Newsjournal of the Society for Industrial and Applied Mathematics sinews.siam.org

siam news

Volume 55/ Issue 6
July/August 2022

The Brain Is a Dynamical System

By Lai-Sang Young

In the human brain, $\mathcal{O}(10^{11})$ neurons communicate mostly via electric impulses. Researchers often model the cerebral cortex as a coupled dynamical system: a large network with nodes that comprise smaller subsystems representing the dynamics of individual neurons. Of course, neuroscience is much more than such a network. Neuronal dynamics are driven by complicated biochemical processes on the molecular and intracellular levels, and outputs of cortical interactions affect cognition and behavior. But dynamical interactions of neurons play an integral role in brain function, and mathematics—particularly dynamical systems—can shed light on biologically meaningful models of these interactions.

Here I focus on primate vision [4]. The macaque monkey's visual cortex is very similar to that of humans. This brain region is rich in data because experimentalists can access it easily, and its close proximity to sensory input enables researchers to correlate cortical responses directly to visual stimuli. These features make primate vision an ideal starting point for a biology-based quantitative theory. The visual cortex serves as a window into the rest of the cerebral cortex; it also offers a glimpse into the world of large and complex dynamical systems.

A Dynamical Model of LGN → V1

Let us focus on a small piece of the action that occurs between the retina and the input layer of the primary visual cortex (V1). Between the retina and V1 lies a single structure called the lateral geniculate nucleus (LGN) (see Figure 1). LGN cells, which do not interact among themselves, relay signals from the retina to V1. There is a natural correspondence called the retinotopic map between a monkey's retina or LGN and its visual field. Via this correspondence, an LGN cell at location x_0 responds roughly linearly to visual input functions of the form

$$I_\pm(t) = \left[I_B \pm C\int_0^\infty\int_{-\infty}^\infty K(s) A(x_0 - x)L(x,t-s)\,dx\,ds\right]^+ . \qquad (1)$$

Here, $L(x,t)$ measures light intensity at time t and location x. The spatial kernel $A(x)$ is a Gaussian-like function that describes the receptive field (RF) of the cell, $K(s)$ is the temporal kernel, and I_B and C are constants. The RF of each LGN cell covers only a fraction of a degree; together, these small RFs tile the visual field. The temporal kernel $K(s)$ takes the shape of a sine function, so convolving with it is akin to taking a time derivative in the luminance of the cell's RF; i.e., LGN cells detect changes in luminance. The $\pm$ in (1) describes two kinds of LGN cells: ON cells that spike when their RF changes from dark to light, and OFF cells that do the opposite.

V1 is a much larger and more complicated network of excitatory (E) and inhibitory (I) neurons, the dynamics of which are described by the leaky integrate-and-fire equation:

See **Dynamical System** *on page 2*

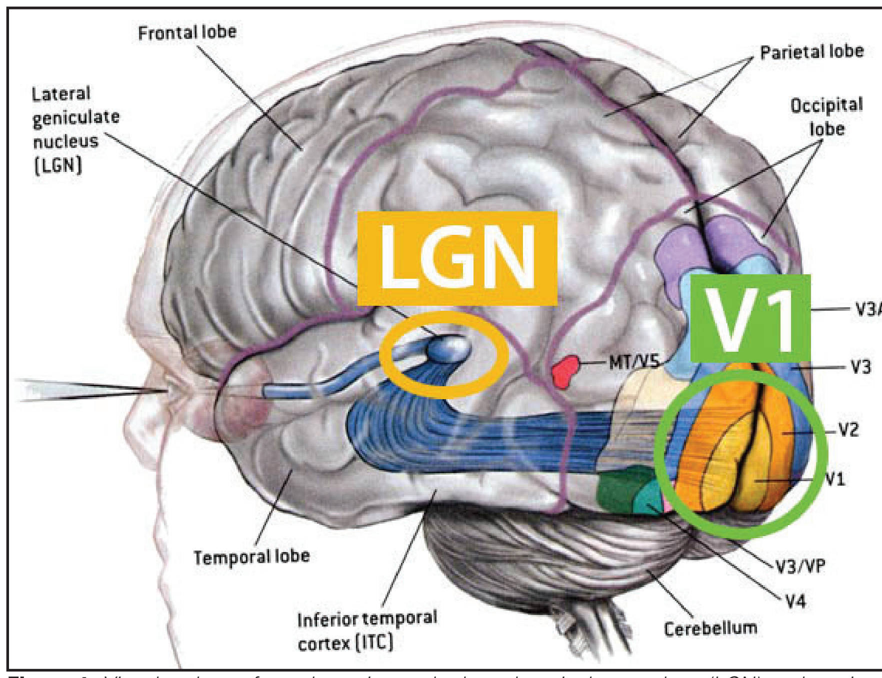


Figure 1. *Visual pathway from the retina to the lateral geniculate nucleus (LGN) to the primary visual cortex (V1). Figure adapted by the author.*

Nonprofit Org
U.S. Postage
PAID
Permit No 360
Bellmawr, NJ

A Deeper Look at Life Within the Army and Air Force Research Laboratories

By Lina Sorg

Many career opportunities in science, technology, engineering, and mathematics (STEM) are available within the U.S. Department of Defense (DoD): the largest federal government agency in the country. The three military departments that comprise DoD—the Departments of the Army, Navy, and Air Force—all employ applied mathematicians and computational scientists in various roles and capacities. These departments are further subdivided into numerous research laboratories and directorates that are dedicated to cutting-edge scientific discovery and technological innovation at the interfaces between traditional research fields.

To get a sense of mathematics-based career paths within specific sections of DoD, *SIAM News* connected with Daniel Eckhardt of the Air Force Research Laboratory[1] (AFRL) and Robert Martin of the U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory[2] (ARL). They spoke about the laboratories' objectives and focal points, discussed their individual roles and projects, and offered guidance for early-career researchers and SIAM members who are interested in governmental work in DoD settings.

***SIAM News*: Tell us a little bit about the respective missions and foci of ARL and AFRL.**

Robert Martin: Both the Army and Air Force research labs invest significantly in the basic research end of the spectrum — they focus on very early concepts. I want to emphasize this dedication to basic research; the product is forward-looking research that can be published openly, and you really get to engage in the work. I think that inability to publish is a common misconception about DoD, but we're really looking for exceptional open research and the best ideas to drive the frontiers of science.

I was formerly at AFRL with Dan and recently switched to DEVCOM ARL in September 2021, where I work in the office that funds extramural fundamental research.[3] Our job is to develop the basic research ecosystem in the U.S. and internationally. Even beyond ARL, we work indirectly with the supporting industries that feed up into ARL to develop the workforce.

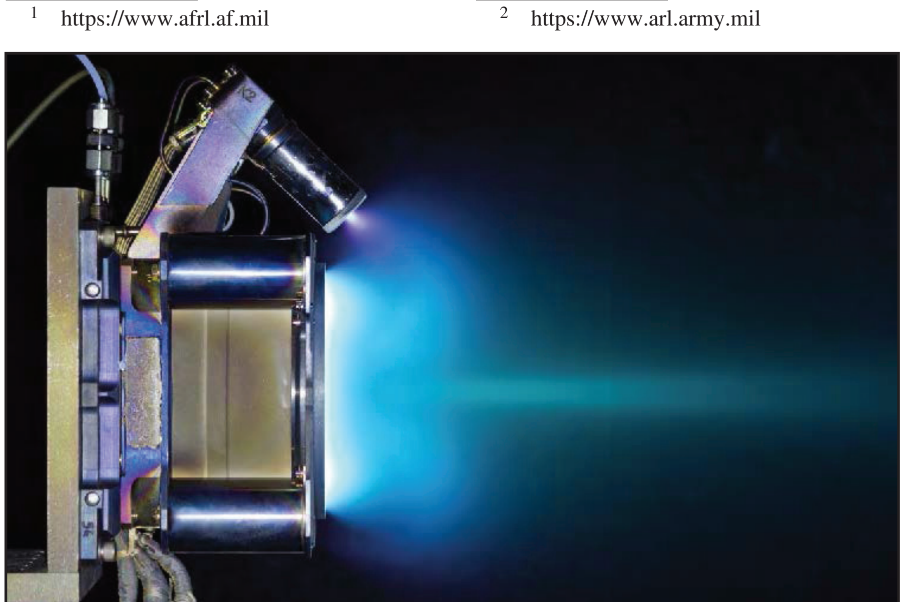
Daniel Eckhardt: AFRL's mission is to lead the discovery and development of technologies and technology solutions for the U.S. Department of the Air Force. The basic research that we're interested in occurs on 10-year- and 30-year-horizon types of timescales. AFRL is split into multiple different technical directorates; there's a Space Vehicles Directorate, Directed Energy Directorate, Materials and Manufacturing Directorate, and so forth. Like Rob said, you're not going to face publication restrictions. And with the creation of Space Force several years ago, AFRL now promotes a "one lab, two services" idea, meaning that the breadth of research topics is very extensive. Each service typically has its own lab, but you'll find both Air Force and Space Force in AFRL.

