Computing Systems for Analog Intelligence (Postprint)

Authors: Tan Guangming, Jia Weile, Wang Zhan, Yuan Guojun, Shao En, Sun Ninghui

Date: 2024-03-27T00:00:00+00:00

Abstract

Computer simulation in scientific research is termed scientific simulation. This article, from both narrow and broad perspectives, categorizes scientific simulation into three stages: numerical computation, simulation intelligence, and science brain, and delineates the characteristics of each stage. At present, scientific simulation is transitioning into the simulation intelligence stage, wherein, propelled by scientific big data and artificial intelligence, it is gradually shifting from traditional numerical computation toward simulation approaches integrated with artificial intelligence. The article addresses the computing systems that underpin the simulation intelligence stage, expounding upon their design guiding principles, fundamental methodologies, and key technological issues.

Full Text

Preamble

Special Topic: Vigorously Promote Scientific Research Paradigm Transformations

Citation Format: Tan Guangming, Jia Weile, Wang Zhan, et al. Computing system for simulation intelligence. *Bulletin of Chinese Academy of Sciences*, 2024, 39(1): 17-26, doi: 10.16418/j.issn.1000-3045.20231201001.

Tan G M, Jia W L, Wang Z, et al. Computing system for simulation intelligence. *Bulletin of Chinese Academy of Sciences*, 2024, 39(1): 17-26, doi: 10.16418/j.issn.1000-3045.20231201001. (in Chinese)

Computing System for Simulation Intelligence

Tan Guangming*, Jia Weile, Wang Zhan, Yuan Guojun, Shao En, Sun Ninghui*

Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Abstract

Computer simulation in scientific research is referred to as scientific simulation. Based on its narrow and broad definitions, this paper divides scientific simulation into three stages: numerical computation, simulation intelligence, and science brain, describing the characteristics of each stage. Currently, scientific simulation is entering the simulation intelligence stage, where driven by scientific big data and artificial intelligence, it is gradually shifting from traditional numerical computation to simulation methods integrated with AI. This paper discusses the computing systems that support the simulation intelligence stage, elaborating on their design philosophy, fundamental methods, and key technical issues.

Keywords: scientific simulation, simulation intelligence, artificial intelligence, computing system, Zetta-scale computing

DOI: 10.16418/j.issn.1000-3045.20231201001
CSTR: 32128.14.CASbulletin.20231201001

1. Computer Simulation in Scientific Research

As humanity enters the era of big science, "simulation" has become the third pillar of scientific research, complementing "theory" and "experimentation." From a conceptual standpoint, scientific research can be viewed as a process of modeling, while simulation represents the execution of these scientific models on computers. The earliest computer simulations emerged after World War II as pioneering scientific tools specifically for nuclear physics and meteorology research. Subsequently, computer simulation has become increasingly important across numerous disciplines, giving rise to interdisciplinary fields such as computational physics, computational chemistry, and computational biology.

Weaver[1] noted in 1948 that humanity' s ability to solve problems of organized complexity and achieve new scientific leaps would primarily depend on advances in computer technology and the collision of expertise among scientists from different disciplines. On one hand, computer technology enables us to tackle complex and previously intractable problems. On the other hand, it effectively stimulates new solution methods for problems of organized complexity. These new solution methods fall within the scope of computational science, allowing scientists to concentrate resources and focus insights from different fields on common problems. This convergence of perspectives fosters "hybrid teams" of scientists from diverse disciplinary backgrounds that are more powerful than single-discipline teams, capable of solving certain complex problems and deriving useful conclusions. In essence, science and modeling are intimately connected: simulation executes models that represent theories, and we refer to

computer simulation in scientific research as scientific simulation.

Currently, no single definition of "computer simulation" adequately captures the concept of scientific simulation. The U.S. Department of Defense defines simulation as "a method for implementing a model over time" and computer simulation as "the process of executing code on a computer, controlling and displaying interface hardware, and interfacing with real-world equipment." Winsberg[2] distinguishes between narrow and broad definitions of computer simulation.

