EDITORS: Mary Ann Leung, mleung@shininstitute.org
Jay Lofstead, gflofst@sandia.gov

DEPARTMENT: COMMUNITY STEWARDSHIP

Ecosystems for Scientific Computing in the Age of Artificial Intelligence

Lois Curfman McInnes and Anshu Dubey, *Argonne National Laboratory, Lemont, IL, 60439, USA*
Dorian Arnold, *Emory University, Atlanta, GA, 30322, USA*
Prasanna Balaprakash, *PrimaLabs, San Francisco, CA, 94127, USA*
Mike Bernhardt, *Team Libra, Kennewick, WA, 99338, USA*
Beth Cerny, *Argonne National Laboratory, Lemont, IL, 60439, USA*
Denice Ward Hood, *University of Illinois Urbana-Champaign, Champaign, IL, 61821, USA*
Mary Ann Leung, *Sustainable Horizons Institute, Rancho Mirage, CA, 92270, USA*
Olivia B. Newton, *University of Montana, Missoula, MT, 59812, USA*
Stefan M. Wild, *Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA*

Scientific computing is at an inflection point. Artificial intelligence (AI) is reshaping how scientific software is developed, how teams collaborate, how projects are governed, and how the next generation is trained. Drawing on insights from a 2025 workshop report, this article argues that the future of discovery will depend on agile, robust ecosystems built through socio-technical co-design—the intentional integration of technical and human systems. This perspective is essential for ensuring that future scientific computing remains trustworthy, sustainable, and scalable. It combines advances in AI, high-performance computing, and software with new models for cross-disciplinary collaboration, education, and workforce development. Key recommendations include building modular, trustworthy AI-enabled software ecosystems; enabling teams to integrate AI into scientific workflows while preserving human creativity, integrity, and rigor; and developing adaptive training pathways that keep pace with rapid technological change. By sharing these perspectives, we hope to stimulate broader community dialogue and encourage coordinated action.

For decades, the high-performance computing (HPC) community has driven scientific innovation, establishing computation as central pillar of discovery alongside theory and experiment. Scientific software lies at the heart of this progress, encoding domain knowledge, computational methods, and collaborative problem solving in forms that scale across researchers and institutions. Today, however, the rapid emergence of artificial intelligence (AI) is fundamentally reshaping this landscape.[1,2] AI is altering how software is written, how teams interact, and how new practitioners are trained. A gap is emerging between traditional computational science approaches and the skills and organizational models required for the future.

Digital Object Identifier 10.1109/MCSE.2026.3685068
Date of current version 17 June 2026.

A TRANSFORMATIVE MOMENT

To explore these shifts, a 2025 workshop convened experts across HPC, AI, computational science, software engineering, and the social sciences. Participants articulated a shared vision: Future scientific progress will depend on integrated ecosystems in which hardware, software, AI, and human expertise coevolve.[3] Building on that report, this article offers a synthesis and interpretation of its findings, with emphasis on implications for the broader scientific computing community. We highlight cross-cutting themes, identify emerging gaps, and frame a set of challenges that call for coordinated community action.

TOWARD ECOSYSTEM PERSPECTIVES FOR NEXT-GENERATION SCIENCE

Advanced simulations (spanning domains, such as materials science, astrophysics, nuclear energy, biology, engine design, weather prediction, and batteries) require integrated expertise across many disciplines. These include domain sciences, applied mathematics, computer science, and AI, as shown in Figure 1. No single team can develop or maintain all of these components. Instead, progress depends on distributed, cross-disciplinary collaboration, in which reusable libraries and shared tools encapsulate expertise and enable wide application across domains.

From Codes to Ecosystems

This shift is evident in modern computational science, which increasingly relies on integrated workflows spanning modeling, simulation, analysis, and AI-driven inference. These workflows depend on layered software stacks that combine programming models, math libraries, tools for visualization and performance, and emerging AI frameworks.

We adopt a broad view of scientific computing ecosystems as dynamic socio-technical systems composed of people, software, infrastructure, institutions, workflows, and cultural practices, as illustrated in Figure 2. Advancing the outer elements of Figure 2 (software, collaboration, and pedagogy) requires intentional co-design supported by strong interactions among stakeholders in the inner ring (including researchers, software developers, and computing facilities), all while developing workforce and community.

Ecosystems are inherently context-dependent, shaped by stakeholder interactions, and effective only when technical and social components are aligned. This perspective shifts the focus from individual codes or tools to interconnected systems of capability. Rather than centralized control, successful ecosystems emerge through cooperative, distributed efforts that support rapid evolution and leverage specialized expertise.

