

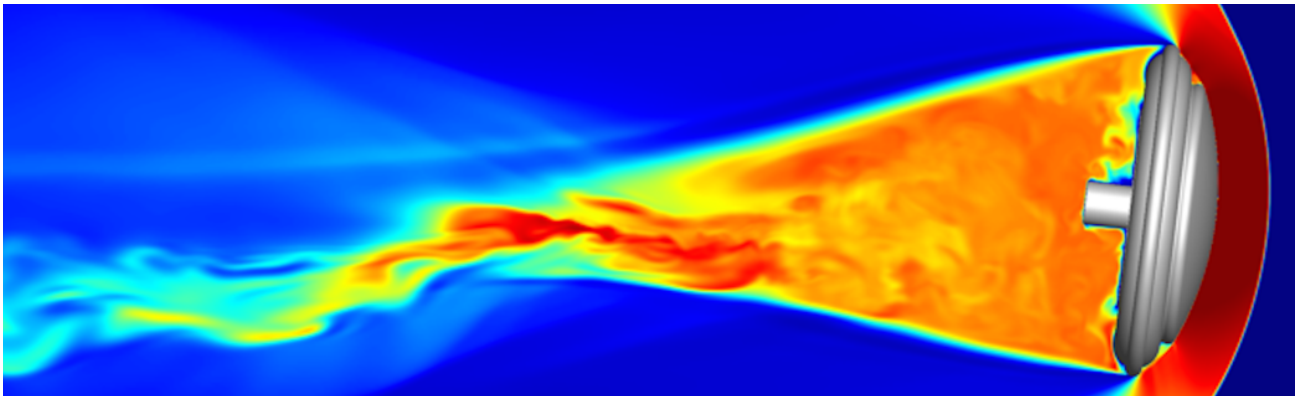
Artificial Intelligence in Mechanical Engineering

It is not only a computer science story

ARVIND PATTAMATTA ■ AUGUST 2026

When people hear “artificial intelligence,” they usually picture a computer scientist training a chatbot or writing software. A mechanical engineer designing a pump, studying a crack, measuring a flame or building a heat exchanger does not fit that picture. That view is already out of date.

Mechanical engineering has always joined mathematics to the physical world. We write equations for motion, heat, fluid flow and stress; turn them into numerical methods; build experiments; and decide whether the answer makes physical sense. AI is beginning to help at every one of those steps. It is not replacing mechanics with computer science. It is becoming another way of doing mechanics.

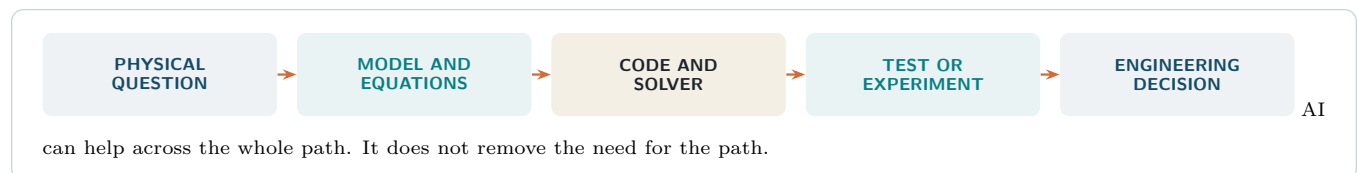


A computed temperature field around a free-flying entry-vehicle model. NASA compared the simulation with measurements from a ballistic-range test—a useful reminder that a striking image is not enough; a model has to meet reality. Image credit: Joe Brock, NASA/Ames. NASA media may generally be used for informational purposes with acknowledgement.[1, 2]

Two rather different things are called “AI”

The first is the AI that mechanical engineers already know. Machine learning finds patterns in data. Neural networks can estimate drag, identify a damaged bearing, classify defects in an infrared image or act as a fast stand-in for a costly simulation. Optimization algorithms search through shapes, materials and operating conditions. Physics-informed neural networks (PINNs) and neural operators try to bring data and governing equations together. These are numerical tools, and they are normally built for a particular task.