In the narrow definition, computer simulation is simply "the process of running a program on a computer." It uses stepwise methods to explore the approximate behavior of mathematical models, where a single program execution represents one simulation of the target system. There are two primary reasons for employing computer simulation: either the original model itself contains discrete equations, or the evolution of the original model is better described by "rules" rather than "equations." Notably, this narrow perspective requires constraints on specific processor hardware implementations, programming languages, kernel functions, and compilers, with varying performance results across different application scenarios due to these constraints.

In the broad definition, computer simulation can be viewed as a comprehensive method for studying systems—a more complete computational process that includes model selection, implementation through models, algorithmic computation, result visualization, and research. This entire simulation process corresponds to the scientific research process as described by Lynch[3]: (1) formulating an empirically answerable question; (2) deriving a falsifiable hypothesis from a theory aimed at answering that question; (3) collecting (or discovering) and analyzing empirical data to test the hypothesis; and (4) connecting the analytical results to the theory that generated the question. In the past, this broad view of computer simulation typically appeared in epistemological or methodological theoretical contexts.

Winsberg[2] further divides computer simulation into equation-based simulation and agent-based simulation. Equation-based simulation is commonly used in theoretical disciplines such as physics, where dominant theories guide the construction of mathematical models based on differential equations. These can be particle-based simulations involving vast numbers of independent particles and differential equations describing their interactions, or field-based simulations involving equations describing the temporal evolution of continuous media or fields. Agent-based simulation, which follows certain evolutionary rules, is the most common approach in social and behavioral sciences, such as Schelling' s segregation model[4]. Unlike equation-based particle simulations, there are no global differential equations controlling particle movement.

From these definitions and classifications, we can discern different levels of expectation for scientific simulation. From the narrow perspective, computer simulation has become a complementary approach to traditional cognitive methods like theoretical analysis and experimental observation. Virtually all scientific

and engineering fields are driven by computer simulation, with some specific application domains and scenarios even transformed by it. Many key technologies could not be understood, developed, or utilized without computer simulation. The broad definition of computer simulation implies a philosophical question: Can computers conduct scientific research autonomously? The goal of scientific research is to understand the world, which means computer programs must create new knowledge. With the latest surge in AI research and applications, expectations have grown for computers to conduct scientific research "intelligently" and automatically. Notably, Kitano[5] proposed the "Nobel Turing Challenge" in 2021—the goal of "developing AI scientists capable of autonomously performing research tasks and making Nobel Prize-level major scientific discoveries by 2050." This concept involves technologies related to both narrow and broad definitions of computer simulation, treating the broad definition' s "philosophical question" as a grand goal of scientific simulation rather than exploring it in depth.

2. Development Stages of Scientific Simulation

From an intuitive perspective, the carrier of scientific simulation is computer programs. Mathematically, computer programs consist of computable functions that map discrete sets of finite input data to discrete sets of finite output data. Technically, computer programs equal algorithms plus data structures. Therefore, implementing scientific simulation requires formal abstraction of scientific problems and their solutions. Borrowing Simon' s[6] perspective that scientists are "problem solvers" who set major scientific questions for themselves, we can view the essence of scientific discovery as determining problems and solution strategies. Based on this "solver" framework and analogizing equation-solving forms, we divide scientific simulation development into three stages: numerical computation, simulation intelligence, and science brain (Figure 1 [Figure 1: see original paper]).

2.1 Numerical Computation

Computer simulation was initially used to perform numerical calculations required in scientific research. Consequently, in many academic contexts, people use numerical computation, numerical simulation, and computer simulation interchangeably. The first electronic digital computer, ENIAC, was built to calculate ballistic equations for wind tunnel design and weather forecasting. In the traditional numerical computation stage, computer simulation program execution represents the process of solving mathematical and physical equations—the simple "solver" $y = F(x)$. Indeed, over the past decades, numerical computation in scientific research has driven continuous improvements in supercomputer performance. Through innovations in hardware, software, and algorithms, we have persistently increased the speed and accuracy of solving $y = F(x)$. Consequently, benchmark programs for solving dense or sparse linear algebra equations, such as Linpack[7] and HPCG[8], are used to evaluate supercomputer

speed.