This growing interdependence raises a central challenge: How to effectively integrate diverse expertise, methods, and requirements across teams. Addressing this challenge requires both improved

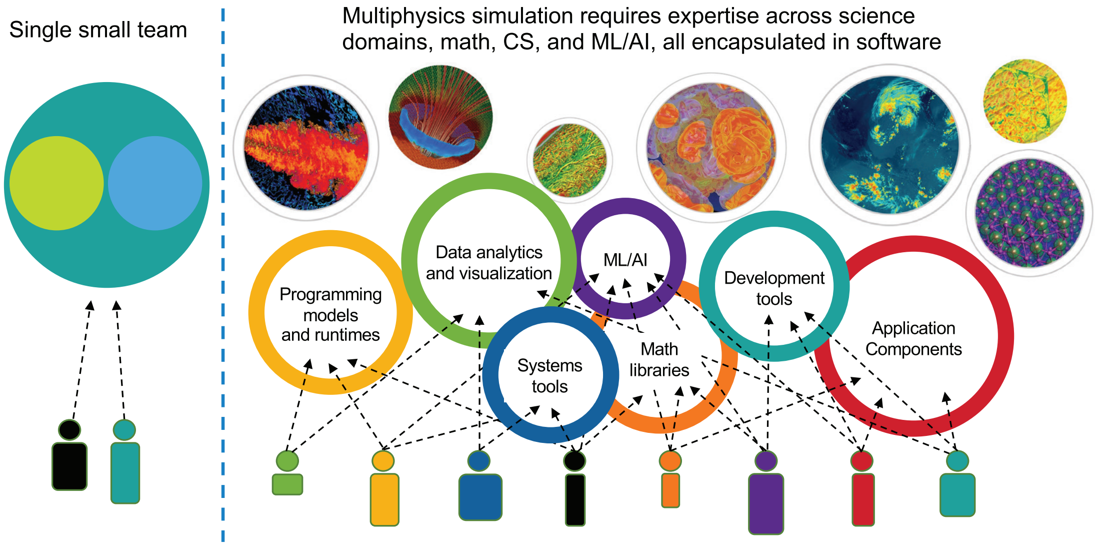

FIGURE 1. Modern scientific computing relies on reusable libraries, tools, and frameworks—developed by distributed teams—to provide shared capabilities across diverse applications. From McInnes et al.,[3] with permission.

software architectures and new approaches to coordination and collaboration.

A NEW LENS: SOCIO-TECHNICAL CO-DESIGN

To tackle these urgent and complex intertwined challenges, the workshop introduced *socio-technical co-design* as a guiding framework (Figure 3). The concept of co-design is widely used in many fields, referring generally to collaborative approaches to designing solutions by involving multiple stakeholders (especially those who will be using or are impacted by the outcomes) throughout the design process. Co-design in HPC and AI has been critical to the design and implementation of contemporary computer architectures, applications, algorithms, and software, considering their interrelationships (as shown in the left-hand side of Figure 3).[4,5] While still essential, this approach is no longer sufficient. Modern scientific computing must also account for AI upending social and pedagogical aspects of the scientific endeavor, such as institutions, training, and collaboration.

Socio-technical co-design extends the HPC co-design paradigm by treating technical and social elements as interdependent and coevolving. Rather than designing architectures, applications, algorithms, and software first and addressing factors related to teams, pedagogy, community, and workforce later, this approach integrates all from the outset, including human–AI collaboration. The result is more resilient and adaptable ecosystems for scientific computing that can respond to rapid technological change while maintaining scientific integrity.

AI-DRIVEN SOFTWARE ECOSYSTEMS FOR SCIENTIFIC COMPUTING

AI is transforming scientific software ecosystems by introducing new capabilities and new complexities. AI-enabled software ecosystems for scientific computing integrate AI inference engines with simulation and analysis workflows, using AI agents to enhance researcher productivity. These inference engines rely upon data from simulations, observations, and archives. Simulation and analysis capabilities in turn depend upon middle layers of the software stack, including math and visualization libraries, workflow tools, and performance optimization capabilities. Such ecosystems must operate across rapidly evolving AI models, data pipelines, and heterogeneous computing environments (e.g., CPUs, GPUs, and quantum systems).