SN: Can you explain the concept of Space Force?

DE: Right now, everything space is very exciting. Space Force aims to make space resilient, democratize space access, and certify that our commercial and allied partners can safely access space. I work in the rocket propulsion division—which is part of the Aerospace Systems Directorate—and we develop models for everything rocket-related, from engines to in-space propulsion. We also develop and test hardware. I concentrate on thrusters, but other people work on sensors, the environment, space weather, and plasma physics.

Being part of something new and watching it grow is pretty awesome. The neat thing

See **Army and Air Force** *on page 3*



A SPT-100 Hall-effect thruster in the Air Force Research Laboratory's In-Space Propulsion Branch. Figure courtesy of [1].

1 https://www.afrl.af.mil

2 https://www.arl.army.mil

3 https://www.arl.army.mil/business/broad-agency-announcements

siam®
SOCIETY for INDUSTRIAL and APPLIED MATHEMATICS
3600 Market Street, 6th Floor
Philadelphia, PA 19104-2688 USA

siam news
Volume 55/ Issue 6/ July/August 2022

Dynamical System

Continued from page 1

$$\frac{dv}{dt} = -\frac{1}{\tau}v - g_E(t)(v - V^E) - g_I(t)(v - V^I). \quad (2)$$

Here, v is the membrane potential of an E or I neuron. In normalized units, when v reaches 1, the neuron fires a spike and v resets to 0. The right side of (2) describes the forces that act on v: the first item is a leak term and the second and third terms are the E and I currents that run into the neuron. The E current is raised when the neuron receives a spike from either an LGN cell or another E cell; the I current, which has the opposite sign, is elevated in magnitude when the neuron receives an I spike. Each neuron has hundreds of presynaptic E and I cells, the spikes of which produce fluctuating current inputs that cause the dynamics of v to resemble a random walk with an upward drift. The postsynaptic neuron fires when v reaches 1, setting off similar changes in other neurons. More details are available in [1].

Detecting Edges and Tracking Moving Targets

We will now focus on two of the most basic visual capabilities in primates: orientation selectivity (OS) and direction selectivity (DS). OS refers to the fact that each V1 neuron has a preferred orientation. When a vertical-preferring neuron detects a vertical edge in its RF, it is excited and fires vigorously; the further the edge is from the neuron's preferred orientation, the weaker the response. Since the RFs of neurons that prefer the full range of orientations cover each spatial location, V1's spiking activity allows the brain to deduce the contours of objects.

Neurons also respond more vigorously to motion in specific directions. This capability, called DS, is implicated in *pursuit eye movements*. Even though the human visual field covers a wide angle, visual acuity is excellent in only a small region of two to three degrees at the center. To track a moving object, the brain must direct the eyes to the target; DS is crucial in this computation.

V1 neurons exhibit both OS and DS, yet LGN cells, which provide the sole source of feedforward input to V1, curiously possess neither capability. The remainder of this article will focus on the origins of OS and DS, which are of fundamental importance in theoretical neuroscience.

The wiring between the LGN and V1 is key to OS. David H. Hubel and Torsten N. Wiesel, who won the 1981 Nobel Prize in Physiology or Medicine[1] for early experimental work that greatly advanced the field of visual neuroscience, hypothesized that each V1 cell is connected to two or three parallel rows of ON and OFF LGN cells [5]. Figure 2 illustrates OS in action. Here we assume that the three ON and three OFF LGN cells are afferent to a V1 cell. Recall that an ON cell is excited when the part of a drifting grating that passes over its RF changes from dark to light, while the opposite is true for OFF cells. When the grating and LGN align (as in the upper portion of Figure 2), all six LGN cells are simultaneously excited for half of the cycle and quiet the other half; this elicits a strong response from the V1 cell to which the LGN cells project. On the other hand, an orthogonal grating (as in the lower portion of Figure 2) excites half of the LGN cells at any given time, producing more evenly spaced LGN spikes that elicit a weaker response from V1. Notice that V1 response is driven by LGN spiking patterns, not their firing rates; the firing rates of LGN cells—which have no OS—are independent of grating orientation. Mathematically, spikes that arrive in rapid succession leave less time for the leak term in (2) to act and can thus more effectively drive v across the threshold.

Mechanistic origins of DS in the macaque V1 eluded the neuroscience community for more than half a century. My collaborators and I recently proposed a biologically plausible explanation, an idealized version of which can be made rigorous [2, 3]. In addition to the usual analytical thinking, the challenge here also involved learning to distill relevant facts from anatomical and physiological data and to operate in the world of biology, where complexity is high and information is incomplete. Our findings are supported by simulations that use LGN→V1 models of the kind described here and are in quantitative agreement with experimental data.

Though individual LGN cells do not have DS, we found that the summed response of LGN cells that are afferent to a V1 cell can have DS. For simplicity, consider one ON and one OFF cell (see Figure 3b). A sinusoidal signal produces a phase difference $\delta\phi$ between the responses of the two cells because of their spatial displacement. When $\delta\phi$ is small and the responses are in-phase, they amplify each other; when they are anti-phase, they cancel. The question is hence as follows: Why is $\delta\phi$ different for signals from the left versus the right? Which biological properties are responsible for breaking this left-right symmetry?

We traced the answer to differences in the temporal kernels of ON and OFF cells (K_{ON} and K_{OFF}, respectively). Experiments show that K_{ON} is delayed and takes the form $K_{\mathrm{ON}} = K_{\mathrm{OFF}} +$ a positive function (see Figure 3a). Figure 3b demonstrates the way in which a delay breaks left-right symmetry. Delays, however, are only effective for signals with relatively high temporal frequencies (TF). At lower TFs, directional bias can be explained heuristically: convolving a sine function with K_{OFF} is like taking its derivative (which gives cosine), whereas convolving with a positive function is a form of averaging (producing another sine function). The addition of a sine function to a cosine shifts its phase.

Working at the interface of mathematics and neuroscience has convinced me that a partnership between the two subjects can be fruitful; I hope to have conveyed a sense of that potential in this short article.

[1] https://www.nobelprize.org/prizes/medicine/1981/summary

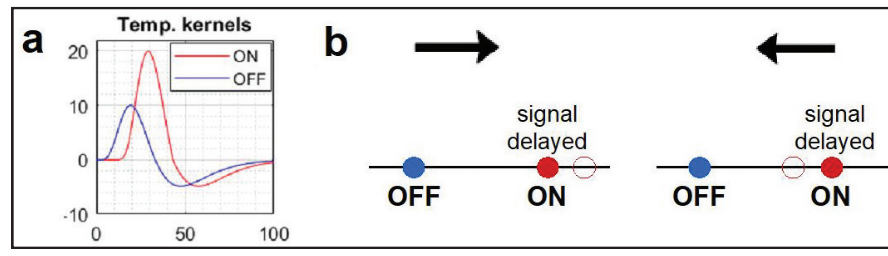


Figure 3. *Ingredients of direction selectivity (DS).* **3a.** *Temporal kernels of ON and OFF lateral geniculate nucleus (LGN) cells.* **3b.** *DS due to delayed arrival of the ON signal. The black arrow indicates the signal's direction and the solid red and blue dots represent ON and OFF cells. A delayed ON signal shifts the ON cell's effective location from the solid red circle to the open red circle, thus breaking the left-right symmetry. Figure 3a courtesy of [2], 3b courtesy of the author.*

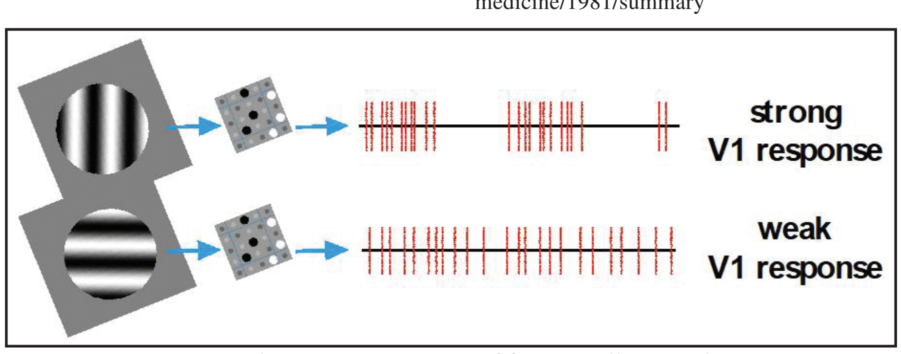


Figure 2. *The workings of orientation sensitivity (OS). Two different drifting gratings on the left—one aligned with the ON-OFF configuration in the group of six lateral geniculate nucleus (LGN) cells (top) and the other orthogonal to it (bottom)—elicit different LGN spike patterns (in red) that lead to very different visual cortex (V1) responses. Figure courtesy of the author.*

References

[1] Chariker, L., Shapley, R., & Young, L.-S. (2016). Orientation selectivity from very sparse LGN inputs in a comprehensive model of macaque V1 cortex. *J. Neurosci.*, *36*(49), 12368-12384.

