The Future of Extended Analog Computers

Jonathan W. Mills
Computer Science Department
Indiana University
Bloomington, Indiana 47405

ANALOG COMPUTERS DISPOSED OF

There are two kinds of computers: analog and digital. (Also hybrid, meaning a combination.) Analog computers are so unimportant compared to digital computers that we will polish them off in a couple of paragraphs.

— Ted Nelson, *Computer Lib/Dream Machines*, 1974.

1. Introduction

Historically, analog computing has been used to simulate specific systems. It has not been a paradigm for universal computation, but it is this paradigm we will examine here, fictionally and playfully near the end of this paper. However, the theory and practice of universal analog computation is real. It is not as implausible as the typical reader might believe.

A number of increasingly sophisticated prototypes of universal analog computers have been built since the mid 1990's at Indiana University. The prototypes have been used to study applications that include models of the vertebrate reticular formation to generate behavior for intelligent agents, image classifiers, feedback controllers, and numerical accelerators for matrix computations. In theory the device is universal; in practice it supplements digital computers exactly where they are "running out of steam," that is, in the difficult-to-program and costly-to-run massively-parallel Grand Challenge problems.

Analog computation balances digital computation. The "Tao of Computing" can be envisioned as digital *yin* and analog *yang*. Both are necessary. Each contains a tiny core of the other. Either alone meets limitations in practice. Together they are unified in computability theory.

Vannevar Bush championed computing by analogy with his Memex, believing it to be the most natural way for humans to interact with computers. However, just as Ted Nelson's Xanadu changed as technology enabled it in the form of the World-Wide Web, so future analog computers will almost certainly be radically different from the prototypes in use today.

2. A Brief History of Extended Analog Computation

Gustav Kirchhoff was one of the first people to use an analog device to study a specific set of problems, heat diffusion in a plane and current flow in a ring. By applying heat and currents as boundary values and measuring the temperatures and voltages at points within the conductive sheet analog, Kirchhoff experimentally solved boundary-value partial differential equations in 1845, some 40 years before Babbage designed his programmable digital computer. Kirchhoff's simulations are notable because they preceded a general and universal Turing-complete theory of analog computation by almost 150 years. What retarded the development of analog computing in theory and practice for so long? Very likely it was the specificity of the problems solved using analog computers. An analog computer derives its function by being placed into a well-defined relationship with another system, that is, an analogy. The concept of a universal analogy is thus an oxymoron, and oxymorons, by definition, resist implementation. Yet, as we will see, the theory of analog computability led to implementations of new types of analog computers—and a new computing paradigm.

Claude Shannon proposed a theory of analog computability in 1948, in a paper that provided the mathematical basis for Vannevar Bush's differential analyzer. While many researchers built such machines, including a fascinating version constructed by a man named Crank—seriously!—of Meccano components (known to readers in the USA as an Erector Set), only Shannon studied the computability of the device. Shannon's result was that the differential analyzer, made rigorous as the general-purpose analog computer, was mathematically equivalent to a system of algebraic differential equations. This result was not related, nor was an attempt made at the time to relate it, to the theory of computability that led to Turing's notion of universal computation.

Shannon's work was slightly flawed in its presentation, as discovered by the mathematician Marion Boykin Pour-El, who provided a revised version of the proof and a more specific model of general-purpose analog computation. Pour-El's proof was in turn found to be incomplete by Rubel, who finally put the link between general-purpose analog computers and algebraic differential equations on a rigorous footing.

The Shannon–Pour-El–Rubel general purpose analog computer can only compute a function over time. Mathematically, everything, even a constant, is a function of time. A general-purpose analog computer may be connected to other general-purpose analog computers *ad infinitum,* but because they only compute functions of the form f(t), where t = time, it has been proven unable to compute certain functions. A general-purpose analog computer is not Turing-complete, and so is not a universal computer.

