Computational Thinking

How Computer Science Is Revolutionizing Science and Engineering

Computational thinking is becoming increasingly essential to the academic experience in the Division of Engineering and Applied Science and at Caltech more generally, with the challenges brought by a rapid increase in the availability and volume of data across disciplines and applications. The Institute not only has one of the leading computer science programs in the country, but it has also recently added a new major in information and data sciences to build on that area of strength and maximize its reach and impact. *ENGenious* sat down with a few of the faculty members who are leading this effort—one that promotes the cross-disciplinary perspective and collaborative research that are hallmarks of a Caltech education—to discuss the current state of the field, how it has evolved on campus, and what exciting opportunities await. Around the table were professors Victoria Kostina, Andrew Stuart, Thomas Vidick, Adam Wierman, and Erik Winfree.

Computational thinking is one of major research thrusts identified by EAS as the division looks to the future—and it's a complex one. Erik Winfree, professor of computer science, computation and neural systems, and bioengineering, explains that "it means different things to different people, but overall, computational thinking is the percolation of ideas from computer science into other areas of science." This has come

Left to right: Andrew Stuart, Thomas Vidick, Victoria Kostina, Erik Winfree, and Adam Wierman

increasingly to mean chemistry, physics, and biology. Winfree's own research interests focus on "how molecules store information, how that information directs other processes, and how we can formalize and analyze that information." This sort of behavior can be precisely and mathematically defined, modeled, and analyzed using computer science.

Andrew Stuart, Bren Professor of Computing and Mathematical Sciences, remembers being taught from a young age that mathematics was the language of science. Now, he says, it's computational thinking that is becoming the basic language of all science and technology. "It is how we

Adam Wierman, professor of computing and mathematical sciences, executive officer of the Department of Computing and Mathematical Sciences, and director of the Information Science and Technology initiative at Caltech, also describes computational thinking in terms of framing questions. "When you look at a topic from a computational point of view, you want to know: What are the basic rules? What is a model for how molecules react? What are the capabilities of the model? What can it do or not do? Then, how do we use programming languages to represent the things a system can do?" These key questions are also of interest to strategic

areas of science, you look at what exists. In other areas, you look at what is possible. Computer science gives us a language for what's possible," he explains. "The resulting characterization of the design space allows us to look at the limits and think of things that may not have existed before. This design space lets us ask more interesting questions."

These faculty members, who apply computational thinking to topics ranging from economics to engineering to climate modeling, have collectively witnessed the growth of computer science at Caltech, from the days before an undergraduate CS degree was even offered (as when Winfree was appointed in 1999) to the exciting addition of the new

biological computing, robotics with mechanical engineering, or aeronautics—a setup that illustrates this integration of a computational approach across areas of study. "This is a 'CS + X' method, with the X being different areas of translation," he says.

Stuart also points to this model, noting that while one can find similar approaches at other schools, nowhere has it taken off quite like at Caltech. "The Department of Computing and Mathematical Sciences at Caltech is unique because we are a department of both continuous thinking and discrete thinking. We are one of a handful of programs in the U.S. that have applied math and computer science sitting under the same roof," he says. "Here we have traditional majors but also the opportunity for interactive degrees; courses are cross-listed and faculty conduct research in multiple departments. There is true collaboration. This is not something that is pressured to occur; it just happens because of the type of people and the environment that has been created." Winfree notes that the particular strengths that enable such a rich environment to develop at Caltech are not due solely to its small size. "One of my favorite attributes of Caltech is the diversity of students that take my class," he says. "They are trained as biologists, bioengineers, physicists, and computer scientists, but they come together. They educate each other."

At Caltech, students learn how to solve problems, how to think, how to not allow fears and obstacles to hinder the acquisition of knowledge. In an age of short attention spans and pressure for fast results that might keep others at a

surface level, they are trained to go broad and deep. Kostina elaborates: "We are not sacrificing depth by offering breadth, because all our students have good foundations." Caltech forces you to challenge yourself, she says, by taking difficult and interdisciplinary courses. The unique new major in information and data sciences embodies this approach, providing a deep mathematical foundation for students while also requiring engineering courses with applicability to different fields. "If I were a Caltech undergrad today, I would take these classes!" Kostina says. "We are in a very exciting moment in history, and by creating this program, Caltech is responding to the great demand in data science." It is clear that the faculty members are eager to engage and encourage a variety of students—including those who may never have considered the interdisciplinary approach. "What we want to do with data science at Caltech is make our students connect the discrete view and the continuous view," says Wierman. The aim is to enable them to "incorporate this into scientific computing and model more data-driven artificial intelligence (AI) machine learning." And he is emphatic about this being the right place for this to happen: "Caltech is a balanced mixture of educators that excel at these areas and students who want to have such interactions and discussions."

The excitement is palpable as Wierman discusses the way all this came about, as the outgrowth of a long process of innovation within traditional modes of inquiry. "When the concepts of 'computational thinking' or an 'algorithmic lens' were first developed, they were broad and ill-defined. It took

a long time for these ideas to percolate into other disciplines and inform change. In the last 10 to 15 years, we have seen the meaningful emergence of this collaboration of computer science with different fields." Stuart adds, "Continuous, discrete mathematics has risen in importance as more people understand this model of thinking. At the same time, there is a need for continuous mathematics in computer science. One great example is machine learning. In the next decade, we will incorporate continuous mathematics into computer science." This integration will continue to be implemented at all levels of education, notes Wierman. "We are continuing to emphasize the importance of computer science, especially programming. This focus is even at the elementary and high-school level. My kids are learning recursive programming! This knowledge base and ease of interaction with computer science bodes well for the future of Caltech." ⬛ⓝⓖ

Computer science gives us a language for what's possible.

transfer ideas that were originally developed as the solution of one technological problem into a concept applicable to many other scientific ideas," Stuart says. "The final number is no longer the solution, but [the solution] is, instead, the algorithm itself." This shift from a more traditional mathematics-centric view to one based around computational thinking is especially conducive to collaboration and the translation of ideas to a variety of fields and problems.

Thomas Vidick, Professor of Computing and Mathematical Sciences, explains that with this shift has come a change in the order of inquiry, the approach to questioning. "In my field of quantum computing," he says, "we first started with quantum mechanics and its corresponding laws. Then came the machines. Now, we can ask a new question: What kind of machines can you build out of these laws of physics?"

partners—such as Newport Beach–based investment firm PIMCO and Amazon Web Services—that are supporting the students, postdocs, and researchers who are pushing computational thinking forward at Caltech.

Victoria Kostina, Assistant Professor of Electrical Engineering, applies this mode of questioning to her work in the field of information theory. Kostina and her team always start with a model of a system. "This model is abstract and based on probability," she says. "Then we ask ourselves: What are the goals of the system? What are the tasks the model system should perform? And what are the fundamental limits that are achievable under the physical constraints of that system?"

Winfree sums up the influence of computational thinking and its potential to invigorate so many other fields. "In some

major in information and data sciences this year. Given the close-knit nature of the scientific community here and the way the Institute promotes—even necessitates—true collaboration, Winfree argues that in a way, Caltech's undergraduates all become computer scientists by nature. "Because of the small size of Caltech, undergraduates get a feel for everything. Caltech is the liberal arts education for science," he says. Wierman agrees: the computer science program is designed for this kind of place and education, with an emphasis on fluid boundaries and the translation of ideas or approaches among disciplines.

In his teaching, Wierman gives particular attention to the many junior and senior students who take project sequences: three-term course sequences in either traditional CS or in networking and distributed systems, quantum computing,