[2] Chariker, L., Shapley, R., Hawken, M., & Young, L.-S. (2021). A theory of direction selectivity for macaque primary visual cortex. *Proc. Natl. Acad. Sci.*, *118*(32), e2105062118.

[3] Chariker, L., Shapley, R., Hawken, M., & Young, L.-S. (2022). A computational model of direction selectivity in macaque V1 cortex based on dynamic differences between on and off pathways. *J. Neurosci.*, *42*(16), 3365-3380.

[4] Hubel, D.H. (1995). *Eye, brain, and vision*. In *Scientific American Library*. New York, NY: W.H. Freeman.

[5] Hubel, D.H., & Wiesel, T.N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.*, *160*(1), 106-154.

Lai-Sang Young is a professor of mathematics at the Courant Institute of Mathematical Sciences and a Henry and Lucy Moses Professor of Science at New York University. She is a member of the National Academy of Sciences and received SIAM's Jürgen Moser Lecture in 2021.

siam news

ISSN 1557–9573. Copyright 2022, all rights reserved, by the Society for Industrial and Applied Mathematics, SIAM, 3600 Market Street, 6th Floor, Philadelphia, PA 19104-2688; (215) 382-9800; siam@siam.org. To be published 10 times in 2021: January/February, March, April, May, June, July/August, September, October, November, and December. The material published herein is not endorsed by SIAM, nor is it intended to reflect SIAM's opinion. The editors reserve the right to select and edit all material submitted for publication.

Advertisers: For display advertising rates and information, contact Kristin O'Neill at marketing@siam.org.

One-year subscription (nonmembers): Electronic-only subscription is free. $73.00 subscription rate worldwide for print copies. SIAM members and subscribers should allow eight weeks for an address change to be effected. Change of address notice should include old and new addresses with zip codes. Please request address change only if it will last six months or more.

Improving Earth System Predictability with Artificial Intelligence

By Sven Leyffer and Salil Mahajan

The U.S. Department of Energy (DOE) recently organized a series of workshops to identify challenges and research opportunities for the use of artificial intelligence (AI) and machine learning (ML) within Earth system models (ESMs). In October 2021, the community-led, multi-lab Artificial Intelligence for Earth System Predictability (AI4ESP) initiative[1] hosted a multitude of interactive sessions that addressed all aspects of ESMs, as well as ML, uncertainty quantification, and computational science. More than 450 attendees with diverse backgrounds—including climate scientists, mathematicians, statisticians, and computer scientists—discussed over 150 white papers[2] during this time. Here we summarize our main impressions of the AI4ESP workshop; these reflections precede a forthcoming formal report from the workshop organizers and DOE's Office of Science.

[1] https://ai4esp.org
[2] https://www.ai4esp.org/white-papers

A common theme within every presentation was the smart use of AI and ML tools in ESMs as surrogates for localized, small-scale, complex physical processes in order to drive higher-fidelity predictions within the overall system. Two points of understanding underlie this approach: (i) ESMs are multiscale models that couple complex phenomena at many scales, and (ii) AI/ML models—while likely incapable of replacing years of climate model research—can help integrate more small-scale phenomena into existing models. Here we briefly outline several examples of ways in which AI and ML are currently impacting ESMs.

Bridging Scales with AI

During the AI4ESP workshop, Tapio Schneider (California Institute of Technology) discussed a scale-bridging approach for AI-accelerated Earth system predictability (ESP). Complex climate models face multiscale challenges in which large-scale variables impact small-scale biophysical effects and vice versa. Cloud models—wherein humidity and temperature serve as the large-scale variables that affect cloud formation—are one such example. Microphysical effects and turbulent dynamics influence droplet growth in clouds, ultimately giving rise to large-scale results such as cloud albedo, cloud cover, and precipitation. Because the use of brute-force computation to resolve such small-scale phenomena on a global scale is impractical, new approaches are necessary (see Figure 1, on page 4). Similar challenges also arise in biosphere and ocean models.

To meet these challenges, the Climate Modeling Alliance[3] (CliMA) is building a new ESM that combines deep learning with reductional science to overcome each subject's respective shortcomings.[4] In the context of cloud modeling, this method involves coarse graining the fluid equations via conditional averaging; doing so yields exact conservation laws with closure functions that can be learned from multiple data sources. CliMA's model is motivated by a three-pronged approach that advances theory to exploit parametric sparsity, harnesses diverse data (including data from physics-based Bayesian emulators), and leverages compute resources like graphics processing units for local high-resolution models.

[3] https://clima.caltech.edu

AI for Subgrid Parameterization

Pierre Gentine (Columbia University) described the application of AI algorithms

See **Earth System** *on page 4*

[4] https://clima.caltech.edu/2018/03/08/earth-system-modeling-2-0

Army and Air Force

Continued from page 1

about Space Force is that it's aiming to be a fully digital service while remaining as lean as possible when compared to more traditional services. Because much of the work for Space Force takes place behind a terminal or in a lab rather than in the field, AFRL is emphasizing a variety of STEM disciplines and trying to hire a lot of technical people.

SN: What kinds of projects are you currently working on?

DE: My main job is electric in-space propulsion, so we generate thrusters for different spacecraft. This work has an applied component where we test actual flight hardware, as well as a research component where we envision the types of space assets that DoD will need in 30 years. Then we begin to create the technology, since technology development takes a very long time.

When you build a spacecraft and send it to space, you can't go up and fix it like you would a car if something breaks. Much of our work thus involves developing techniques that can help us understand rare events and emergent behaviors, which is very fun.

RM: I started out with high-fidelity plasma physics modeling at AFRL, so I was doing computational mathematics directly. After I switched to ARL, the application changed significantly but the underlying mathematics remained the same. I'm now a program manager for the Modeling of Complex Systems Program, which is more application-agnostic — it can apply to social networks, pandemics, or mechanical systems. Our math branch also includes programs in biomathematics and computational mathematics, so I differentiate the modeling of complex systems from other math programs with an emphasis on inverse and outer-loop problems. I'm still interested in verification and validation as well, especially model validation for rare events when the data are sparse.

SN: What does a typical workday look like for each of you?

RM: I was actually the first remote program manager for the extramural research program at ARL; I'm in Milwaukee, Wis., and most of the other program managers are in North Carolina. A big piece of the program management role involves interacting with principal investigators (PIs). I spend most of the week receiving and reviewing white papers and grants, then I talk with the proposers about how their ideas fit within my program — or try to route them to some of ARL's other programs. There is also a lot of coordinating the program within the bigger picture to support the future Army.

DE: If I'm onsite, I check in with my team about whatever we're working on and discuss future steps. I do work remotely some days, but I usually go into the lab because I run an experimental group that needs to be physically present to actually perform the experiments. We typically meet with the modeling and simulation folks—who are more remote than we are—and information dump to assess progress.

In terms of research projects, we either work alone or in small teams of two to three people. Our basic research operates very similarly to the way that academics run their research at a university: we write a proposal, get funding for it, and work on it until the next cycle comes around.

SN: Have you experienced a shift in work or funding priorities over the last several years?

DE: Because basic research is mostly forward looking, the priorities don't change that drastically across administrations and leadership. But with space becoming very interesting these days, there's been a priority increase on space research and a push to involve more of the domestic community with space-related work. Only a handful of universities pursue research that is relevant to space efforts, so DoD is attempting to democratize this endeavor and make sure that it extends across the entire nation. In general, DoD is working to tap into a broad cross-section of the nation's research talent with initiatives such as Defense Established Programs to Stimulate Competitive Research[4] (DEPSCoR). In addition, AFRL is reaching out to Historically Black Colleges and Universities (HBCUs) and Minority Serving Institutions (MSIs) to broaden the country's tech base. Good things come out of expanding beyond the traditional work base, including a diverse workplace and diversity of ideas and people.