However, this problem-solving pattern, which transforms complex scientific problems into relatively simple computational ones through coarse-grained modeling, encounters computational bottlenecks in certain scenarios. When addressing complex physical models in real-world applications, we often face the problem of excessive computational demands from fundamental physical principles, rendering the principles ineffective for solving scientific problems despite their existence. For example, first-principles molecular dynamics requires solving the quantum mechanical Kohn-Sham equation, with its core algorithm involving repeated solutions of large-scale eigenvalue problems with computational complexity of N^3 (where N is matrix dimension). In practical physical problems, commonly used plane-wave basis sets are typically 100–10,000 times the number of atoms. This means for systems with thousands of atoms, matrix dimension N reaches 10^6, and the corresponding total floating-point operations reach 10^{18} FLOPS—exaflop-level computation . Notably, a single molecular dynamics step requires multiple eigenvalue solutions, making the simulation time per step typically several minutes to one hour. Since a single step simulates only 1 femtosecond of physical time, achieving nanosecond-scale molecular dynamics requires 10^6 steps, corresponding to at least 10^{24} FLOPS. Such enormous computational demands cannot be completed in short timeframes even with the world' s largest supercomputers.

To address the excessive computational cost of pure first-principles calculations, researchers developed multiscale methods, most notably the quantum mechanics/molecular mechanics (QM/MM) method that won the 2013 Nobel Prize in Chemistry. This approach uses high-precision first-principles methods for core physical-chemical reaction regions (e.g., active site atoms of enzymes and their substrates) while employing lower-precision, less computationally intensive classical mechanics for surrounding regions (solvent, proteins, and other areas). This combination of high and low precision effectively reduces computational cost. However, it still faces enormous challenges in real applications. For instance, a single *Mycoplasma genitalium* cell with a radius of approximately 0.2 micrometers contains 3×10^9 atoms and 77,000 protein molecules. Since core computational time comes from the QM portion, simulating a 2-hour process would require 10^9 years. Extending this to human brain simulation would involve 10^{26} atoms, conservatively requiring 10^{10} active sites for QM calculation. Simulating one hour of QM portion would take 10^{24} years, while the MM portion would require 10^{23} years. This situation of excessively long computation time is known as the "curse of dimensionality."

2.2 Simulation Intelligence

As discussed, traditional numerical computation encounters the "curse of dimensionality" in multiscale simulations of the microscopic world, where computational cost increases exponentially with dimensionality. The renowned physicist Dirac[9] noted that the fundamental physical laws required for the mathematical

theory of most physics and all of chemistry are completely known. However, the difficulty lies in the fact that exact solution methods for these laws involve excessively complex equation-solving processes that cannot be completed in finite time. In a 2022 survey report, the Pasteur Laboratory research team[10] introduced the concept of "simulation intelligence"—the fusion of scientific computing and artificial intelligence aimed at studying system evolution through computer simulation to better understand and discover in-situ phenomena. The key challenge here is exploring "in-situ" phenomena. Therefore, compared to the first stage of scientific simulation, simulation intelligence must expand the capabilities of traditional numerical computation and develop innovative techniques (such as AI integration) to address the dimensionality curse in both physical and data modeling. This exploration of "in-situ" phenomena is precisely the key problem that Academician E Weinan' s team advocates for AI for Science (AI4S)[11] to solve.

