VIBE CODING FOR SCIENCE

Using AI tools can speed up your work, but are the risks worth it?

By Nicola Jones

ILLUSTRATION BY PAWEŁ JOŃCA

Last year, climate researcher Zeke Hausfather was playing around with climate-data visualizations, trying to find new and shocking ways to show just how fast Earth is warming. He was brainstorming ideas with an artificial-intelligence tool and getting it to code and create them quickly. Together, they made innovative tree-ring-style plots with the months of the year around each ring, the annual circles growing outwards with time and the colours showing temperature. Then Hausfather asked the AI tool: what if these plots were 3D?

The result was what Hausfather calls a thermal helix animation, showing temperature spiralling upwards through time into a shape reminiscent of a tornado (see 'A new view'). In a world in which most people have seen the classic 'hockey-stick' graph of rising global temperatures, it is a refreshing graphic: compelling and beautiful. And, despite being a competent coder, Hausfather had no idea how to make it on his own.

Hausfather, a researcher at the climate data non-profit organization Berkeley Earth in California, is not alone in using AI tools in this way. Thanks to large language models (LLMs), people can now simply ask their computers to write and implement code for graphics, applications, data processing and just about anything else they can imagine.

This kind of laid-back, conversational technique is often called vibe coding. Andrej Karpathy, co-founder of US firm OpenAI, coined the term last year. It refers to asking an LLM-powered tool to build or do something with code behind it, with the user providing clarifying prompts until the results look right. At its purest, vibe coding doesn't involve looking at the code — just the product. But the term has no strict definition, so what counts as vibe coding is fuzzy. Plenty of people with coding know-how start a project by vibing and then check the code by hand, or start coding by themselves and then ask an AI tool to fill in the gaps.

Nature spoke to a variety of scientists, from highly adept coders to complete beginners, and those in the middle, such as Hausfather, who are using AI to stretch the limits of what they can do. Many use AI-assisted coding in their work, and some are intentionally testing its limits. All of them said the AI tools that are already out there are impressive, helping them to drastically speed up their work or try out fresh ideas. But they also warn that the tools should be used with caution, and some had scary stories to tell.

All aboard

In some ways, vibe coding is the culmination of a long evolution of computer interfaces. In the 1960s, people used punch cards to communicate with machines. Computer scientists soon developed coding languages — such as BASIC and later Python — which made giving instructions to computers more natural. And developers made software systems so that non-coders could create with aplomb in limited contexts: Microsoft Word, for example, lets users make formatting changes to documents without knowing how to code. What's new is the unparalleled speed and versatility that LLMs bring to generating code, alongside their quirky tendency to make things up and get things wrong.

Although any LLM can be used to generate code, systems have emerged over the past few years that are specialized for the task, including GitHub Copilot, Anysphere's Cursor, Anthropic's Claude Code, Google's Gemini Code Assist and OpenAI's Codex. These systems can make a functional app with as little as a single-sentence prompt. The results can contain glitches, however. For example, Anthropic's Claude Opus 4.7 is currently leading the pack on Vibe Code Bench — a benchmarking test for the functionality of web applications that have been autonomously coded by AI tools — but with an accuracy score of 71%.

Similarly to other AI products, AI coding tools are improving over time. Those that have been released in the past year or so act like friendly project managers, says Hausfather. You can feed them pages-long descriptions of goals and requirements, and they will respond with, say, a coding plan, suggested verification tests, multiple-choice options for interface design and thousands of lines of code that is well documented with explanations. The progress has impressed Hausfather: today, he says, AI code is "as bug-free as my code".

Among professional coders in the software community, almost everyone now leans heavily on AI, vibing or otherwise. A survey this year by DX — a company in Salt Lake City, Utah, which focuses on measuring developer productivity — showed that more than 90% of software developers use AI coding assistants at least once a month, and entirely AI-authored code now makes up more than one-quarter of customer-facing code.

It's hard to know how many researchers are jumping on the vibe-coding train, but there is clearly interest. When Argonne National Laboratory in Lemont, Illinois, set up a one-day vibe-coding hackathon last June for its researchers, it hit its capacity of 200 participants.

Manuel Corpas, a genomicist and health-data scientist at the University of Westminster in London, says there's a real thirst for vibing in his community. He vibe-coded a project called ClawBio in two days and released it at an Imperial College London hackathon in early March. It acts as a kind of library for pieces of code that are useful in bioinformatics, such as instructions for extracting data from scientific figures, or for developing personalized medication advice on the basis of a genome sequence hosted on your computer. AI agents can pull code from the library to incorporate these 'skills' into their own tasks.

Following its launch, Corpas says, ClawBio racked up an impressive 5,000 downloads in just two weeks, and the community had added dozens of new skills, which were themselves vibe coded, he says.