In the 1950's through the 1970's this obstacle was not noticed because people would run multiple independent trials—iterative analog computations—to solve problems involving spatial parameters. For a short time hybrid digital-analog computers were used for iterative analog computations. There were advantages to using an analog processor in a hybrid computer, including the ease of understanding the system cited by Bekey and Karplus:

> *Provisions permit the engineer to experiment by adjusting coefficient settings on the [analog] computer, thereby gaining direct insight into system operation.*

As much as Bush would have applauded this approach, the flexibility gained by improved programming languages for digital computers, and the improvements in digital computer architectures, led to the demise of general-purpose analog computers and hybrid machines.

After Project Whirlwind, analog computation and analog machines become more and more specialized—and unpopular. By the 1960's, digital machines and paradigms had won the field. Digital computers, not analog machines, have since defined how we think about computing: its theory, its implementation, and its capability.

Yet some holdouts remained. As late as 1996 Professor Oleg Wasynczuk of Purdue University used an analog computer to solve powerline transmission problems that were too complex for, and ran too slowly on, digital computers. Still, by 1964, when Granino and Teresa Korn published the last edition of their book *Electronic Analog and Hybrid Computers* the field was dying. By 1974, when the last edition of Arthur Hausner's *Analog and Analog/Hybrid Computer Programming* was published, the field was essentially dead. A generation of computer scientists would grow up without ever having seen a slide rule, and unaware that analog computers had ever existed.[1]

It was in the years of transition to digital computing that analog machines fell into obscurity, believed useless in a modern world. The history of the switch from analog to digital was written largely by the winners, and, like all such histories, overstated its case. What Rubel and others re-examined were not matters of speed or precision but the questions of what can rendered and understood through a machine that does computation. Advocates of the digital machines believed that speed and precision were all that mattered. Those that worked with analog machine worried about the requirement to reduce computations from phenomena to models to equations to digital computations. It was this willingness to trade technical advantage, speed and precision for realism (a point-by-point correspondence between phenomena and the computer) that troubled those like Bush, who believed in analog machines. But to focus on either view alone is to lose sight of the different strengths of different computing paradigms.

Roughly beginning in 1980, various investigations by computer scientists into the theory of computability swung into the sphere of mathematicians who studied general-purpose analog computers. Notable among them was the real-valued computational model of Blum, Shub and Smale that introduced continuity as an algorithmic concept, and Vergis, Steiglitz and Dickson, who examined the complexity of analog computation, and hypothesized a "strong" Church-Turing-Rosser hypothesis; essentially, that analog and digital computation was equivalent.

A small group of mathematicians and computer scientists, working independently of each other in the early 1990's, proposed models of analog computation that were Turing-complete. In 1993, two years before he died, Lee Rubel developed a new mathematical paradigm for computing, the extended analog computer. The extended analog computer adds spatial parameters via a partial differential equation solver. This device, which is complex when implemented as a digital computer and its program, is astonishingly simple as a conductive sheet. In fact, conductive sheets and other related analog devices were how the Grand Challenge problems were solved before massively-parallel electronic digital computers became available. Add a conductive sheet to a general-purpose analog computer, and you get an extended analog computer. Mills was the first to realize that extended analog computers can be built out of the silicon substrate of a VLSI integrated circuit (known to VLSI designers as "empty space") and a handful of diodes. When questioned by the silicon fabrication facility about a design that had connections that appeared to go nowhere and do nothing, he coined the saying he remains fond of today:

> *Empty space computes.*

---

[1] Professor Steven Johnson, a colleague of mine at Indiana University's Computer Science Department, contributed a five-foot long yellow Pickett teaching slide rule. It is displayed over the departmental lounge, set to compute π. He found it by accident in a high-school classroom where it was no longer used. Our students now know what a slide rule looks like!

Rubel's model of the extended analog computer was an extension of the general-purpose analog computer, and a direct descendant of Bush's differential analyzer. Rubel added various components to the general-purpose analog computer, but the most important, the "quintessential black box", was the partial differential equation solver. Rubel was able to prove that certain problems that were not computable with a general-purpose analog computer, notably the Dirichlet problem on the disk, were computable using the extended analog computer. Rubel was unable to prove that some transcendentally transcendental functions were not solvable using the extended analog computer, or, stated another way, that the extended analog computer was only Turing-complete, and not a super-Turing computer.