In Figure 1, simulation intelligence embeds AI computation processes $A(x)$ within traditional numerical computation, expressed as the solver $y = F(f(x), a(x))$. Its distinctive feature is using physical principles and massive data to construct AI models and computational processes. Thus, simulation intelligence embeds AI models (currently primarily deep learning models) within traditional numerical computation. Unlike the "black box" nature of deep learning models in other AI application domains, simulation intelligence requires that the fundamental starting points and basic structures of these models be interpretable[10]. Currently, extensive research exists in this direction, with Zhang et al.[12] providing a systematic review of the latest advances in simulation intelligence in 2023. From understanding the subatomic (wave functions and electron density), atomic (molecules, proteins, materials, and interactions), to macroscopic (fluids, climate, and subsurface) scales, they categorize research objects into three major systems—quantum, atomistic, and continuum—covering seven scientific fields including quantum mechanics, density functional theory, small molecules, proteins, materials science, intermolecular interactions, and continuum mechanics. They also discuss in detail the key common challenge: how to capture physical first principles, particularly symmetries in natural systems, through deep learning methods. Intelligent models based on physical principles have permeated nearly all domains of traditional scientific computing, significantly enhancing simulation capabilities for microscopic multiscale systems and providing more comprehensive support for online experimental feedback loops. For example, rapid real-time iteration between computational simulation systems and robot scientists helps improve research efficiency. Therefore, simulation intelligence also includes the control process of "theory-experiment" iteration to some extent, while involving parts of the broad definition of scientific simulation.

2.3 Science Brain

Traditional scientific methods have fundamentally shaped the step-by-step "guidelines" for human exploration of nature and scientific discovery. When facing new research problems, scientists are trained to think in terms of hypotheses and alternatives, specifying how to conduct controlled tests[13]. While this process has been effective for centuries, it is very slow and, in some sense, subjective—driven by scientists' ingenuity and biases. These biases sometimes hinder necessary paradigm shifts[14]. AI technology has sparked expectations for optimal and innovative solutions through the fusion of science and intelligence.

This stage represents the advanced implementation of the broad definition of computer simulation, dependent on the development of artificial general intelligence. As shown in Figure 1, this advanced implementation is expressed as $y = A(f(x))$, meaning that machines (agents) can perform the process of "creating and discovering" new theories. This paper does not discuss its specific technical features but only describes the macro-level vision. When Kitano[5] proposed the concept of AI scientists, he used two Nobel Prize-winning works—somatic cell reprogramming and the discovery of conductive polymers—as examples. To some extent, the processes of these two scientific discoveries are equivalent to a common research discovery process: "having a clear goal with obvious scientific significance, followed by search and optimization." The discovery of conductive polymers was particularly dramatic and accidental, essentially resulting from a laboratory accident where an intern in Shikawaka' s lab mistakenly used an abnormally high concentration of chemicals to form a film. Shirakawa noticed this accidental discovery and optimized the film formation conditions, eventually determining the conditions for conductive polymer formation together with MacDiarmid and Heeger. Thus, under the gradual maturation of artificial general intelligence, if we can formulate the process of scientific discovery and develop a scalable system to execute this process, we may obtain an entirely new form of scientific discovery—the "science brain." This form would achieve intelligent full automation of planning, speculation, experimentation, validation, and analysis.

The three stages of scientific simulation development described above clearly distinguish the progressive enhancement of computer simulation in terms of computability and intelligence capabilities. (1) The numerical computation stage involves coarse-grained modeling of relatively simple computational problems within complex scientific problems, belonging purely to the narrow definition of computer simulation. It has not only facilitated scientific discoveries at macro scales across numerous fields but also initiated preliminary exploration of the microscopic world. (2) The simulation intelligence stage addresses multiscale exploration of the microscopic world, pushing computational capabilities to a new level. Beyond achieving orders-of-magnitude improvements within the narrow definition of computer simulation, this stage also involves computational acceleration of certain key experimental links, laying the foundation for the next stage of scientific simulation. (3) The science brain stage will realize the broad

definition of computer simulation, where computer simulation will possess the ability to create knowledge.

3. Key Issues in Designing Computing Systems for Simulation Intelligence

Based on our coarse-grained division of scientific simulation development stages, corresponding computing systems have evolved in parallel. Supercomputers played an irreplaceable role in the numerical computation stage. As we advance to the new simulation intelligence stage, the design of underlying computing systems becomes foundational. What design philosophy should guide the development of computing systems for simulation intelligence?