Good news first

Rosemarie Wilton, a molecular biologist at Argonne National Laboratory, has no coding experience. But she uses established software packages to compile and analyse her data sets on viruses found in waste water, so she attended the lab's hackathon last June to see what AI coding tools might do for her.

Wilton was impressed. She doesn't have a graduate student, but AI tools acted like one. She could ask them to do simple tasks such as running data through one software package after another, cross-checking or producing graphs of outputs in particular ways. The AI tools could run independently all day.

To test out new data-processing pipelines, Wilton would ordinarily start each step manually or contact Argonne's Data Science and Learning division to code the pipeline for her, but now AI can speed up this exploratory phase. If she finds an approach that works well, she says, she would ask the division to code it for her properly before, say, submitting processed data to the state health department.

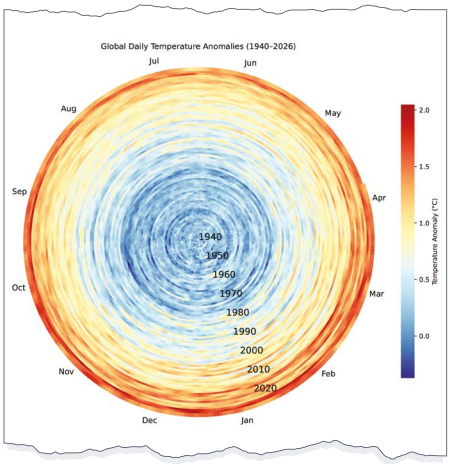
As a side benefit, says Wilton, the ease of vibe coding makes it less intimidating for her to learn some coding herself. "I can learn a lot from it, not having done a lot of Python coding," says Wilton. "It has opened up my world."

Speed and nimbleness are highlights of AI coding for everyone who *Nature* spoke

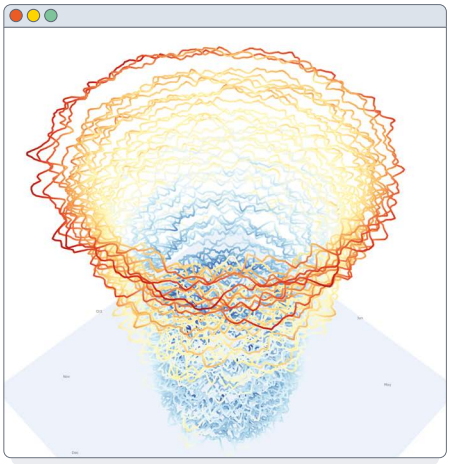
A NEW VIEW

AI coding tools enable researchers to use conversational prompts to visualize their data in new ways. In this case, AI tools were used to plot the increase in global temperatures since 1940.

Tree-ring plot
A request for a tree-ring plot produced this visual, which shows daily global mean surface temperature relative to the 1850–1900 average.



Tornado plot
A request to make the graphic 3D produced this version, along with an animation of the data over time.



SOURCE: ZEKE HAUSFATHER

to. Hausfather says that the code for, say, turning one axis of a graph into a log scale or layering extra information into a chart is often non-intuitive. The ability to tell his computer to do such things in plain English is, he says, "magical". Vibe coding has also enabled him to build and host websites in a day (including a dashboard that creates constantly updated, visually appealing charts of global temperature), which he had never done before.

Tim Hobbs, a theoretical physicist at Argonne who also attended the hackathon, says he uses AI "all the time", because coding is a big part of his job. He explores physics that goes beyond the standard model of particle physics, for example, by analysing reams of data from particle accelerators to test underlying theories or attempt to discover new ones. There are plenty of mathematical approaches he could take, so he has vibe-coded to see which ones seem more promising.

"It's like handing off a problem to an extremely competent graduate student," he says. "I can just quickly try a variety of ideas and maybe discard some that are suboptimal." He says he checks the code behind anything important, such as a research paper intended for publication.

Hobbs is impressed by how cleanly AI code is written, with plenty of hand-holding annotations, often to a higher standard than the human-generated code he sees accompanying published papers. "Human code has a messiness to it, because we're human," he says.

For a paper[1] published earlier this year, Jesse Meyer, an analytical chemist and expert in computational biomedicine at Cedars Sinai Medical Center in Los Angeles, California, attempted to demonstrate the power of AI. His team has previously developed software packages for biological data processing. This time, he used an LLM-powered app builder to vibe-code a pipeline for analysing proteomics and other 'omics' data.

He found that it took less than ten minutes, four well-written prompts and less than US$2 in fees to create something that might reasonably take professional coders months or even years to develop without AI. "The barrier to trying something new is very low," Meyer says. He sees a future in which, instead of publishing code, researchers publish their prompts or 'vibe blueprints' for others to use.