RM: Dan mentioned DEPSCoR, which offers a separate line of funding from the broader DoD and targets states and institutions with fewer research projects. In addition, ARL's extramural research programs emphasize HBCUs and MSIs to engage them with our ecosystem.

SN: What advice would you offer to early-career mathematicians who hope to pursue government positions?

DE: It's funny because I'm actually coded as an aerospace engineer, but I got my degree in mathematics. And Rob is a mathematician, but he got his degree in engineering. So my advice is not to look at a position's title and instead look at what you bring to a particular opportunity. I got into AFRL through the National Research Council's Research Associateship Programs,[5] which are a great way for fresh,

[4] https://basicresearch.defense.gov/Pilots/DEPSCoR-Defense-Established-Program-to-Stimulate-Competitive-Research
[5] https://sites.nationalacademies.org/PGA/RAP/index.htm

Site visit with the summer undergraduate mathematics interns who participated in an Air Force Research Laboratory (AFRL)-sponsored project for the Institute for Pure and Applied Mathematics' (IPAM) 2019 Research in Industrial Projects for Students (RIPS) program. From left to right: Daniel Eckhardt, Abhishek Shivkumar, Brianna Fitzpatrick, Becks Lopez, Mykhaylo Malakhov, and Robert Martin. Photo courtesy of the 2019 IPAM RIPS AFRL team.

young Ph.D.s to experience the type of work that government labs do. I came out here, fell in love with it, and the rest is history.

RM: I think Dan nailed it with the research associateship programs, which vary by lab; for instance, ARL's Research Associateship Program[6] is run by Oak Ridge Associated Universities. Some even include opportunities for junior- and senior-level researchers, in addition to postdoctoral positions. Going to research conferences in your field and interacting with folks from the labs is also a great networking opportunity. I got my first job at AFRL because I went to one of these conferences and met my future boss there. ARL even sponsors some research conferences that unite people from different communities and advance career development.

On the extramural side, the Small Business Innovation Research and Small Business Technology Transfer programs[7] encourage small business to partake in federal research and development. All government agencies have them, and interesting opportunities arise through those channels.

SN: How can SIAM members explore possible career paths or opportunities for involvement with ARL and AFRL?

DE: AFRL runs internships every summer via its Scholars Program.[8] We also have the Summer Faculty Fellowship Program,[9] where early-career faculty can apply for a grant and spend a summer working with Air Force researchers. In addition, AFRL maintains educational research partnerships with universities and directly funds research or visiting faculty members.

[6] https://www.orau.org/arlfellowship
[7] https://www.sbir.gov/about
[8] https://afrlscholars.usra.edu/scholarsprogram
[9] https://afsffp.sysplus.com

There are also program reviews. A list of all Air Force Office of Scientific Research programs[10] is available on the All Partners Access Network,[11] and you don't have to be a PI to attend program reviews — you just have to register or request an invitation. You can meet the program officers who give grants to universities and see if the type of research in their portfolios fits with what you're doing.

RM: When I was in Los Angeles, I attended a lot of Institute for Pure and Applied Mathematics workshops[12] at the University of California, Los Angeles. Now I virtually attend some events by the Institute for Mathematical and Statistical Innovation, which runs long programs[13] twice a year with really interesting topics. I also often go to American Physical Society conferences, and SIAM conferences of course. DoD folks are present at all of these events, so they are good opportunities for engagement.

Even just reading published papers and looking at the coauthors can be helpful. If you're passionate about an article topic and see that one of the coauthors is from a DoD lab, reaching out to them is a great way to make connections.

References

[1] Nakles, M., Holmes, M., & Hargus Jr., W. (2015). *An investigation into the spectral imaging of Hall thruster plumes*. Paper presented at the 34th international electric propulsion conference (IEPC-2015-416). Kobe, Japan.

Lina Sorg is the managing editor of SIAM News.

[10] https://community.apan.org/wg/afosr
[11] https://www.apan.org
[12] http://www.ipam.ucla.edu/programs/workshops
[13] https://www.imsi.institute/programs

Earth System

Continued from page 3

to learn parameterization of subgrid-scale processes in the atmosphere—such as turbulence and convection—that are present in conventional coarse-grained climate models with approximately 100 kilometers of horizontal resolution. These models come from high-resolution models that resolve processes at a finer scale (e.g., cloud-resolving models). The new data-driven parameterizations replace the traditional empirical parameterizations in coarse-grained models, reduce the biases in these models, and are computationally cheaper. However, instantaneous mass and energy conservation in these types of data-driven approaches continues to be difficult. Solving this challenge will ensure that climate change simulations remain generalizable under different scenarios and initial conditions.

Process-based Models in ML

Chaopeng Shen (Pennsylvania State University) highlighted advances in the field of hydrology, wherein data-driven approaches like long short-term memory models outperform traditional dynamical models. These approaches have successfully achieved multi-year forecasting of numerous factors, including soil moisture, stream flow runoff, tracer transport, and so forth. However, the challenges that accompany the interpretation of ML models persist.

Scientists have recently begun to incorporate process-based models (PBMs) into the ML training framework. Doing so requires the use of differentiable PBMs—which typically necessitates reimplementation of the PBM or its replacement with a differentiable surrogate—in order to leverage existing gradient-based training techniques, such as backpropagation. Data-driven approaches also show promise in parameter calibration of hydrology models via neural networks — a practice that improves generalizability. They can even help extract information from big data to improve/modify process representation in physical models.

Learning Equations From Data

Ocean models and cloud models face similar multiscale challenges; traditional closure models that represent subgrid turbulence struggle to accurately predict climate effects. Laure Zanna[5] (New York University) presented an approach that learns subgrid closure models from data, including detailed simulations of small regions to predict subgrid forcing terms. This technique can also identify physical quantities and equations from a set of basis functions that are more interpretable than neural networks and better capture energy and momentum transfer between scales. It seeks to learn new physics from data, test the robustness of subgrid models, enable validation and verification, and accurately quantify uncertainties.

[5] https://sinews.siam.org/Details-Page/machine-learning-for-multiscale-systems-from-turbulence-to-climate-prediction

Applied Math Challenges

In addition to charting the progress of AI and ML within ESP, the AI4ESP workshop also identified a number of applied math challenges in areas like knowledge-informed ML. This field integrates physical laws into the learning process and can even design specialized network architectures that automatically satisfy physical invariants [2]. More generally, researchers can include constraints through differentiable optimization, which adds a convex optimization layer to neural networks [1]. Remaining challenges include the learning of emerging constraints, identification of erroneous constraints, and inclusion of state-of-the-art optimization methods. Future research could also involve the development of explainable AI to provide physical insight or causal relationships for the system in question, as well as the identification and classification of rare events such as wildfires and heat waves.

References

[1] Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., & Kolter, J.Z. (2019). Differentiable convex optimization layers. In *Advances in neural information processing systems 32 (NeurIPS 2019)*. Vancouver, Canada.

[2] Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nat. Rev. Phys.*, *3*, 422-440.

[3] Schneider, T., Jeevanjee, N., & Socolow, R. (2021). Accelerating progress in climate science. *Phys. Today*, *74*(6), 44.

[4] Schneider, T., Lan, S., Stuart, A., & Teixeira, J. (2017). Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophy. Res. Lett.*, *44*(24), 12,396-12,417.

Sven Leyffer is a senior computational mathematician in the Mathematics and Computer Science Division at Argonne National Laboratory. He works on nonlinear optimization, currently serves as secretary of the International Council for Industrial and Applied Mathematics, and is the president-elect of SIAM. Salil Mahajan is a staff scientist at Oak Ridge National Laboratory. His research interests lie in climate dynamics and variability, climate statistics and machine learning applications to climate modeling, and analysis and predictability.