Independent work by Siegelmann and Sontag on the computability of rationally-valued and real-valued neural networks suggests that this was due to Rubel's use of real numbers in the extended analog computer: rationally-valued nets are Turing-complete, while real-valued nets are super-Turing (that is, capable of solving problems that no Turing machine can solve). Other independent researchers including MacLennan, Wolpert, and Moore developed models of analog or continuous computation that corresponded to these results. However, the notion of a super-Turing computer remains a mathematical concept, not able to be implemented with any technology currently envisioned. Maass showed that super-Turing computation is theoretically impossible in the presence of noise, thus ruling out all known technologies to construct either analog or digital super-Turing machines.

However, the proof that analog computers are Turing-complete was a powerful incentive to go looking for one: if a universal analog computer existed, what would its primitive components consist of? How would it be programmed? What would an interface to it look like? What kind of applications could it solve, and of those, which ones could it solve efficiently?

Thus began a long and sometimes frustrating search that led to the construction of small, simple prototypes of extended analog computers at Indiana University. These machines are the descendants of the conductive sheet analogs of Kirchhoff and the general-purpose analog computer of Bush and Shannon, as theoretically merged by Rubel in the extended analog computer. The VLSI implementations are just as restricted, although in different ways, as any digital computer: no digital computer is a Turing machine with an infinite tape; no extended analog computer that can be built operates on infinitely precise real numbers. However, within its domain, each machine solves problems efficiently.

Next, let us try to predict the future of extended analog computers.

4. You Can Never Go Back, or How Technology Affects Bush's Memex

Bush's ideal of an analog computer that paralleled the human process of thought, visualizing and rapidly exploring patterns of data directly, without programming, suffered the curious paradox of being as exactly right in principle as it was precisely wrong in practice. The future is never what we think it will be, in part because we cannot conceive of the effects that other improvements to technology, or other breakthroughs, or even the price of chips in China, may have on our visions.

A good example is Ted Nelson's Xanadu project, the dream of a study carrel that embodied a vast hypertext database and authoring system. Xanadu was intended to be a franchise for knowledge, just as McDonald's® is a franchise for fast food. But Nelson's vision of hypertext could not encompass the future developments of a global network that reached into many homes and businesses. Nelson's vision was shaped at the start of an era when networks were limited to academic and military use, and when a dumb terminal felt (and was!) fantastically faster than a 110 baud teletypewriter, and when the PDP-8 was the pinnacle of personal computer architectures. Franchising Xanadu was brilliant but unnecessary as the web penetrated the planet. By the time hypertext became popular, the vision of Xanadu-as-conceived was obsolete.

If extend analog computers prove viable in the future of computing, then they will also have effects that are difficult to foresee. However, given that they will be constrained by the same principles of technology that constrain digital computers, such as low cost leading to penetration into the mass market of consumer goods and automobiles, as well as the simultaneous development of higher-performance computers that give better cost-performance ratios, there are some interesting scenarios that can be imagined.

The next section of this article is fiction that could become fact. Jules Verne's explorations of future technology provide the model for the stories, updated and streamlined. Three short scenarios from the future year of 2030 A.D. are described. They show how supercomputers that cost pennies could be embedded in a toaster, in a hybrid analog-digital laptop, and a massively-parallel hybrid supercomputer that solves problems in seconds that might take weeks, months or years today. In this future laptops and workstations are "programmed" in a virtual world where intelligent agents—say, an army of talking ants—multiply the user's ability to construct simulations of systems as complex as galaxies or DNA molecules, and watch them change in real-time movie-quality animation.

Vannevar would have loved it.

5. Extended Analog Computers of the Future

There are supercomputers in my toaster!

You wake up, go into the kitchen, and put four supercomputers to work. Inside the toaster an array of ceramic heating squares begins to glow red hot. A fiber optic sensor array with binary optic lenses, only approximately focused on the muffin but capable of precise measurement with their overlapping "eyes," transmit the infrared reflection of the surface of your muffin back to four extended analog computers.