Design Challenges in Dynamic Environments

A central challenge is determining how to design, develop, and sustain software for next-generation science in fast-evolving heterogeneous environments,

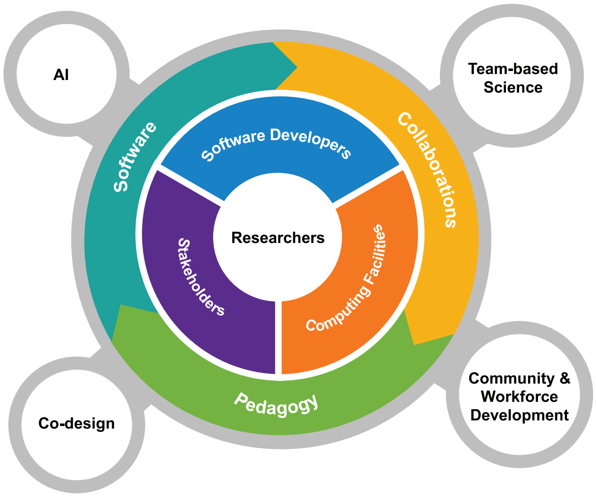

FIGURE 2. A view of next-generation scientific computing ecosystems, with researchers at the core. From McInnes et al.,[3] with permission.

FIGURE 3. Socio-technical co-design for next-generation scientific computing extends traditional HPC co-design by integrating technical elements with social dimensions such as team science, human–AI collaboration, pedagogy, and development of the workforce. Adapted from McInnes et al.,[3] with permission.

while simultaneously building the human enterprise and collaboration protocols. Prior studies, including the National Science Foundation/Department of Energy National Artificial Intelligence Research Resource software report[6] and broader software engineering analyses,[7] highlight the scope of these challenges. Addressing them will require a strategic blend of technical innovation and community-driven co-design.

Trust, Validation, and AI-Generated Code

A critical challenge lies between current AI capabilities and the stringent requirements of scientific computing for reliability and replicability.[8] Unlike general-purpose software, scientific applications must preserve physical principles, maintain numerical stability, and produce consistent results across platforms. AI-assisted ("vibe") coding remains challenging for scientific software because natural language specification is by its nature imprecise. Traditionally, developers rely on iterative verification and debugging to resolve such ambiguity. How AI can accelerate or safely replace this iterative convergence to trustworthy software remains an open question. Additional challenges arise from federated learning, including privacy-preserving collaboration and vulnerabilities such as adversarial attacks and data poisoning. Additionally, human-in-the-loop and continuous learning mechanisms are critical for maintaining alignment between AI-generated software and evolving scientific requirements.

Integration, Reuse, and Scientific Machine Learning (SciML) Tradeoffs

Developing entirely new scientific software ecosystems is impractical, yet integrating AI into existing software presents its own challenges. The changes required to code will be deeply invasive, making intrusion-aware design essential to minimize disruption. At the same time, the field continues to debate when traditional numerical methods should be used, when SciML[9] methods may be preferable, and how the two can be combined effectively. Advancing this integration will require robust, widely accepted metrics for evaluating SciML against established numerical approaches.

TEAMS, TRUST, AND AI AS A COLLABORATOR

Evolving Teams in Scientific Computing

As software ecosystems evolve, so too must the teams that develop and use them. Scientific collaboration increasingly involves cross-disciplinary teams with diverse expertise, vocabularies, and goals.[10] These differences introduce challenges in communication, coordination, and decision making. The growing role of AI further complicates this landscape, raising important questions about how AI systems should participate, as tools, collaborators, or something in between.

Trust in Teams, Software, and AI

Trust is a central issue across all of these dimensions. Effective collaboration requires trust not only among team members but also confidence in software systems and AI-generated outputs. Building and maintaining this trust requires transparency, shared understanding, and sustained interaction.

Institutional Barriers and Incentives

In addition, existing norms and institutional structures often hinder effective collaboration. Funding models, recognition systems, and career incentives may undervalue critical contributions, such as software maintenance, documentation, and community building. Addressing these misalignments is essential for sustaining the scientific computing ecosystems on which modern research depends.

WORKFORCE AND PEDAGOGY IN THE AI ERA

Challenges in Cross-Disciplinary Training

Pedagogy and development of the workforce in the computing sciences have long been challenged by the need to integrate concepts across multiple disciplines and by the constraints of traditional academic departmental structures, which often limit cross-disciplinary training. Mission-driven research laboratories have historically provided environments where such integration can occur. With the rise of AI and expansion of computational science across virtually all aspects of science and society, both the opportunity and urgency to embed cross-disciplinary training have increased.

Uncertainty in Workforce Preparation

Perhaps the most profound challenge is the uncertainty surrounding how to prepare the future workforce for the disruptive changes underway in computing for science. As AI tools become more capable, there is growing concern about producing scientists who can use AI-generated code but lack the fundamental understanding needed to validate, debug, or extend it.