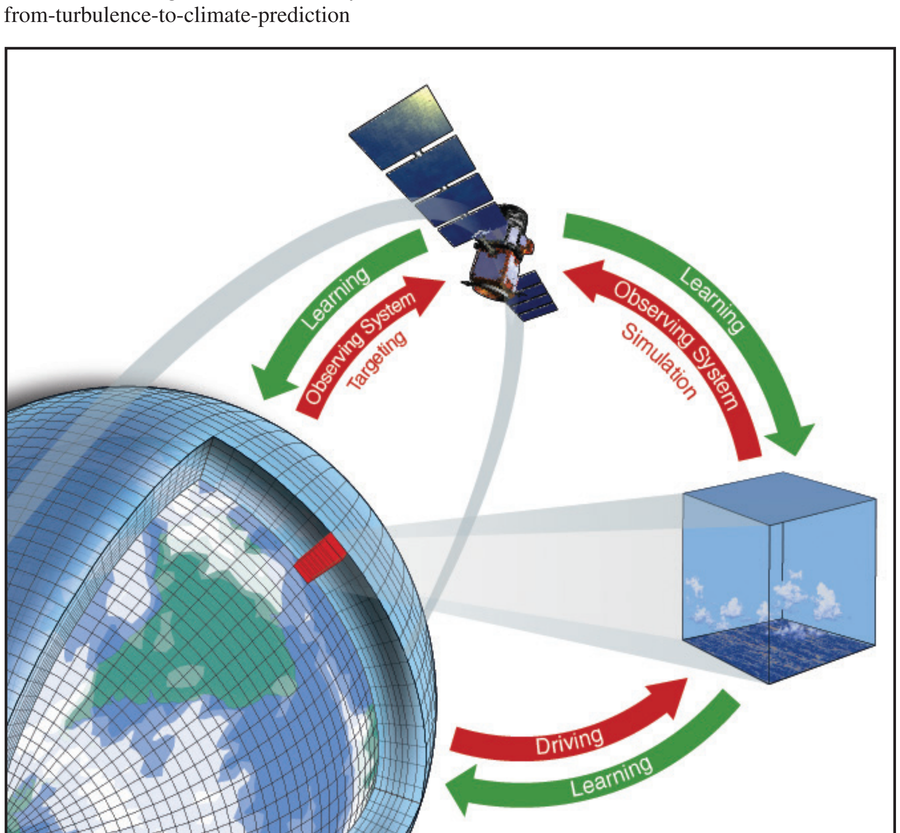


Figure 1. *A schematic of an Earth system modeling framework that integrates global observing systems and targeted high-resolution simulations. Figure courtesy of Tapio Schneider [3, 4].*

Rapid Prototyping with Julia: From Mathematics to Fast Code

By Michel Schanen, Valentin Churavy, Youngdae Kim, and Mihai Anitescu

Software development—a dominant expenditure for scientific projects—is often limited by technical programming challenges, not mathematical insight. Here we share our experience with the Julia programming language[1] in the context of the U.S. Department of Energy's Exascale Computing Project[2] (ECP) as part of ExaSGD,[3] a power grid optimization application. Julia is a free and open-source language that has the potential for C-like performance while extending the expressivity of MATLAB [5]. It is meant for scientific computing and has helped us effectively collaborate as a team of mathematicians and software engineers. Our ratio of mathematical reasoning to software development tasks has vastly improved as a result, empowering us to do more mathematics per unit of code.

ECP is an aggressive research, development, and deployment project that focuses on the delivery of mission-critical applications, an integrated software stack, and exascale hardware technology advances. It partners with leadership computing facilities at the Department of Energy, including Oak Ridge Leadership Computing Facility and Argonne Leadership Computing Facility. These facilities employ heterogeneous architectures that feature graphics processing units[4] (GPUs) in their recent[5] and imminent exascale-barrier-breaking systems, Frontier[6] and Aurora.[7]

We opted to use Julia in our work to handle the complexity of GPUs: hardware accelerators that are crucial for ECP's ambitious goals yet increasingly available as commodity hardware for scientists. The appearance of GPUs obliterated our original optimization algorithm plans for ECP; our algorithms relied on direct sparse indefinite linear solvers, which had little to no support on GPUs for our use case at the time. We thus found ourselves needing to redesign and reimplement many of our core functions — a process that allowed us to contemplate radically different approaches.

Before GPUs entered the scene, we utilized Julia's algebraic modeling package JuMP[8] as a modeling language [3]. We quickly realized that portability and modeling are two sides of the same coin, and found Julia to be a fast tool for prototyping novel optimization algorithms and models on GPUs. We developed and published various packages in a timeframe that would have been unimaginable for our team's previous projects.[9]

As a result of Julia's built-in features, our packages are now available to a wider audience and follow best practices in software engineering, such as unit testing and documentation. They are also capable of running on any machine, from laptops to supercomputers like Summit[10] at Oak Ridge National Laboratory. We recently shared our adventure diary in an article for the SIAM Activity Group on Optimization[11] that details a possible generic template for how Julia might help implement newly-developed mathematical methods for a plethora of high-performance computing architectures in record time [1].

Now we will describe the core features of Julia that enable fast pathways from mathematics to applications in a nearly hardware-agnostic fashion.

SOFTWARE AND PROGRAMMING

Symbolic Calculus and Metaprogramming

Calculus represents the core of numerical algorithms. It is largely restricted to function compositions $g \circ f$ in mathematics, which directly map to function compositions in programming. However, these functions are a black box after compilation; f and g become input-output functions, and their expressions are lost at runtime. For example, the application of symbolic derivative calculus to a function f becomes difficult to implement. Metaprogramming precisely addresses this lack of access to the expressions of f and g. This programming technique allows a program A to take another program B's *expressions* as an input (instead of B's output), then semantically transforms program B into new expressions for program C. As such, interfaces are not reduced to mere exchanges of data. Instead, they acquire an emergent property of creating a program C through expression transformation of program A (transformer) on program B (transformed):

```
mul(a,b) = a*b
∇mul(a,da,b,db) = (a*b),
    (a*db + da*b)
```

For instance, program `mul` that implements `a*b` transforms into program `∇mul` that implements its derivative via an automatic differentiation tool. Recently, the "differentiable programming" technique emerged to describe the capability of such a differentiation workflow. Similarly, more complex expression transformations appear in uncertainty quantification, differential equations, performance profiling, and debugging — nearly everywhere in scientific computing.

These expression transformations often hijack programming language features like operator overloading and C++ templates, or they are compiler integrated. We believe that three of Julia's design features make it uniquely positioned to tackle such classes of metaprogramming problems:

- Macros (functions over expressions)
- Multiple dispatch and specialization
- Just-in-time (JIT) compilation through the LLVM compiler.[12]

Consequently, the process from compilation to a binary and eventually an executable is no longer sequential. The compilation and execution steps are intertwined, thus making compilation a part of the runtime.

1 https://julialang.org
2 https://www.exascaleproject.org
3 https://www.exascaleproject.org/research-project/exasgd
4 https://bssw.io/blog_posts/a-gentle-introduction-to-gpu-programming
5 https://www.hpcwire.com/2022/05/30/top500-exascale-is-officially-here-with-debut-of-frontier
6 https://www.olcf.ornl.gov/frontier
7 https://www.alcf.anl.gov/aurora
8 https://jump.dev
9 https://exanauts.github.io

10 https://www.olcf.ornl.gov/summit
11 https://www.siam.org/membership/activity-groups/detail/optimization
12 https://llvm.org

```
using ChainRules
function ChainRules.rrule(::typeof(Base.reduce), op, x::AbstractArray)
    y = reduce(op,x)
    function reduce_pullback(dy)
        dx = ChainRules.@thunk((dy.*y) ./ x)
        return ChainRules.NoTangent(), ChainRules.NoTangent(), dx
    end
    return reduce(op,x), reduce_pullback
end
```

Figure 1. *Custom differentiation rule for* `reduce`.

	CUDA-C (s)	Julia/CUDA.jl (s)	Ratio
1354pegase	0.62	0.97	1.56
2869pegase	1.87	4.39	2.34
9241pegase	6.70	12.66	1.88
13659pegase	8.12	14.58	1.79

Figure 2. *Runtime comparison of ExaTron written in C/CUDA versus Julia/CUDA.jl.*

This dynamic compilation is both Julia's biggest strength and one of its weaknesses. LLVM's original design did not support a JIT language, so the time-to-first-plot is an ongoing challenge in Julia. To reduce the compilation time, users must be aware of both type stability and the language's various intricacies. Ongoing work focuses on caching compiled code and making this issue more transparent and user-friendly.

On the upside, macros enable the manipulation of expressions; multiple dispatch allows an algorithm to apply other algorithms based on their type, then specialize upon the inputs; and JIT compilation creates optimized programs that can take advantage of modern hardware. Instead of addressing each feature from a theoretical perspective, we illustrate the way in which they jointly facilitate a compact and transparent code that remains portable and differentiable.