Each one registers the changes in infrared intensity to develop a toasting map for the surface of your muffin it monitors, and having been configured by the settings "light brown" and "whole wheat" raises or lowers the temperature of each ceramic tile to produce a heat map that will toast your muffin perfectly, with no burnt edges, and that colder spot in the middle perfectly done. When the muffin halves are a crusty, even, light brown, they pop up, piping hot. You did not set a timer, but simply sipped a bit of coffee while your ToasterKing checked the progress of your muffin. After the muffin popped up, the sensors inside were cleaned automatically, ready to transmit the raw data for the next computation. You leave the four supercomputers idle until tomorrow morning, ready to make perfect toast—if you want it. You might choose to have a bowl of cereal instead.

Crayon Inside! (an analog supercomputer in your laptop)

"Crayon" became the popular name that applied to the extended analog computers that are now found everywhere, from embedded controllers in toasters to massively-parallel workstations.

Your laptop, built in 2029 A.D., has a "Crayon" inside that accelerates the intensive matrix operations needed to compute the animations, visual displays of data, gesture processing, and neural field interactions that turn your laptop into a quasi-intelligent and devoted assistant.

Your accelerator looks unimpressive from the outside, being a steel-gray flat square embedded in optical memory slabs. But on the inside gigabytes of data are used to digitally configure its functions in a nanosecond-by-nanosecond response to the analog computation, and binary optic lenses—nothing more than fancy Fresnel lenses—are used to translate the binary data into optical inputs, and split the analog outputs back to binary to perform the equivalent of trillions of computations each second. Because the machine is an analog computer, it doesn't actually perform each operation explicitly, as a digital computer would, so this number cannot be measured exactly. Instead, analog computers rely on the laws of physics to perform the mathematics implicitly. This is one of the characteristics of analog computing: the precision of the computation is "buried" inside the analogy.

The "Crayon" processor looks like a tiny "sandwich", but is a marvel of micromachining and VLSI fabrication. The two pieces of "bread" that make up the sandwich are the input and output layers, an array of infrared laser dots and a binary optic lens to focus data onto the "meat" of the sandwich: a carefully implanted thin layer of an organic semiconductor that solves partial differential equations. Output is obtained from another slice of "bread": an array of tunneling electron microscope probes that measure the picosecond changes in voltage in the conductive sheet, which feed the Lukasiewicz logic arrays that are configured to the specific function needed for each 20 picosecond computing cycle.

This tiny "sandwich" is the equivalent of dozens of parallel digital processors that can be individually configured to perform complex steps in an even more complex computation, and cycle results back to optical memory quickly enough to perform thousands of trillions of operations each second. What is amazing is that these operations are not the ones that your laptop uses when you write a letter in your word processor, or balance your checkbook. Most of the tasks you do are the same things a digital computer did 60 years ago—and still does today.

What differs is the operating system you use, and the need for extremely rapid recognition of the gestures you make above the flat keyslab, and the keyspots you touch, and the mistakes your laptop has learned that you make, so that you can look at what appears to be a desk with paper and tools on it, and use those tools just by gesturing. A lot of the Crayon's power goes into its artificial intelligence and agent routines, using the field computing paradigm pioneered by MacLennan and applied by others to create the powerful recognition and behavior systems that lie at the heart of the agents that you interact with. They run your programs in a DWIM operating system: "Do What I Mean". It does, too—or it asks!

In an extended analog computer, much of the operating system and many non-symbolic applications are implemented with parameter templates for the hardware, not algorithms and programs. In effect, an analog computer is "programmed" by vast look-up tables, a very simple concept (and one that has had an equally vast effect on computer scientists, who suddenly saw much of their "science" disappear as it was replaced by physics). With template-based configuration for virtual realities and AI systems, programming the operating system and the specialized virtual worlds became elevated to an art form similar to sculpting, painting, or playing a musical instrument. New doctorates now apprentice themselves to "masters" to learn the “feel” of the computer, an art that they refine over time with the help of an army of intelligent agents who have watched their predecessors. Let’s meet an artist at work.