The second is newer: large language models (LLMs) and coding agents. They work through ordinary language, equations, software and documents. An engineer can ask for a first derivation, a MATLAB script, an OpenFOAM case, a Python image-processing pipeline, a parameter study or an explanation of why a calculation failed. A coding agent can inspect several files, edit them, run tests and read the resulting errors before trying again. The important change is not that the machine “knows engineering” in the human sense. It is that much of the translation between an idea and a working computational experiment can now be assisted.



This distinction matters. A neural surrogate can be fast but narrow. An LLM can be broad but unreliable. A solver may be rigorous but only for the equations and boundary conditions it was given. The useful future is not one tool replacing all the others. It is a well-checked combination of physics, computation, experiment and AI.

Where AI already fits in mechanical engineering

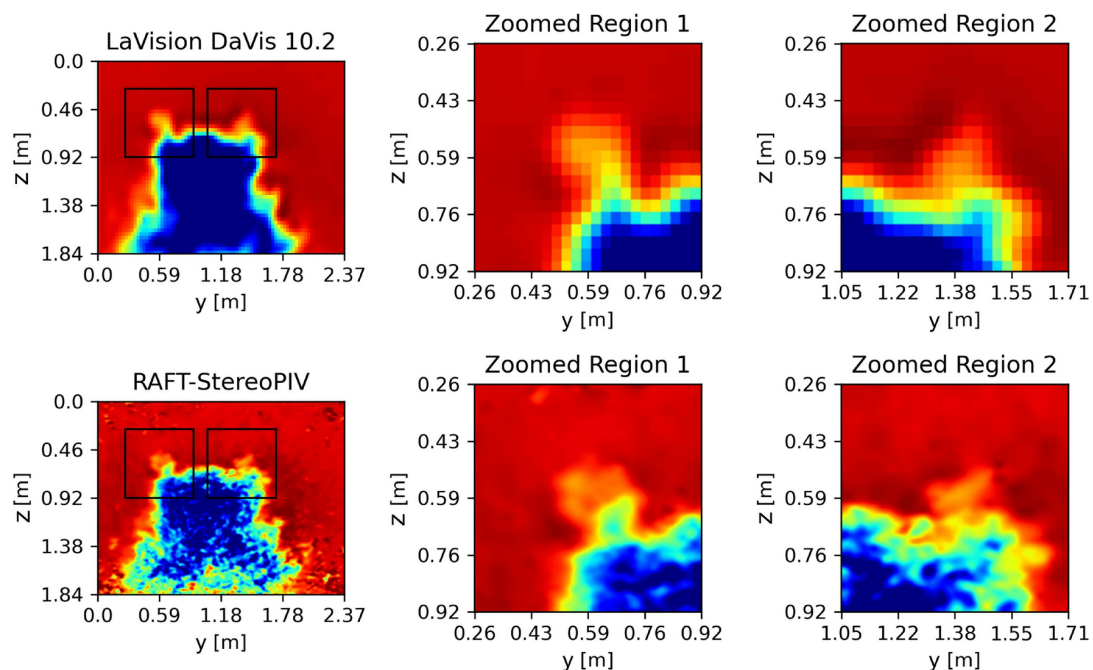
Design, solids and machines

In design, AI can rank many candidate geometries before the expensive calculations begin. A surrogate model can learn the relation between shape and stress, fatigue life or vibration response; an optimizer can then explore thousands of combinations. In machine health, algorithms can look for a small change in vibration, sound or temperature that suggests wear. The engineer still has to ask whether the training data cover the actual material, load, speed and failure mode.