Portability and Differentiability

As an example, we implement the function `speelpenning` $y = \prod_{i=1}^{n} x_i$ [6]:

```
speelpenning(x) = reduce(*,x)
```

This function is mapped to the Julia internal reduction `reduce` with the product operator `*`. So far, `x` has no type. Initializing `x` to a vector of type `T=Vector{Float64}` yields the following:

```
x = [i/(1.0+i) for i in 1:n]
    # Vector{Float64}
speelpenning(x)
    # 0.090909...
```

The function `speelpenning` is then *dispatched* on this type `T`, calling the appropriate and optimized reduction and culminating in the compiled binary machine code. We compute the gradient of `speelpenning` with Zygote,[13] an automatic differentiation tool in Julia [4]:

```
using Zygote
g = speelpenning'(x)
```

The Zygote operator `'` transforms `speelpenning` into its gradient computation via access to the LLVM intermediate representation and based on differentiation rules that are defined in the package ChainRules.jl.[14] All of this code is only compiled once we call `speelpenning'`.

Suppose that we decide to run our code on a NVIDIA GPU. CUDA.jl[15]—the package for writing CUDA kernels in Julia [2]—uses the broadcast operator `'.'` to naturally vectorize statements like `a .= b .+ c`, where `a`, `b`, and `c` are vectors. The entire statement is effectively fused into one kernel, so let's try to execute our differentiated `speelpenning` on a NVIDIA GPU. To do so, we move vector `x` onto the GPU via the `CuArray` constructor[16] and call the gradient computation. The entire machinery now compiles the code with this new data type and dispatches onto the GPU. However, executing this code in its current state fails:

```
cux = CuArray(x)
cug = speelpenning'(cux)
```

Why does it fail? No differentiated function is defined for `reduce(*,x)`. A reduction is a nontrivial operation that is difficult to efficiently parallelize, but we now know that we can compute the gradient with $\frac{\partial x_i}{\partial y} = y/x_i$ for $x_i \neq 0$. Use of the broadcast operator allows for a nicely parallel implementation `dx[i] .= y ./ x[i]`. Such custom derivatives are often necessary for differentiable programming, and their integration into the code requires detailed knowledge of the automatic differentiation tool internals. Assuming that our application's domain excludes $x = 0$, how can

See **Julia** *on page 8*

13 https://fluxml.ai/Zygote.jl/latest

14 https://zenodo.org/record/6371664/export/dcite4#.YppLfHbMJhE
15 https://github.com/JuliaGPU/CUDA.jl
16 https://github.com/JuliaGPU/CuArrays.jl

Biden Administration's New Budget Request Impacts Computational Science and Applied Mathematics Research

By Andrew Herrin

On March 28, President Biden released his second budget proposal to Congress. This proposal highlights new and continued priorities for federal agencies that advocate for applied mathematics and computational science. Over the last few months, the SIAM Committee on Science Policy[1] (CSP) has championed federal support for these research areas.

From April 4-6, the CSP and SIAM Science Policy Fellowship[2] recipients gathered in Washington, D.C., to discuss legislative and policy issues that can potentially impact applied math and computational science (in either a real or perceived manner). Over the course of two days, members of these groups interacted with speakers from Congress and federal agencies to address federal funding and policy initiatives. CSP members and Science Policy Fellowship recipients conducted both in-person and virtual meetings with congressional offices to advocate for robust funding to support applied mathematics and computational science research at the National Science Foundation (NSF), Department of Energy (DOE), Department of Defense (DOD), and National Institutes of Health (NIH). SIAM representatives also conversed with key federal agency leaders, including the following:

- David Manderscheid, Division Director of NSF's Division of Mathematical Sciences (DMS)
- Junping Wang, Deputy Division Director of DMS
- Barbara Helland, Associate Director of the Advanced Scientific Computing Research (ASCR) program[3] at DOE's Office of Science
- Cetin Kiris, Chief of the Computational Aerosciences Branch at the National Aeronautics and Space Administration's (NASA) Ames Research Center
- Nikunj Oza, Leader of the Data Sciences Group at NASA's Ames Research Center
- Carrie Wolinetz, Deputy Director for Health and Life Sciences at the White House Office of Science and Technology Policy.

The resulting discussions addressed agency budget plans for applied mathematics and computational science, and explored ways in which the SIAM community can engage with emerging initiatives.

Similar to last year, the Biden administration's fiscal year (FY) 2023 budget request[4] [1] proposed increases to non-defense discretionary spending at most science agencies, though these increases were more modest than in the FY 2022 request. While the 2023 proposal does increase funding for fundamental research, it focuses primarily on use-inspired research, translation, and technology development and deployment. The current budget request recommends a nearly six percent boost in discretionary spending from the FY 2022 enacted level, for a total of roughly $1.6 trillion. Consistent with the previous year, President Biden's top FY 2023 budget priorities are public health, climate and clean energy, manufacturing, innovation, and education.

While the request proposes increases across most federal agencies, the notable exception is a $2.4 billion cut to basic research, applied research, and advanced technology development accounts for the DOD's science and technology (S&T) programs. This reduction is consistent with a greater emphasis on the prototyping, deployment, and commercialization of technologies, rather than early-stage research and development. In addition, some nondefense research and development programs that fall outside of the administration's priority areas are slated for much less growth than in the FY 2022 request.

The Biden administration again proposes major increases across NSF for FY 2023 in both research and education priority areas as well as core NSF programs. Overall, Research and Related Activities would grow by 18 percent from the FY 2022 estimated level, and the Directorate for Education and Human Resources (EHR) would grow by 39 percent (though this figure becomes 20 percent if one accounts for the consolidation of all Graduate Research Fellowship Program[5] funding to EHR in FY 2022). The 2023 budget request recommends nearly $1.7 billion for the Directorate for Mathematical and Physical Sciences (MPS) and $259 million for DMS. Given the late process of finalizing FY 2022 appropriations, NSF has not yet released its final FY 2022 allocations; instead, it measures the proposed amounts against the FY 2021 levels that Congress enacted. Compared to FY 2021 levels, the 2023 request would provide an increase of $153 million (or 9.6 percent) for MPS and an increase of 16 million (or 6.5 percent) for DMS. If the request is enacted by Congress, the Office of Advanced Cyberinfrastructure[6] in the Directorate for Computer and Information Science and Engineering would see an increase of $22 million — 9.5 percent from FY 2021 levels.

The FY 2023 request proposes funding to fully establish the new Directorate for Technology, Innovation, and Partnerships[7] (TIP), which aims to advance science and engineering research and innovation, accelerate the translation of basic research, solve national and societal problems, and support educational pathways. Along with the existing NSF directorates, TIP would fund activities in priority areas such as climate and energy, advanced wireless research, biotechnology, microelectronics and semiconductors, advanced manufacturing, artificial intelligence (AI), and quantum computing. The request's emphasis on TIP, climate, clean energy, and equity aligns with congressional priorities and interest in boosting NSF's competitiveness. However, it remains to be seen whether appropriators will have the resources to deliver such major increases to NSF's budget. If not, tough choices may again be necessary to strike a balance between growing TIP and other priority areas versus protecting core programs.

The Biden administration's FY 2023 budget request would provide the DOE's ASCR program office with $1.07 billion — a $34 million (or 3 percent) increase from the FY 2022 enacted level. As the Exascale Computing Project[8] (ECP) moves towards completion, the appeal for ASCR re-emphasizes foundational research that will advance AI, quantum information science, and strategic computing initiatives while bolstering the competitive advantage of U.S. industry in terms of new

See **Budget Request** *on page 7*

[1] https://www.siam.org/about-siam/committees/committee-on-science-policy-csp
[2] https://www.siam.org/students-education/programs-initiatives/siam-science-policy-fellowship-program
[3] https://www.energy.gov/science/ascr/advanced-scientific-computing-research
[4] https://old.lewis-burke.com/sites/default/files/budget_update_-_lba_analysis_of_the_fy_2023_presidents_budget_request.pdf
[5] https://www.nsfgrfp.org
[6] https://www.nsf.gov/div/index.jsp?div=OAC
[7] https://beta.nsf.gov/tip/about-tip
[8] https://www.exascaleproject.org

Right Results via Wrong Arguments and Wrong Results via Right Arguments

MATHEMATICAL CURIOSITIES
By Mark Levi

In his very entertaining book [1], V. Arnold discusses the adiabatic invariance of the pendulum — a classical problem that Einstein addressed when quantum mechanics was being born. Einstein was motivated by the desire to explain the constancy of Planck's constant. Why does the energy-to-frequency ratio of photons remain unchanged, despite the buffeting of the emitting atom by surrounding fields? How does the atom "remember" this ratio?