This situation creates a paradox: AI tools designed to democratize software development may inadvertently create new barriers to deep scientific understanding.

Curricula, Skills, and New Learning Models

At the same time, the rapid pace of AI advancement makes it difficult for educational programs to update curricula as technologies and practices change. The shift toward AI-assisted development also raises fundamental questions about what constitutes essential knowledge. Building curricula and the workforce for rapidly changing technological ecosystems requires new instructional and workforce development paradigms, including greater reliance on apprenticeship-based approaches[11] and collaborative learning environments supported by communities of practice.

A ROADMAP GROUNDED IN COMMUNITY ACTION

While technical research is a key driver of innovation, progress in scientific computing also depends on community-driven action. Building trustworthy, capable, and sustainable software ecosystems requires more than technical breakthroughs. It calls for strategic coordination, shared infrastructure, supportive policies, and workforce investment. Various organizations and foundations—such as the Consortium for the Advancement of Scientific Software,[a] High Performance Software Foundation,[b] NumFOCUS foundation,[c] Research Software Alliance,[d] Software Sustainability Institute,[e] Research Software Engineer organizations,[f] —as well as a variety of projects are already advancing cultural and structural change,[12] but significant gaps remain in infrastructure, recognition, and career pathways. Raising awareness of these efforts and contributing to them, whether directly or through complementary action, will benefit the broader scientific community.

Strategic Public–Private Partnerships

Strategic public–private partnerships must go beyond the current model of short-term research goals to support co-designed solutions that align with long-term scientific needs and technological shifts. The private sector includes both commercial developers of computing technologies—such as hardware manufacturers, cloud service providers, and AI platform companies—and major industrial users of scientific software. Sustained coordination is needed to shape not only the technical requirements of software tools but also the practices and training ecosystems that support their adoption. Joint efforts might include cross-sector learning communities, affinity groups, shared training initiatives, and advisory partnerships that bring together HPC practitioners, AI developers, and domain scientists. Of particular importance is the need to align the rapidly evolving commercial AI and hardware landscapes with the long-term needs of the scientific community. These shared efforts play a vital role in maintaining competitiveness, advancing research, and ensuring that science, engineering, and manufacturing progress together for long-term socioeconomic resilience.

Community Engagement and Broad-Access Infrastructure

Scaling scientific software requires engagement models that ensure trust, transparency, and shared ownership, supported by open governance and broad participation. Equally important is investing in infrastructure that lowers barriers for under-resourced communities, such as accessible cloud resources and mentoring networks.

Recognition, Incentives, and Long-Term Community Investment

The sustainability of scientific software ecosystems depends on valuing and supporting the full range of essential contributions, including development, documentation, user support, training, maintenance, and community stewardship. These roles must be reflected in hiring, promotion, and funding decisions and reinforced through long-term investment strategies. Stable, visible support for these roles strengthens the resilience of the overall ecosystem and builds the capacity needed for lasting progress.

Responsible Innovation Guidelines and Fair AI

Advancing AI-driven scientific discovery without oversight regarding fairness risks opening a Pandora's box of unintended consequences. Community-wide guidelines for the responsible integration of AI are essential and must be treated as a top priority. These guidelines should address algorithmic transparency, data governance, and reproducibility, while also

[a] https://cass.community
[b] https://hpsf.io
[c] https://numfocus.org
[d] https://www.researchsoft.org
[e] https://www.software.ac.uk
[f] https://researchsoftware.org

establishing norms to prevent bias, protect users, and anticipate unintended consequences. A key part of this effort is ensuring that datasets are complete and representative of the full range of individuals and perspectives they aim to serve. Responsible innovation must remain a core design principle as both tools and teams evolve.

CONCLUSION

These policy and collaborative efforts are not secondary to technological innovation, they are foundational. As AI-generated code, agentic systems, and complex workflows reshape scientific computing, maintaining correctness, trust, and accountability can no longer rely on traditional assumptions or processes. Meeting these demands requires strategic partnerships, reinforced by community-driven practices, institutional reform, and sustained investment across research and the public and private sectors. This integrated approach supports ecosystems in which technical and social components must continuously coevolve to address emerging challenges in validation, human–AI collaboration, and workforce development. We present these perspectives to inform ongoing efforts and to encourage continued coordination across the community.