But Meyer emphasizes that his work is just a demonstration of what can be done: he wouldn't recommend vibe-coding anything important without substantial checks. When he published this experiment, he included a disclaimer in the paper's introduction: "Vibe coding is not a substitute for understanding statistical analysis or computational logic."

Use at your own risk

This is a recurring theme among people who investigate vibed code and its outputs. Some researchers who tested ClawBio, says Corpas, have been horrified by misleading outputs created by some of its code, including bad medical advice, although he says people continue to leap in to fix such errors.

Richard Morey, a psychologist who studies statistics at Cardiff University, UK, and who has been coding for decades, decided to pit AI's abilities against his own coding prowess. He asked it to make a graphical interface enabling a user to pick up a curve and drag it around, illustrating the sensitivity or power of experiments with different sample sizes – something he had previously coded himself.

Morey had asked the AI to use a statistical *t*-test, and it labelled its results as such. But when Morey dug into the code, he found it was instead using a *z*-test – a simpler test that Morey calls a "training wheels" version of the *t*-test because it is often used in teaching. This usually gives roughly the right results but fails for small sample sizes. The result is "wrong in a subtle way that you wouldn't necessarily know", Morey says. Perhaps, he speculates, the AI was influenced by a lot of educational material in its training data. "It is surprising what you can do. But it is also clear you can really get into trouble," he says.

> **"Vibe coding is one thing, vibe debugging is chaos."**

AI's sycophantic tendency to always complete an assigned task can result in code that runs but is wrong, many researchers say. Vibed code might, for example, make up missing data to help an algorithm run, or even smooth a graph to make it look as expected.

When instructions are ambiguous, AI can make assumptions that lead it astray. Similar problems pre-date AI. Around one-fifth of genomics papers analysed for a 2016 study[2], for example, had Excel-related formatting errors in the supplementary files, in which the program had automatically (and erroneously) converted gene names into dates or numbers.

Although code is more formal, and therefore easier to evaluate, than AI-generated prose, that doesn't mean it is trivial to spot errors and fix them, Hobbs says. "It takes extraordinary effort and expert domain knowledge." Working out why AI code went wrong can be "at least as hard of a task as just writing it from scratch in the first place". Even if your code works for now, adds Meyer, trying to update or maintain it in the future might prove problematic.

For a study reported last year, Veronica Pimenova, an information scientist at the University of Michigan in Ann Arbor, and her colleagues trawled through more than 190,000 words from social-media threads and about a dozen informal interviews with vibe coders to unpick the joys and pains of the method[3]. They found that AI code can cause version-control problems, mixing up old changes with new ones or losing track of the most recent version, and can be overly verbose, making tracking down problems a nightmare. As one commenter put it: "Vibe coding is one thing, vibe debugging is chaos."

Given the known limits of AI, researchers had some specific advice for anyone wading into vibe-coding waters. Prompts need to be detailed and avoid ambiguities, these researchers say. Based on his own experience, Hausfather says, current AIs can get into trouble in 'mixed media' tasks, such as asking them to extract data from an image of a graph.

Meyer emphasizes that AI code should only be trusted if it's verifiable – if you can devise specific tests to ensure it is doing what you think it's doing. AIs might propose their own tests, but take those with a grain of salt, he says. "It can make some pretty hilarious tests for itself where it just always passes."

Others say that someone on a research team still needs to understand their code. The code, many say, whether written by humans, AI tools or a combination of both, should be published alongside the research paper, and ideally be peer reviewed – which doesn't always happen.

As with any use of AI, there are logistical issues and unresolved ethical questions. Many journals require authors to disclose AI use, including for coding, but the threshold for disclosure is vague and compliance is limited. What might constitute plagiarism or copyright violation – either in AI coding tools themselves or in their code – is also not clear. AI coders are trained on publicly available code, but that doesn't mean that all of that code is available for any use. One court case, for example, has questioned whether AI coding tools are violating the licences that protect some open-source code. Other issues relate to bias and hallucinations that might go unnoticed, along with the risks of confidential data possibly leaking.

But researchers told *Nature* that the benefits AI coding offers for speed and creativity mean it can't be ignored. "It's never perfect, and sometimes it gets things wrong and it's exasperating," says Corpas. "But when you're able to harness that horse, you see the world from a whole new perspective. You can go 20 times faster."

Nicola Jones is a freelance journalist in Pemberton, Canada.

1. Meyer, J. G. *J. Proteome Res.* **25**, 1191–1197 (2026).
2. Ziemann, M., Eren, Y. & El-Osta, A. *Genome Biol.* **17**, 177 (2016).
3. Pimenova, V., Fakhoury, S., Bird, C., Storey, M.-A. & Endres, M. Preprint at arXiv https://doi.org/10.48550/arXiv.2509.12491 (2025).