Examining the history of computing and scientific research reveals a fundamental periodic pattern in computing system development: in the early stages of new computing paradigms and demands, system design pursues extreme specialization; after a period of technological evolution and application expansion, the focus shifts toward generality. In the long early stages of technological civilization, computing systems were various specialized mechanical devices assisting simple calculations (Figure 2 [Figure 2: see original paper]). Modern breakthroughs in electronics spawned electronic computers, whose continuously improving computational capabilities have advanced mathematics, physics, and other disciplines, particularly through large-scale numerical simulations on supercomputers that have led numerous frontier scientific research and major engineering applications. Thus, increasingly powerful general-purpose high-performance computers continuously accelerate large-scale applications across macro-scale sciences, achieving significant results. Next, multiscale exploration of the microscopic world will be the core scenario for future Zetta-scale (10^{21}) supercomputing applications. However, existing general-purpose high-performance computer architectures will encounter bottlenecks in power consumption and efficiency, making them unsustainable.

Combined with the new characteristics of the simulation intelligence stage, this paper argues that computing systems for simulation intelligence should aim for extreme specialization in Zetta-scale computing as their design goal. The highest-performance computing systems will be customized specifically for simulation intelligence applications, both in hardware itself and in underlying algorithms and abstractions. Intuitively, computing systems for simulation intelligence cannot be separated from intelligent components (software and hardware). However, can we truly meet simulation intelligence demands by building intelligent computing systems based on existing components? The answer is no. Academician Li Guojie once pointed out: "The current situation in the information field has been jokingly described as: 'Software is eating the world, AI is eating software, deep learning is eating AI, and GPUs are eating deep learning.' Developing higher-performance GPUs or similar hardware accelerators seems to be the main approach to dealing with big data. However, it is unwise to blindly rely on hardware brute force without understanding where acceleration

is needed." Therefore, the key to designing intelligent systems lies in deeply understanding the problems to be solved. The role of computer architects is to select appropriate knowledge representations, identify overhead-intensive tasks, learn meta-knowledge, determine fundamental operations, and then support these tasks with software and hardware optimization techniques.

Designing computing systems for simulation intelligence is a newly emerging research topic with significant uniqueness compared to other computing system designs. Therefore, a unified holistic perspective is needed to advance the intersection of AI and simulation science. In 1989, Wah and Li[15] summarized three levels for intelligent computer system design—a viewpoint that remains relevant today. Unfortunately, no more in-depth discussions or practical research have emerged on this topic. Specifically, intelligent computer system design must consider three levels: representation level, control level, and processor level. The representation level deals with the knowledge and methods for solving given AI problems and how to represent them. The control level concerns algorithmic dependencies and parallelism, as well as program representation of problems. The processor level addresses the hardware and architectural components required to execute algorithms and program representations. Below, we discuss key issues in designing computing systems for simulation intelligence based on these three levels.

3.1 Representation Level

The representation level is a crucial element in the design process, encompassing domain knowledge representation and common feature (meta-knowledge) representation, which determines whether a given problem can be solved within reasonable time. The essence of defining the representation level is to perform high-level abstraction of behaviors and methods adaptable to broad applications, decoupling them from specific implementations. Below are cases for domain knowledge representation and common feature representation.

From the perspective of current AI research for science, symmetry research will become an important breakthrough in representation learning. This is because physical conservation laws result from symmetry (Noether' s theorem), and conservation laws are commonly used to study fundamental particle properties and interactions. Physical symmetry refers to invariance under certain transformations or operations, where no distinguishable measurement can be made (indistinguishability). Small molecule representation models based on multilayer perceptrons (MLP), convolutional neural networks (CNN), and graph neural networks (GNN) have been widely applied to structure prediction of proteins, molecules, crystals, and other materials after effectively incorporating symmetry[12].