In a (probably tentative) attempt at an explanation, Einstein pointed out a classical analog of this "memory" in the simple pendulum. He showed that the energy-to-frequency (E/ω) ratio for a linearized pendulum remains nearly unchanged if the string's length is *slowly* changed by a finite amount (e.g., by half).[1] The physical quantity E/ω has a geometrical meaning: it is the area inside the closed orbit in the phase plane when the pendulum's length is fixed. Such near-constant quantities are called *adiabatic invariants*.

In his book, Arnold deals with the linearized pendulum (see Figure 1), whose angle θ he models by the standard textbook equation

$$\ddot{\theta}+\lambda\theta=0, \tag{1}$$

where $\lambda=g/\ell$ and ℓ is the length of the string. If ℓ—and hence λ—changes adiabatically (i.e., slowly), the area enclosed by the trajectory in the phase plane $(\theta,\dot{\theta})$ with a frozen λ is an adiabatic invariant. For example, if ℓ changes with small speed ε over a long time $1/\varepsilon$, then the area changes by the small amount $O(\varepsilon)$ over the course of this time. The area thus changes little even if ℓ changes appreciably, say by half.

[1] This "memory" is not perfect, unlike in quantum mechanics.

To see what this near-constancy says about the amplitude of oscillations, we observe that the trajectory in the phase plane of (1) when ℓ is frozen is an ellipse:

$$\theta=\theta_{\max}\sin\omega t,\qquad p=\dot{\theta}=\theta_{\max}\omega\cos\omega t,$$

where $\omega=\sqrt{g/\ell}$. The area of this ellipse is $\pi\theta_{\max}^2\omega=\pi\sqrt{g}(\theta_{\max}\ell^{-1/4})^2$, so that

$$\frac{\theta_{\max}}{\ell^{1/4}}\approx\text{const.} \tag{2}$$

is an adiabatic invariant. According to (2), slowly shortening ℓ decreases the amplitude $\theta_{\max}$. *But this answer is wrong* — the opposite happens in reality (as I show later). Arnold escaped the wrong conclusion (2) by making an error in another place: he took $\lambda=\ell/g$ instead of the correct $\lambda=g/\ell$. This incorrect choice yielded

$$\ell^{1/4}\theta_{\max} \tag{3}$$

instead of (2) as an adiabatic invariant. Despite being the product of an error, (3) correctly predicts that shortening ℓ increases $\theta_{\max}$. How can the right solution give the wrong answer and the wrong solution give the seemingly right answer?[2]

Actually, (3)—although more plausible physically than (2)—is still incorrect. What is going on? The resolution of this mess is that the fundamental premise—i.e., the familiar equation (1)—is an incorrect model of a pendulum with *variable* length.

[2] In fact, a reader recently corrected Arnold's error and came, as we just did, to the conclusion (2) that—despite the correct argument—is wrong. My attempts to resolve this paradox prompted this article.

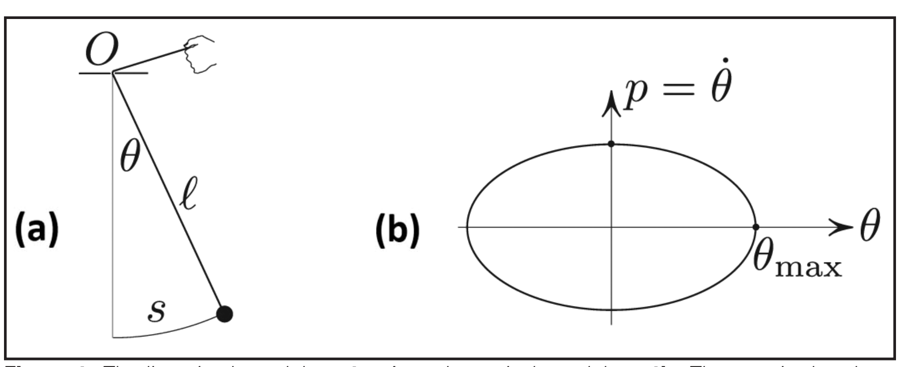


Figure 1. *The linearized pendulum.* **1a.** *A mathematical pendulum.* **1b.** *The area in the phase plane for fixed ℓ is an adiabatic invariant. Figure courtesy of Mark Levi.*

Fixing Equation (1)

The standard textbook derivation of (1) does not work if ℓ varies; instead, we apply the rotational version of Newton's second law to the pendulum's bob — the point where the entire mass is concentrated:

$$\frac{d}{dt}(\text{angular momentum})=\text{torque}. \tag{4}$$

Here, the angular momentum and torque are relative to the fixed pivot O. Deciphering the quantities in (4) yields the ordinary differential equation for θ:

$$\frac{d}{dt}(\ell^2\dot{\theta})=-g\ell\theta \tag{5}$$

(after cancelling the mass and replacing $\sin\theta$ with θ on the right). This equation coincides with (1) for $\ell=\text{const.}$, but not otherwise. What is the correct adiabatic invariant? We write (5) as a Hamiltonian system and express the area bounded by a trajectory for frozen ℓ. To that end, we introduce the momentum $p=\ell^2\dot{\theta}$ (angular, in fact) and rewrite (5) as a system:

$$\dot{\theta}=\ell^{-2}p,\qquad \dot{p}=-g\ell\theta.$$

For frozen ℓ, trajectories are ellipses with semiaxes $\theta_{\max}$ and $p_{\max}=\ell^2\theta_{\max}\omega$, where ω is as before. The area of such an ellipse is

$$\pi\ell^2\theta_{\max}^2\omega=\text{const.}\cdot(\ell^{3/4}\theta_{\max})^2$$

using $\omega=\sqrt{g/\ell}$, so that

$$\ell^{3/4}\theta_{\max}\approx\text{const.} \tag{6}$$

is an adiabatic invariant. This is a correction of (3)—which is still off by a factor of $\sqrt{\ell}$ despite being an improvement over (2)—and confirms that shortening ℓ increases $\theta_{\max}$.

A Physical Plausibility Argument

How can we see without calculation that shortening ℓ increases $\theta_{\max}$? As we shorten ℓ by pulling in the string, we do work against the string's tension. Averaged over one swing, this tension is a little bit more than the weight due to the centrifugal effect. So we do extra work in addition to raising the bob. This extra goes towards increasing the pendulum's "internal" energy, i.e., the kinetic energy (K.E.) at the lowest point in the swing. And so K.E. increases as we shorten ℓ. But a shorter ℓ and greater K.E. imply greater angular amplitude.

I must confess that when I reviewed Arnold's book [2], I missed the fundamental error: the inapplicability of (1). I only realized that something was wrong when a reader noticed that ℓ/g should be g/ℓ, thus changing Arnold's plausible conclusion (3) to the implausible (2).

A Wrong Solution with the Right Answer

Here is another twist to the story. Instead of θ, we could use the arc length $s=\ell\theta$ that—for the pendulum of fixed length—satisfies

$$\ddot{s}+\lambda s=0,\qquad \lambda=\frac{g}{\ell}. \tag{7}$$

This again is a wrong model of the pendulum with variable ℓ, just like the first equation in this article. But surprisingly, (7) has the correct adiabatic invariant (6) (I leave out the verification). Explaining why this wrong equation gives the correct answer is an interesting puzzle.

To sum up, the main mistake lies at the junction between math and physics. This is a bit reminiscent of electricity and plumbing, where shorts or leaks often arise at connections. To mutilate Winston Churchill's famous phrase almost beyond recognition, rarely have so many mistakes been made in such a small problem.

References

[1] Arnold, V.I. (2014). *Mathematical understanding of nature: Essays on amazing physical phenomena and their understanding by mathematicians*. Providence, RI: American Mathematical Society.

[2] Levi, M. (2020). Book review: Mathematical understanding of nature: Essays on physical phenomena and their understanding by mathematicians by V.I. Arnold, translated by Alexei Sossinsky and Olga Sipacheva. *Mathemat. Intelligen.*, *42*(1), 84-86.

Mark Levi (levi@math.psu.edu) is a professor of mathematics at the Pennsylvania State University.

Budget Request

Continued from page 6

high-performance computing capabilities. The budget request also includes a continued focus on climate change and the development of a clean energy future. The proposal designates $379 million for mathematical, computational, and computer science research programs within ASCR, which represents an increase of $93 million (or 32 percent) above the FY 2022 enacted level. Applied mathematics research programs would be funded at $72 million — an increase of $21 million (or 41 percent) from the previous level.

The request also recommends $98 million (a 23.2 percent increase) for ASCR's computational partnerships, which primarily support the Scientific Discovery through Advanced Computing[9] (SciDAC) program. In FY 2023, these funds would finance the transition of mission-critical ECP applications into SciDAC and further develop partnerships with DOE's applied energy offices and data-intensive applications. Furthermore, the administration recommends sustained increases for the Computational Science Graduate Fellowship[10] program so that it can fund additional fellows in AI and quantum computing and expand participation from members of underrepresented groups.

