity, ensembles, irreversibility in boltzmann and beyond. *J. Stat. Phys.* **78**, 1571–1589 (1995).
 37. Sinai, Y. G. On the foundations of the ergodic hypothesis for a dynamical system of statistical mechanics. In *Doklady Akademii Nauk*. 1261–1264 (Russian Academy of Sciences, 1963).
 38. Kubo, R. The fluctuation-dissipation theorem. *Rep. Prog. Phys.* **29**, 255 (1966).
 39. Weber, J. Fluctuation dissipation theorem. *Phys. Rev.* **101**, 1620 (1956).
 40. Christ, C. D., Mark, A. E. & Van Gunsteren, W. F. Basic ingredients of free energy calculations: a review. *J. comput. Chem.* **31**, 1569–1582 (2010).
 41. Jarzynski, C. Nonequilibrium equality for free energy differences. *Phys. Rev. Lett.* **78**, 2690 (1997).
 42. Aifer, M. et al. Error mitigation for thermodynamic computing. *arXiv* <https://arxiv.org/abs/2401.16231> (2024).
 43. Haensch, W., Gokmen, T. & Puri, R. The next generation of deep learning hardware: analog computing. *Proc. IEEE* **107**, 108–122 (2018).
 44. Small, J. S. General-purpose electronic analog computing: 1945–1965. *IEEE Ann. Hist. Comput.* **15**, 8–18 (1993).
 45. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* 1st edn (Cambridge University Press, Cambridge, 2000).
 46. Huang, Y., Guo, N., Seok, M., Tsvividis, Y. & Sethumadhavan, S. Analog computing in a modern context: a linear algebra accelerator case study. *IEEE Micro* **37**, 30–38 (2017).
 47. Zhong, W., Gao, X., Yelin, S. F. & Najafi, K. Many-body localized hidden born machine. *arXiv* <https://doi.org/10.48550/arXiv.2207.02346> (2022).
 48. Tangpanitanon, J., Thanasilp, S., Dangniam, N., Lemonde, M.-A. & Angelakis, D. G. Expressibility and trainability of parametrized analog quantum systems for machine learning applications. *Phys. Rev. Res.* **2**, 043364 (2020).
 49. Crooks, G. E. Measuring thermodynamic length. *Phys. Rev. Lett.* **99**, 100602 (2007).
 50. Cafaro, C., Luongo, O., Mancini, S. & Quevedo, H. Thermodynamic length, geometric efficiency and legendre invariance. *Phys. A. Stat. Mech. its Appl.* **590**, 126740 (2022).
 51. Quevedo, H. Geometrothermodynamics. *J. Math. Phys.* <https://doi.org/10.48550/arXiv.physics/0604164> (2007).
 52. Andresen, B. Finite-time thermodynamics and thermodynamic length. *Rev. générale de. Therm.* **35**, 647–650 (1996).
 53. Chen, J.-F., Sun, C. & Dong, H. Extrapolating the thermodynamic length with finite-time measurements. *Phys. Rev. E* **104**, 034117 (2021).
 54. Saptharishi, R. *A Survey of Lower Bounds in Arithmetic Circuit Complexity*. <https://pgadey.ca/seminar/Saptharishi-survey.pdf> (2015).
 55. Chiribella, G., Meng, F., Renner, R. & Yung, M.-H. The nonequilibrium cost of accurate information processing. *Nat. Commun.* **13**, 7155 (2022).
 56. Riechers, P. M. Transforming metastable memories: the nonequilibrium thermodynamics of computation. *arXiv* <https://doi.org/10.48550/arXiv.1808.03429> (2018).
 57. Melanson, D. et al. Thermodynamic computing system for AI applications. *arXiv* <https://arxiv.org/abs/2312.04836> (2023).
 58. Doerries, T. J., Loos, S. A. & Klapp, S. H. Correlation functions of non-markovian systems out of equilibrium: Analytical expressions beyond single-exponential memory. *J. Stat. Mech. Theory Exp.* **2021**, 033202 (2021).
 59. H'walisz, L., Jung, P., Hänggi, P., Talkner, P. & Schimansky-Geier, L. Colored noise driven systems with inertia. *Z. für. Phys. B Condens. Matter* **77**, 471–483 (1989).
 60. Gardiner, C. W. *Handbook of Stochastic Methods*, vol. 3 (Springer Berlin, 1985).
 61. Frostig, R., Johnson, M. J. & Leary, C. Compiling machine learning programs via high-level tracing. *Syst. Mach. Learn.* <https://doi.org/10.48550/arXiv:2402.06268v1> (2018).
 62. Donatella, K. et al. Thermodynamic natural gradient descent. *arXiv* <https://doi.org/10.48550/arXiv.2405.13817> (2024).
 63. Shewchuk, J. R. et al. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. <https://dl.acm.org/doi/10.5555/865018> (1994).
 64. Robinson, S. Toward an optimal algorithm for matrix multiplication. *SIAM N.* **38**, 1–3 (2005).
 65. Bartels, R. H. & Stewart, G. W. Solution of the matrix equation $ax + xb = c$ [f4]. *Commun. ACM* **15**, 820–826 (1972).
 66. Aho, A. V., Hopcroft, J. E. & Ullman, J. D. *The Design and Analysis of Computer Algorithms* 1st edn, Vol. 480 (Addison-Wesley, 1974).
 67. Fokker, A. D. Die mittlere energie rotierender elektrischer dipole im strahlungsfeld. *Ann. der Phys.* **348**, 810–820 (1914).

Acknowledgements

Not applicable.

Author contributions

The manuscript was written by M.A., K.D., M.H.G., and P.J.C. The algorithms were proposed by M.A., T.A., G.C., P.J.C., and D.S. The analytical results were derived by M.A. The numerical experiments were performed by K.D., M.A., M.H.G., and S.D. The figures were created by M.A., K.D., and M.H.G. All authors have read and approved the manuscript.

Competing interests

Normal Computing has filed a PCT patent application related to this manuscript. The application number is PCT/US2023/076759. The status is pending. The aspect of the manuscript covered in the patent application is the electronic hardware design for implementing the algorithms described in the manuscript. Maxwell Aifer, Kaelan Donatella, Max Hunter Gordon, and Patrick J. Coles are listed as inventors on this patent application. All other authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s44335-024-00014-0>.

Correspondence and requests for materials should be addressed to Maxwell Aifer.

Reprints and permissions information is available at
<http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024