In 2004, Colella[16] proposed the "Seven Dwarfs" of scientific computing to DARPA—dense linear algebra, sparse linear algebra, structured grid computations, unstructured grid computations, spectral methods, particle methods, and

Monte Carlo simulation. Each represents a computational method that captures computation and data movement patterns. Inspired by this, Lavin et al.[10] from the Pasteur Laboratory similarly defined nine motifs of simulation intelligence—multiphysics and multiscale modeling, surrogate modeling and simulation, simulation-based inference, causal modeling and inference, agent-based modeling, probabilistic programming, differentiable programming, open-ended optimization, and machine programming. These nine motifs represent complementary computational method types, laying the foundation for synergistic simulation and AI technologies to advance science. The themes identified for traditional scientific computing once provided a clear roadmap for developing numerical methods (and parallel computing) across disciplines; similarly, the themes for simulation intelligence are not limited to narrow performance or program code but instead stimulate innovation in algorithms, programming languages, data structures, and hardware[17].

3.3 Processor Level

From the numerical computation stage to the simulation intelligence stage, a key driver of technological development has been that current hardware technology cannot meet computational demands. Therefore, the primary issue in processor level design is: Will changes in the representation level (such as symmetry and motifs) generate entirely new hardware architectures? Will they be implemented based on traditional ASICs, or will they transcend CMOS technology? From the high-performance computing roadmap, this is also a core issue for future Zetta-scale supercomputer hardware design. We can boldly predict that around 2035, Zetta-scale supercomputing may emerge. Although CMOS platforms will remain mainstream for performance and reliability reasons, some core components will be specialized hardware built on non-CMOS processes.

Although Moore' s Law has slowed, it remains effective. The key challenge is how to approach its limits—in other words, how to fully exploit the potential of CMOS-based hardware through software-hardware co-design. Even in supercomputing, where performance is the highest priority, most algorithmic workloads achieve only a tiny fraction of raw hardware performance. Reviewing the early development of supercomputing, the fundamental design philosophy was software-hardware co-design. Over the next decade, as the rapid development "dividends" of microprocessors are exhausted, hardware architecture for simulation intelligence computing systems should return to ground-up software-hardware co-design. A prominent example is molecular dynamics simulation, as mentioned earlier. The Anton series[20] is a supercomputer family designed from scratch to enable large-scale, long-timescale molecular dynamics simulations—a necessary condition for exploring the microscopic world. However, the latest Anton can only simulate 20 microseconds using classical force field models, unable to perform long-timescale simulations with first-principles accuracy, which is required for most practical applications such as drug design.

Recently, as a typical application of simulation intelligence, breakthroughs in the

DeePMD model on traditional large-scale parallel systems have demonstrated its enormous potential. The supercomputing team at the Institute of Computing Technology, Chinese Academy of Sciences, has achieved nanosecond-scale simulations of first-principles accuracy molecular dynamics for 170 atoms[21]. However, long-timescale simulations require hardware architectures with extreme scalability, necessitating innovations in computational logic and communication operations. This paper believes two technologies can be expected to play key roles: (1) in-memory computing architectures that improve computational efficiency by reducing data movement latency; and (2) silicon photonics interconnect technology that provides high-bandwidth communication capabilities with high energy efficiency, helping to improve parallelism and data scale. Furthermore, with extensive and in-depth research on simulation intelligence applications, we believe that new "floating-point" operation units and instruction sets for scientific simulation will gradually emerge.

4. Conclusion

This paper argues that scientific simulation is currently in the early stage of simulation intelligence, making research on enabling technologies for simulation intelligence crucial. In general scientific research, individual concepts, relationships, and behaviors may be understandable, but their combined behavior leads to unpredictable results. Understanding the dynamic behavior of complex systems is invaluable for researchers tackling challenging domains. In designing computing systems for simulation intelligence, an essential component is interdisciplinary collaboration among domain scientists, mathematicians, computer scientists and engineers, and modeling and simulation practitioners. Such collaboration will build better simulation computing systems and form more comprehensive and holistic approaches to solving complex real-world problems.