Fluids and thermal sciences

Computational fluid dynamics (CFD) and heat-transfer calculations can take hours or days when a fine mesh, turbulence or many design cases are involved. A trained neural operator or another surrogate may produce an approximate field very quickly. The original Fourier Neural Operator paper reported speed-ups of up to three orders of magnitude on selected PDE benchmarks after training. That is impressive, but it is not a universal replacement for CFD: the training cost, the range of cases and the error outside that range still matter.[3]

PINNs are useful when equations and limited measurements need to be combined—for example, when estimating an unknown heat-transfer coefficient from sparse temperature data.[4] They are often described as alternatives to conventional solvers, but that description is too simple. PINNs can be hard to train and have failed even on apparently simple problems.[5] In practice they are best seen as complements, inverse-model tools or validated surrogates, not as a button that makes CFD or finite-element analysis disappear.



The wake of a moving car measured by stereoscopic particle-image velocimetry (PIV). The conventional cross-correlation result is shown above; the dense deep-learning result is below. In this experiment, the learned method retained the image resolution while the comparison field was about 16 times coarser. The authors also warn that fine-looking structures can include noise or artefacts. Reproduced from Elrefaie et al. (2024), Fig. 11, CC BY 4.0.[6]

Experiments: turning images into measurements

High-speed cameras, laser diagnostics and infrared cameras produce a great deal of raw data. AI can segment a spray, follow particles, identify a crack or reconstruct a velocity field from pairs of images. In the PIV study above, the selected deep model cut error by 68% on a validation set and 47% on an unseen synthetic test set; on the car experiment it produced a much denser field.[6] Machine learning is also being used in thermographic non-destructive testing, although recent reviews still identify calibration, limited datasets and poor transfer to new conditions as major concerns.[7]

An LLM adds a practical layer around those methods. It can draft code to read camera files, apply calibration, call the vision model, calculate averages, plot uncertainty and write a processing note. That can save a researcher from spending days on file formats and plotting details. It cannot decide whether an unexpected hot spot is physics, a reflection, a bad emissivity value or a faulty sensor.

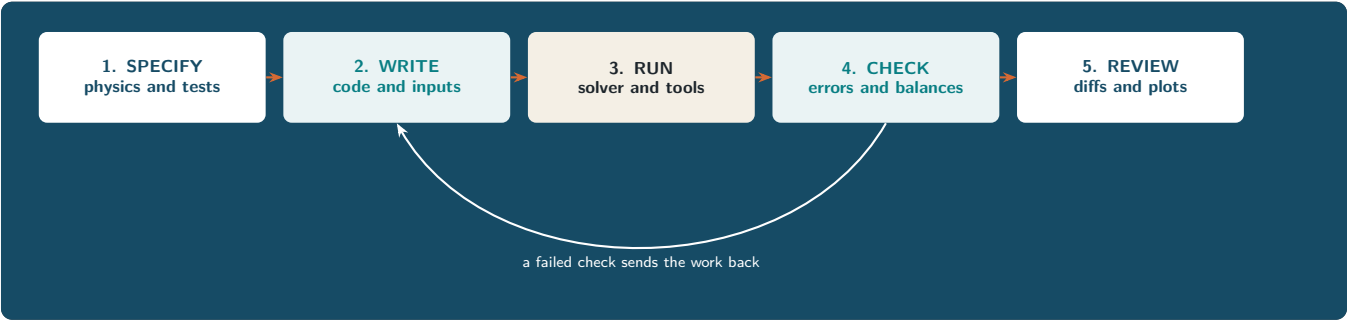
THE SIMPLE TEST

Before trusting any AI result, ask three questions: Was it checked on a case it had not seen? Does it preserve the physical quantity we care about? Can we trace the answer back to the raw data, calibration, code and model version?

The newer change: an engineering co-pilot

The most immediate use of an LLM is not predicting a flow field. It is helping an engineer build the calculation that produces and checks that field. Consider a simple request: “Write a two-dimensional transient heat-conduction solver for a composite plate. Use SI units, state the assumptions, add a manufactured-solution test, compare two time steps and plot the energy balance.” A useful agent can turn that request into code, run it, notice a failed test and revise the implementation.