ACKNOWLEDGMENTS

The workshop that led to the insights described in this article was partially supported by the U.S. Department of Energy (DOE) Office of Science Distinguished Scientist Fellows Program. We especially thank our DOE contacts: Hal Finkel and David Rabson, DOE Office of Advanced Scientific Computing Research (ASCR). We sincerely thank all workshop participants for their contributions to workshop insights as summarized in the full workshop report[3] and this article.

REFERENCES

1. M. A. Heroux et al., "Basic research needs in the science of scientific software development and use: Investment in software is investment in science," *Rep. DOE Adv. Sci. Comput. Res. Workshop*, 2023. [Online]. Available: https://www.osti.gov/biblio/1846009
2. J. Carter et al., "Advanced research directions on AI for science, energy, and security: Report on summer 2022 Workshops," 2023. [Online]. Available: https://www.osti.gov/biblio/1986455
3. L. C. McInnes et al., "Report of the 2025 workshop on next-generation ecosystems for scientific computing: Harnessing community, software, and AI for cross-disciplinary team science," 2025, *arXiv:2510.03413*.
4. J. Ang et al., "Reimagining codesign for advanced scientific computing: Report for the ASCR workshop on reimagining codesign," US DOE Office of Sci., Washington, DC, USA, 2022. [Online]. Available: https://www.osti.gov/biblio/1822199
5. T. C. Germann, "Co-design in the exascale computing project," *Int. J. High Perform. Comput. Appl.*, vol. 35, no. 6, pp. 503–507, 2021, doi: 10.1177/10943420211059380.
6. M. E. Papka et al., "Final report of the 2024 NSF/DOE workshop on NAIRR software," US National Science Foundation Rep., 2025. [Online]. Available: https://events.cels.anl.gov/event/529
7. A. Carleton et al., "Architecting the future of software engineering: A national agenda for software engineering research & development," 2021. [Online]. Available: https://insights.sei.cmu.edu/library/architecting-the-future-of-software-engineering-a-national-agenda-for-software-engineering-research-development
8. National Academies of Sciences, Engineering, and Medicine, *Reproducibility and Replicability in Science*. Washington, DC, USA: The National Academies Press, 2019, doi: 10.17226/25303.
9. N. Baker et al., "Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence," U.S. Dept. of Energy, Washington, DC, USA, 2019. [Online]. Available: https://www.osti.gov/biblio/1478744
10. National Academies of Sciences, Engineering, and Medicine, *The Science and Practice of Team Science*. Washington, DC, USA: The National Academies Press, 2025, doi: 10.17226/29043.
11. M. A. Leung et al., 2026, "Iterative Co-design, Co-development and Co-delivery: Accelerating American S&T Productivity Through Public-Private Partnerships," Figshare, doi: 10.6084/m9.figshare.31564060.
12. D. S. Katz et al., "Community organizations: Changing the culture in which research software is developed and sustained," *Comput. Sci. Eng.*, vol. 21, no. 2, pp. 8–24, Mar./Apr. 2019, doi: 10.1109/MCSE.2018.2883051.

LOIS CURFMAN MCINNES is a senior computational scientist at Argonne National Laboratory, Lemont, IL, 60439, USA. Contact her at mcinnes@anl.gov.

ANSHU DUBEY is a senior computational scientist at Argonne National Laboratory, Lemont, IL, 60439, USA. Contact her at adubey@anl.gov.

DORIAN ARNOLD is an associate professor at Emory University, Atlanta, GA, 30322, USA. Contact him at dorian.arnold@emory.edu.

PRASANNA BALAPRAKASH is the cofounder and CEO of PrimaLabs, San Francisco, CA, 94127, USA. Contact him at pbalapra@primalabs.ai.

MIKE BERNHARDT is founder and chief strategy officer at Team Libra, Kennewick, WA, 99338, USA. Contact him at mike@teamlibra.com.

BETH CERNY is head of communications for the Argonne Leadership Computing Facility, Argonne National Laboratory, Lemont, IL, 60439, USA. Contact her at bcerny@anl.gov.

DENICE WARD HOOD is a teaching professor at the University of Illinois Urbana-Champaign, Champaign, IL, 61821, USA. Contact her at dwhood@illinois.edu.

MARY ANN LEUNG is founder and president of Sustainable Horizons Institute, Rancho Mirage, CA, 92270, USA. Contact her at mleung@shinstitute.org.

OLIVIA B. NEWTON is a member of the faculty at the University of Montana, Missoula, MT, 59812, USA. Contact her at olivia.newton@umt.edu.

STEFAN M. WILD leads the Applied Mathematics and Computational Research Division at Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA. Contact him at wild@lbl.gov.