Massively parallel analog biocomputation evolves new pharmaceuticals

Where your laptop has a single Crayon inside, a modern workstation has over one million. Where workstations in the late 20th century had four, eight, 16 or possibly 32 processors, a workstation today must have a massively parallel array of “Crayons,” supported by digital processors to perform symbolic and discrete computations. The Evolution Engine has one megachip of “Crayons” inside, and an array of 1024 digital Decium processors from Intel®, each Decium roughly equivalent to ten 1990’s-era Pentium processors. The overall computing power of the single Evolution Engine is not definable using any of the metrics used by computer scientists in 2000 A.D. There is no way to measure the operation of a Crayon in terms of MIPS, MOPS, or MFLOPS. It is, after all, a non-algorithmic non-Von Neumann machine.

Let us talk with Dr. Sanjay Patel, Senior Design Artist of Artificial Pharmaceuticals, Inc., as he works with a massively-parallel extended analog computer, the Evolution Engine he has nicknamed “Eve”.

“Do you see those little green ants over there? They are part of Eve. That’s my name for my Evolution Engine. You can think of the way she operates as splitting herself up to assemble a random collection of bushes from the ones that met my criteria. Those bushes are pharmaceutical molecules. When the bushes are planted then Eve goes to work to evolve them. That’s where the Crayon array is used. Each bush goes through a series of tests against membrane patch identification, neurotransmitter activity, bond energy levels (which are critical by the way, we have to build in a way to flush these undesirable pharmaceuticals out of the system after they have finished their work), and reaction path sequences.

“All of these tasks would have taken a digital processor centuries to perform, if the mathematics had to be done in binary. In fact, the first, simple artificial pharmaceutical designs took several years to complete on a network of digital supercomputers. I have been told that the biggest problem was getting the machines to talk to each other using their different primitive programming languages over a clumsy network that spanned the continent. Eve fits in my office!

“Eve is fast. She has the equivalent of a network of thousands of supercomputers in her, and they can do the computations we need to evaluate a molecular structure in seconds. She can evaluate a million molecules at a time, or just ten. Why would I do that? Well, I use Eve at different resolutions to increase or decrease the precision with which I manipulate the molecule. Yes, back in the ’90s they thought that analog processors had a limited precision, but those poor fellows only had one analog machine to use at a time, and their computers were built out of discrete components—op amps, resistors, etc. Eve provides the scaling I need to solve really interesting problems.”

6. Conclusions

This paper opened with a quote from 1974, when the advocates of analog computing had lost an undeclared war to the advocates of digital computing. Analog computers became extinct, leaving only their fossil traces in the operational amplifiers and analog-to-digital and digital-to-analog converters needed to interface digital computers to the world around them. No text on computer architecture, and very few texts on the history of computing give analog computers even more than a passing mention today.

A few stubborn theorists—Rubel and MacLennan foremost among them—and advocates of neural computing—Carver Mead primarily—led the renaissance which the author believes will become a revolution, resulting in the next computing paradigm described in this "future history" of Crayon machines. Shortly before his death Lee Rubel encouraged computer scientists to pursue the design and application of extended analog computers. He said:

> *The future of analog computing is unlimited. As a visionary, I see it eventually displacing digital computing, especially, in the beginning, in partial differential equations and as a model in neurobiology. It will take some decades for this to be done. In the meantime, it is a very rich and challenging field of investigation, although (or maybe because) it is not in the current fashion.*

Acknowledgments

Lee Rubel, Bruce MacLennan, and Bill Kilmer have all encouraged me to pursue the design of VLSI extended analog computers and field computers. Lee died in 1995, before much of the recent work was accomplished. Since 2002 the students in my VLSI Design classes at Indiana University have been applying extended analog computers in a wide variety of applications. If the future in this paper becomes reality, they will have been instrumental in bringing it about.