That is more helpful than autocomplete, but it is not independent engineering. The prompt may omit contact resistance. The generated finite-difference scheme may be unstable. The energy plot may look smooth while the boundary condition is wrong. The agent supplies speed; the engineer supplies the problem definition and the standard of proof.



Working with the software engineers already use

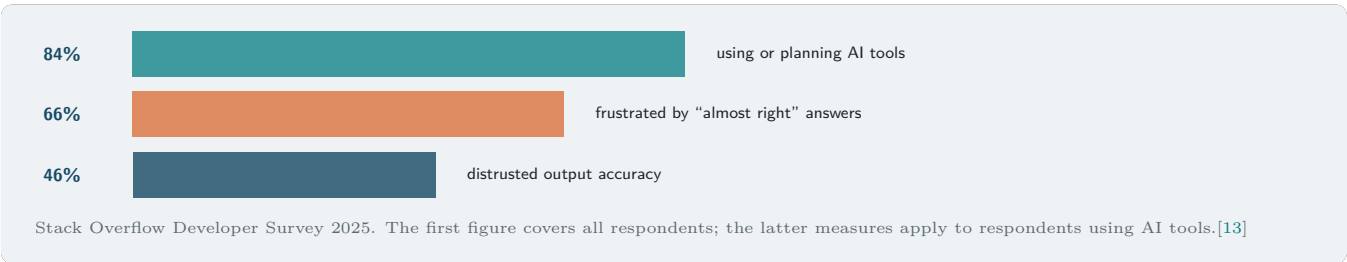
The agent does not need to replace MATLAB, Ansys or OpenFOAM. It can work around them through scripts and documented interfaces.

Environment	What an AI coding tool can prepare	What still gives the answer
MATLAB	Functions, Live Scripts, optimization, data import and plotting; Python can call MATLAB through its official Engine API.[8]	MATLAB’s numerical routines, together with the equations, data and tests chosen by the engineer.
Ansys	APDL or Python automation through the PyAnsys ecosystem; result extraction and report scripts.[9]	The licensed Ansys solver, its element formulations, material models, mesh and settings.
OpenFOAM	Case dictionaries, meshing scripts, function objects, parameter sweeps and C++ extensions.[10]	OpenFOAM’s numerical methods and the physical models selected for the case.
Own solver	A first Python prototype, tests, documentation and later translation to C++ or GPU code.	The implemented algorithm, convergence study, reference cases and experimental validation.

OpenFOAM is a good illustration. A 2025 research prototype called OpenFOAMGPT used retrieval from documentation and iterative correction to set up several CFD tasks. It could be useful, but its authors explicitly kept a person in the loop and reported that performance could vary.[11] This is the right mental model: the AI operates the workflow; it is not the flow solver.

Does it actually save time? Sometimes—and context matters

In three randomized field experiments involving 4,867 software developers, access to an AI coding assistant increased completed tasks by 26.08% in the combined analysis, with larger gains for less-experienced developers. The study measured completed development tasks, not the physical correctness of engineering software.[12] In Stack Overflow’s 2025 survey, 84% of respondents were using or planning to use AI tools, yet 46% distrusted the accuracy of AI output, compared with 33% who trusted it. Among users, 66% complained about answers that were “almost right,” and 45% said debugging AI-generated code could take longer.[13]



A sensible division of work Let AI handle boilerplate, interfaces, plots, documentation and first drafts. Let the engineer own physics, assumptions, acceptance criteria, validation, safety and final sign-off.