References

1. Weaver W. Science and complexity. *American Scientist*, 1948, 36(4): 536-544.
2. Winsberg E. Computer simulations in science// Zalta E N, Nodelman U, eds. *The Stanford Encyclopedia of Philosophy* (Winter 2022 Edition). (2019-09-26) [2024-01-07]. https://plato.stanford.edu/archives/win2022/entries/simulations-science.
3. Lynch S M. *Using Statistics in Social Research: A Concise Approach*. New York: Springer, New York, 2013.
4. Schelling T C. Dynamic models of segregation. *The Journal of Mathematical Sociology*, 1971, 1(2): 143-186.
5. Kitano H. Nobel Turing Challenge: Creating the engine for scientific discovery. *npj Systems Biology and Applications*, 2021, 7: 29.

6. Klahr D, Kotovsky K. *Complex Information Processing: The Impact of Herbert A. Simon*. New Jersey: Lawrence Erlbaum Associates, 1989: 375-398.

7. Petitet A, Whaley R C, Dongarra J, et al. HPL—A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers. (2018-12-02) [2024-01-07]. https://netlib.org/benchmark/hpl/.

8. Dongarra J, Heroux M A. Toward a New Metric for Ranking High Performance Computing Systems. East Ave, Livermore: Sandia National Laboratories, 2013.

9. Dirac P A M. Quantum mechanics of many-electron systems. *Proceedings of the Royal Society of London Series A*, 1929, 123: 714-733.

10. Lavin A, Krakauer D, Zenil H, et al. Simulation intelligence: Towards a new generation of scientific methods. (2021-11-27) [2024-01-07]. https://arxiv.org/abs/2112.03235.

11. Zhang L F, Li X Y, E W N, et al. *Observation and Outlook on the Global Development of AI for Science (AI4S)*. Beijing: AI for Science Institute, Beijing, 2023. (in Chinese)

12. Zhang X, Wang L M, Helwig J, et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. (2023-07-17) [2024-01-07]. http://arxiv.org/abs/2307.08423.

13. Voit E O. Perspective: Dimensions of the scientific method. *PLoS Computational Biology*, 2019, 15(9): e1007279.

14. Kuhn T S, Hawkins D. The structure of scientific revolutions. *American Journal of Physics*, 1962, 31(7): 554-555.

15. Wah B W, Li G J. A survey on the design of multiprocessing systems for artificial intelligence applications. *Transactions on Systems, Man, and Cybernetics*, 1989, 19(4): 667-692.

16. Colella P. Defining Software Requirements for Scientific Computing, 2004, (2012-05-19) [2024-01-07]. https://www.slideserve.com/kiral/software-requirements.

17. Asanović K, Bodík R, Catanzaro B, et al. The landscape of parallel computing research: A view from Berkeley. Technical Report No. UCB/EECS-2006-183. Berkeley: EECS Department, University of California, 2006.

18. Zhang L F, Han J Q, Wang H, et al. Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics. *Physical Review Letters*, 2018, 120(14): 143001.

19. Zhang L F, Han J Q, Wang H, et al. End-to-end symmetry preserving interatomic potential energy model for finite and extended systems. *Advances in Neural Information Processing Systems*, 2018, 31: 4436-4446.

20. Shaw D E, Adams P J, Azaria A, et al. Anton 3: Twenty microseconds of molecular dynamics simulation before lunch// SC21: International Conference for High Performance Computing, Networking, Storage and Analysis. St. Louis: IEEE, 2021: 1-11.

21. Jia W, Wang H, Chen M, et al. Pushing the limit of molecular dynamics with ab initio accuracy to 10 billion atoms// Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. Seoul: ACM, 2022: 205-218.

Tan Guangming is a professor at the Institute of Computing Technology, Chinese Academy of Sciences. His primary research direction is high-performance computing. E-mail: tgm@ict.ac.cn

Sun Ninghui is an academician of the Chinese Academy of Engineering, a professor at the Institute of Computing Technology, Chinese Academy of Sciences, and an editor of *Bulletin of Chinese Academy of Sciences*. His primary research direction is high-performance computing. E-mail: snh@ict.ac.cn

Responsible Editor: Yue Lingsheng

Corresponding author

FLOPS: floating-point operations per second; EFLOPS: exaflops, 10^{18} floating-point operations per second.

Note: Figure translations are in progress. See original paper for figures.