Becoming a Computational Scientist

The editorial board for Computing in Science & Engineering meets only once a year. One topic we sometimes discuss is whether we have some (good) ideas for new columns. We certainly welcome suggestions from our readers, for we're here to serve you. We also discuss how we can encourage young people to enter our field. One of my ideas for a new column is "How I Became a Computational Scientist," a series of biographical pieces that might inspire others to join in our fun. At the risk of killing my idea for a column and discouraging people from entering the field, I thought I would tell my story.

How I Became a Computational Scientist

It certainly wasn't by design. I enjoyed mathematics and science from an early age. In high school, I was captain of the math team. Entering Cornell, I thought I would study biochemistry so I could understand what life was all about. However, to study biochemistry you had to satisfy a depth requirement and a breadth requirement, and for all I know a length requirement. In physics, they told you what to take every semester, and I had the most marvelous, inspiring introductory physics professor, David Cassel. (I also had a nice, but quite clueless advisor as a freshman, who later won a Nobel Prize. The cluelessness only applied to knowledge of the undergraduate course requirements.) I wound up majoring in both mathematics and physics. By the way, I have an interesting story about how I got out of taking my first-semester calculus final, but that's for another time.

I wound up in a US National Science Foundation Undergraduate Research Participation program mentored by an Oberlin College professor after my freshman year (this program was the Research Experiences for Undergraduates program of the last millennium). That took me to Chalk River Nuclear Lab in Canada for the summer. I learned a little about sailing and that a person could eat French fries with malt vinegar rather than ketchup.

At Chalk River, I was able to get to the card reader at night and load my own programs to reduce the turnaround time debugging my programs. For younger readers, computer programs had to be laboriously typed onto 80 column-wide cards, and a machine punched holes in the cards as you typed. The card reader was the only way to get the computer to read your code, and the compiler could arrange for a bunch of unreadable punch cards to be produced that contained your executable. Except at night, there were operators protecting the computer from naive users (like me). As an undergraduate, I took a two-credit course in Fortran programming, but otherwise seldom had contact with computers. I don't recall anyone using computers for instruction. I took a course in probability, but I thought statistics was a messy subject and wanted nothing to do with that.

Fast forward to graduate school. I still wasn't doing much with computers. My thesis, entitled “A New Twist in Deep Inelastic Scattering” on perturbative Quantum Chromodynamics (QCD), was the product of research done almost exclusively with paper and pencil. I did use a computer for writing my thesis, which involved a box of punch cards to feed a primitive word-processing program, and all the equations were hand written. Upon graduation, my father (wanting to help boost my career) asked if he could get me a nuclear reactor or something. I replied, “Dad, I am theorist. I just need pads and pencils.” If only he had made a similar offer after I became a computational physicist—I might have my own supercomputer.

During my first postdoc at Argonne National Laboratory, Michael Creutz, Claudio Rebbi, and others were pioneering the Monte Carlo approach to lattice QCD. Creutz was kind enough to share his code for SU(2) lattice gauge theory, and I had a great title for a paper: “Looking for Glue in SU(2).” This could have been the first calculation of the glueball spectrum, but I really didn't know enough to do that calculation, and I was busy doing other things.
A couple of years later, Don Weingarten from Indiana University visited Fermilab (where I was then employed), and he, Hank Thacker, Paul Mackenzie, and I started a calculation of the coupling of the rho meson to pions on a VAX computer at Fermilab. We had all of about 20 configurations of size $6^2 \times 12 \times 18$ over which to average. It took weeks or months to do calculations we could now do in the blink of an eye. Today, we have grids up to size $96^3 \times 256$, and they include virtual quark-antiquark effects that were missing in the early calculations.

In 1982, I moved to the University of California, San Diego. I was still working with Weingarten, Thacker, and Mackenzie remotely. (This was before the Internet, and I used quite a bit of time on the dedicated phone lines the experimentalists had to connect with Fermilab.) I was also doing some phenomenological work with Tom Weiler on a fourth-generation lepton. The next year, I reconnected with my graduate school housemate, Doug Toussaint, who had been working on lattice field theory at the University of California, Santa Barbara (UCSB) with Bob Sugar and others. Working with Bob Pierson and John Richardson at the UCSB Physics Department and the Institute of Theoretical Physics, Doug had built a special-purpose computer for the Ising model. He also had a Sun workstation and there was a VAX in the department. After more than 200 papers and close to 30 years later, Doug, Bob Sugar, and I, plus a much-expanded collaboration, are still working together.

One of our first projects involved an ST100 array processor. At half a million dollars, it was a lot less expensive than a Cray supercomputer, so the NSF was willing to buy one for Bob Sugar at UCSB. With a speed of 25 MHz, it had four floating-point processors, specifically two adders and two multipliers. Each processor was pipelined, so every floating-point operation took several cycles, but if you could keep the pipeline full, you would get up to 100 megaflops per second (Mflops), a tremendous speed in those days. We were able to sustain 87 Mflops for SU(3) gauge theory calculations.

Don Weingarten left Indiana University for IBM to build the GF11 computer with a peak speed of 11 gigaflops per second, and I was lucky enough to get hired by Indiana University.

In the ensuing years, computers got faster, the projects were exciting, and there were always new things to learn: physics, programming skills, numerical methods, statistics (yes, even statistics, which I now enjoy), and computer technology. Starting in the early 1990s, we formed the MILC collaboration, which has been a great source of joy and camaraderie in my career. About 10 years later, I did a sabbatical back at Fermilab, which helped to cement a growing collaboration with some of my early collaborators there. The Fermilab Lattice group and MILC have worked closely and productively together since that time. Once we started including a dynamical strange quark in our calculations, we were able to make accurate calculations of known quantities and several predictions of other quantities that were verified in subsequent experiments.

There has also been a great international community in lattice gauge theory, with annual conferences moving among Asia, Europe, and the US. I’ve been to Seillac, Capri, Tsukuba, Amsterdam, Bielefeld, Melbourne, Edinburgh, Bangalore, Berlin, Dublin, Regensburg, Sardinia, and Cairns from that conference series alone.

So, although I never planned to be a computational physicist, I’ve thoroughly enjoyed being one. I’ve also enjoyed being involved with CiSE, and I hope you will let us know if you would like to see more profiles of computational scientists.

Steven Gottlieb is a distinguished professor of physics at Indiana University, where he directs the PhD minor in scientific computing. He’s also the Associate Editor in Chief of CiSE. His research is in lattice quantum chromodynamics (QCD). Gottlieb has a PhD in physics from Princeton University. Contact him at sg@indiana.edu.