Teaching and research: perhaps the biggest opportunity

Mechanical-engineering education is already moving in this direction. At IIT Madras, the four-year B.Tech. in Computational Engineering and Mechanics combines machine learning, data science and high-performance computing with solid and fluid mechanics, dynamics and materials.[14] The Mechanical Engineering department's M.Tech. in Computational Engineering for Mechanical Systems lists *Scientific Computing using Python* and *Data Science for Mechanical Engineers* alongside finite-element analysis, finite-volume CFD and simulation laboratories.[15] The department also has a Mechanics, Modeling and Machine Learning laboratory working on scientific machine learning for structural and fluid-mechanics problems.[16]

That mix is important. Students should not learn AI instead of mechanics. They should learn how AI fits around mechanics. In a heat-transfer class, for example, a student can first predict the trend by hand, ask an LLM to generate a small solver, compare it with a limiting case, and then explain why the numerical answer should be believed. In a laboratory, students can use AI to prepare image-processing code or presentation slides, but they should submit the raw data, calibration, prompts or generated code, and their own checks.

A CLASSROOM EXERCISE THAT STILL TEACHES ENGINEERING

1. Predict the trend, units and limiting behaviour before opening the AI tool.
2. State the governing equations, assumptions and tolerance in the prompt.
3. Run the generated code; do not submit text that has not been executed.
4. Verify it with a hand calculation, benchmark, grid study or experiment.
5. Explain the result orally and disclose what the AI contributed.

A 2026 global higher-education survey collected 45,398 responses across 35 countries. Only 29% of students felt their instructors were well equipped to guide AI use, and only 28% said that most or many assessments reflected the skills needed in an AI-enabled workplace.[17] The answer is not to prohibit the tools or to accept everything they produce. It is to teach verification, disclosure and responsible use as part of the subject.

Can AI fill a shortage of PhD researchers? It can fill part of the gap

India's research workforce is stretched. The Department of Science and Technology cited 253 full-time-equivalent researchers per million people in 2018.[18] The Economic Survey 2025–26 reports gross expenditure on research and development at 0.64% of GDP, below the levels it lists for the United States, China and South Korea.[19] These figures do not prove that every department lacks PhD applicants, but they do show why research time and skilled manpower are precious.

253 researchers per million

India, full-time equivalent, 2018; cited by DST.[18]

0.64% of GDP spent on R&D

Gross expenditure on R&D; Economic Survey 2025–26.[19]

Where a laboratory finds it difficult to recruit doctoral students who are already trained in advanced computation or experiments, AI can reduce the long starting period. It can search and sort papers, explain unfamiliar methods, clean data, write first-pass code, translate MATLAB to Python or C++, prepare solver cases, suggest test problems, produce plots and draft documentation. A new student can reach a useful computational baseline sooner; a supervisor can explore more ideas with the same team.

But a PhD is not a collection of routine tasks. It is training in how to ask a worthwhile question, notice when an experiment is misleading, invent a model, live with uncertainty and defend a genuinely new claim. An AI system does not take responsibility for any of that. It may also fabricate a paper, hide a coding error or offer a familiar answer when the problem needs an original one.

AI can substitute for parts of the workload

literature triage; boilerplate code; data cleaning; plotting; file conversion; routine parameter sweeps; first-pass documentation.

AI should complement the researcher

derivations; solver development; image analysis; experiment planning; interpretation; writing and teaching.

The researcher must still own

the question, new physical insight, experimental judgment, validation, research ethics, authorship and the final claim.

For a lab with limited doctoral manpower, this is still a major gain. AI can substitute for some labour and complement a PhD scholar. It cannot substitute for the scholar as a whole.

What changes—and what does not

AI changes the cost of trying an idea. A small research group can build a prototype solver, automate a parameter study or process thousands of images with less programming overhead. An experienced engineer can move between MATLAB, Python, C++ and commercial tools without remembering every command. Students can ask for another explanation at any hour. These are real benefits.

The basic rules of engineering do not change. Units must agree. Mass, momentum and energy must balance. A mesh must be refined. A time step must be checked. A material law must match the material and loading range. A camera must be calibrated. A surprising result must survive an independent test. Fluent code and attractive plots make these checks more important, not less.

A practical verification ladder

1 Units, signs and order of magnitude

2 Limiting cases and a hand calculation

3 Residuals, conservation and solver warnings

4 Grid, time-step and parameter sensitivity

5 Benchmark, manufactured solution or experiment

6 Independent review, provenance and sign-off

There are also questions of privacy and ownership. Unpublished geometry, student records, industrial data and laboratory images should not be placed in an unapproved service. The model and software version, important prompts, generated files and human changes should be recorded. UNESCO's guidance for generative AI in education and research similarly calls for human-centred use, protection of data and careful validation.^[20]

The long-term skill, then, is not simply “prompt engineering.” It is the ability to describe a physical problem precisely enough that an AI system can help, and to know enough mathematics, mechanics, computation and experimentation to reject the answer when it is wrong.

Mechanical engineering will not become less mechanical. If anything, the opposite may happen. When less time is spent fighting syntax, there is more time to think about models, boundary conditions, measurements and real machines. The best use of AI is not to avoid learning engineering. It is to make good engineering easier to attempt, faster to test and clearer to explain.

References and image sources

- [1] NASA, “Predictive Modeling for NASA's Entry, Descent, and Landing Missions” (2016). NASA image article.
- [2] NASA, “Images and Media Usage Guidelines” (accessed 28 Aug. 2026). Official guidance.
- [3] Li et al., “Fourier Neural Operator for Parametric Partial Differential Equations,” *ICLR* (2021). OpenReview.
- [4] Raissi, Perdikaris and Karniadakis, “Physics-informed neural networks,” *Journal of Computational Physics* 378 (2019). doi:10.1016/j.jcp.2018.10.045.
- [5] Krishnapriyan et al., “Characterizing Possible Failure Modes in Physics-Informed Neural Networks,” *NeurIPS* (2021). OpenReview.
- [6] Elrefaie et al., “On-site aerodynamics using stereoscopic PIV and deep optical flow learning,” *Experiments in Fluids* 65, 183 (2024), CC BY 4.0. doi:10.1007/s00348-024-03914-w.
- [7] Peng et al., “Machine Learning in Thermography NDT: A Systematic Review,” *Applied Sciences* 15, 9624 (2025). doi:10.3390/app15179624.
- [8] MathWorks, “MATLAB Engine API for Python.” Official documentation.
- [9] Ansys, “PyAnsys Documentation.” Official documentation.
- [10] OpenFOAM Foundation, “OpenFOAM Source Code Guide.” Official documentation.
- [11] Pandey et al., “OpenFOAMGPT,” *Physics of Fluids* 37, 035120 (2025). doi:10.1063/5.0257555.
- [12] Cui et al., “The Effects of Generative AI on High-Skilled Work,” *Management Science* (2026). doi:10.1287/mnsc.2025.00535.
- [13] Stack Overflow, “2025 Developer Survey: AI.” Survey page.
- [14] IIT Madras, “B.Tech. in Computational Engineering and Mechanics” (2025). Official programme announcement.
- [15] IIT Madras Mechanical Engineering, “M.Tech. Computational Engineering for Mechanical Systems” (2026). Official curriculum.
- [16] IIT Madras Mechanical Engineering, “Mechanics, Modeling and Machine Learning Lab.” Faculty profile.
- [17] Digital Education Council, *AI in Higher Education Global Survey 2026*. Survey page.
- [18] Department of Science and Technology, Government of India, “Material Opportunities for India's Quantum Technology Mission” (2023), citing 2018 researcher density. Official article.
- [19] Government of India, *Economic Survey 2025–26*, ch. 8, para. 8.48. Official report.
- [20] UNESCO, *Guidance for Generative AI in Education and Research* (2023). Report.

Scope note. Percentages are reported with their original populations and dates; benchmark and case-study results should not be generalized to every engineering problem. Product names are examples, not endorsements.