President Biden's FY 2023 budget request for DOD's S&T accounts proposes cuts to basic research, applied research, and advanced technology development in favor of more deliverable capabilities. The S&T accounts would be funded at $16.5 billion — a $2.4 billion decrease from FY 2022 enacted levels. Basic research would be funded at $2.4 billion (a 14 percent decrease), applied research at $5.8 billion (a 16 percent decrease), and advanced technology development at $8.3 billion (a 10 percent decrease). The Army and Navy basic research accounts would respectively decrease by $143 million (23.4 percent) and $109 million (15.6 percent), while the Air Force account would increase by $5.8 million — roughly 1 percent when compared to FY 2022 enacted levels. The budget request would also reduce the National Defense Education Program by 8.9 percent and basic research initiatives by 18.8 percent; in addition, it would transfer $816 million from Air Force to Space Force[11] for weather services research. However, given Congress' continued support for DOD basic research, it is unlikely that many of the proposed cuts will remain in a final appropriations package.

The budget request would provide $49 billion in FY 2023 discretionary funding for the NIH base budget — an increase of $4 billion (9.1 percent) above the FY 2022 enacted level. Of this $49 billion, $5 billion ($4 billion in new funding) is intended for the Advanced Research Projects Agency for Health[12] (ARPA-H): a new agency that will "support transformative high-risk, high-reward research to drive biomedical and health breakthroughs—ranging from molecular to societal—that would provide transformative solutions for all patients" [2]. In addition to the discretionary funding, the budget seeks $12.1 billion in mandatory funds for pandemic preparedness activities. Congress will likely not accept the administration's FY 2023 proposal due to its longstanding support of basic research, meaning that appropriators will then face the difficult task of balancing investment in NIH's base budget with funding for ARPA-H.

The FY 2023 budget request formally initiated the congressional appropriations process. However, the timing of final FY 2023 appropriations remains uncertain. Furthermore, the process' late start increases the likelihood of a stop-gap funding measure—known as a continuing resolution—to avoid a government shutdown and maintain funding for federal agencies beyond the end of FY 2022 on September 30. SIAM has submitted testimony that highlights funding priorities for NSF and DOE to both the House and Senate Committee on Appropriations as they work to finalize an FY 2023 spending package. In the meantime, the Society will stay abreast of the FY 2023 appropriations cycle and its impact, advocate for strong funding for applied mathematics and computational science programs at relevant agencies, and keep members informed.

[9] https://www.scidac.gov

[10] https://www.krellinst.org/csgf

[11] https://www.spaceforce.mil

[12] https://www.nih.gov/arpa-h

References

[1] Lewis-Burke Associates, LLC. (2022, March 31). *Analysis of the President's FY 2023 budget request for federal research, health, and higher education programs: Implications for research, higher education, and academic medicine*. Washington, D.C.: Lewis-Burke.

[2] National Institutes of Health. (2022). *Advanced Research Projects Agency for Health (ARPA-H)*. Retrieved from https://www.nih.gov/arpa-h.

Andrew Herrin is an associate at Lewis-Burke Associates LLC—SIAM's governmental relations partner—and a SIAM liaison in Washington, D.C.

MDS22 Will Showcase the Latest Advances in the Mathematics of Data Science in San Diego (and Online)

By Lior Horesh, Lars Ruthotto, and Karen E. Willcox

The 2022 SIAM Conference on Mathematics of Data Science[1] (MDS22) will take place in a hybrid format and welcome a diverse set of attendees—representing SIAM's breadth of leaders in industrial and applied mathematics—to San Diego, Calif., from September 26-30. Since MDS20[2] (the conference's first iteration) was completely virtual due to the COVID-19 pandemic, this second installment will be the first MDS meeting where attendees can gather in person, though the option to participate virtually still exists. Conference themes range from the mathematical foundations of data science to statistical inference, machine learning (ML), applied probability, scalable algorithms, and applications of data science across science, engineering, technology, and society. MDS22 provides a unique opportunity for individuals to aggregate data science advances, exchange ideas, and set the agenda for the new SIAM Activity Group on Data Science[3] (SIAG/DATA) — while also recognizing that many other SIAGs share important overlaps with the broad field of data science. As co-chairs of the MDS22 Organizing Committee, we are particularly excited to see just how many SIAGs are reflected in the conference themes.

[1] https://www.siam.org/conferences/cm/conference/mds22
[2] https://www.siam.org/conferences/cm/conference/mds20
[3] https://www.siam.org/membership/activity-groups/detail/data-science

Organized by a diverse global committee of thought leaders from academia, national laboratories, and industry, the meeting will be packed with cutting-edge research. Academicians and practitioners alike will find many opportunities to learn new skills, network with friends and colleagues, and embark on novel research challenges. In addition to nine invited plenary talks and a SIAG/DATA Early Career Prize lecture by Weijie Su (University of Pennsylvania), the conference will include eight minitutorials that offer entry points into the mathematics of data science. MDS22 is also set to feature more than 100 posters, over 160 minisymposia sessions, and 20 contributed presentation sessions. We hope that early-career researchers will find the industry and funding panels, networking events, and co-located SIAM Career Fair—the latter of which will take place entirely on site—particularly valuable.

Central components of nearly any data science task are optimization, linear algebra, and inverse problems — all of which have a rich history within the SIAM community. These topics will thus be central to many minisymposia presentations, contributed posters, and invited talks. Plenary speaker Wotao Yin (Alibaba Group) and minitutorial speaker Stephen Wright (University of Wisconsin–Madison) will both address the role of optimization methods in data science. Numerical linear algebra, its use in network science, and multilinear extensions for the analysis and processing of tensors will be the focus of David Gleich's (Purdue University) invited talk and a minitutorial on tensor decompositions by Tamara Kolda (MathSci.ai), Grey Ballard (Wake Forest University), and Daniel Dunlavy (Sandia National Laboratories). Another traditional SIAM area with strong ties to data science is inverse problems and data assimilation, which will feature in the presentations of plenary speaker Andrew Stuart (California Institute of Technology) and minitutorial speakers Noemi Petra and Tucker Hartland (both of the University of California, Merced).

MDS22 will also provide new links to several aspects of cutting-edge ML research. The invited talk by Joan Bruna (New York University) and minitutorial by Michael Bronstein (University of Oxford), Petar Veličković (DeepMind Technologies), and Francesco Di Giovanni (Twitter) will cover the emerging area of geometric deep learning. Caroline Uhler's (Massachusetts Institute of Technology) plenary talk will underscore the important role of statistics and causal modeling, while Marco Cuturi's (Google) minitutorial will introduce computational optimal transport. In a two-part minitutorial, Johannes Blaschke (Lawrence Berkeley National Laboratory) will introduce attendees to the Julia programming language and its use in data science applications.

Additionally, MDS22 intends to expose challenges and advances in the ethics and fairness of data science. Invited speaker Talitha Washington (Clark Atlanta University) will deliver a mathematical perspective on how to address bias and ethics in data science; John Kleinberg's (Cornell University) plenary talk will expand upon this important topic. The rising frontiers of MDS in complex societal and industrial applications will be evident in Dawn Woodard (LinkedIn) and Lester Mackey's (Microsoft Research) invited plenary talks.

The Organizing Committee's efforts to maximize the diversity of speakers and perspectives at MDS22 will be complemented by the Sustainable Horizons Institute's[4] Broader Engagement (BE) program.[5] The BE track promotes inclusion within the MDS community through interdisciplinary technical sessions, dedicated minitutorials, and discussions that help advance knowledge and skillsets. The program also includes a comprehensive networking and mentoring component that seeks to foster a sense of belonging among all participants.

Beyond the many opportunities for technical growth at MDS22, students will find numerous occasions to extend their networks and develop their careers. For instance, a student career panel will offer attendees a chance to hear panelists' perspectives in a question-and-answer-style discussion on topics such as career paths, work environments, success measures, value alignment, and work-life balance. In addition, we encourage students to participate in the in-person SIAM Career Fair, explore future job prospects, and interact with mentors from the BE program to evaluate their career goals.

[4] https://shinstitute.org
[5] https://shinstitute.org/be-mds-2022

MDS22 will be a fantastic setting for companies and organizations to meet and recruit top talent. For example, several talks will showcase mathematics applications in real-world data science situations. Companies can also engage with attendees via the Career Fair and several conference sponsorship opportunities. The industry panel will facilitate open conversation about the interplay between industry and academia as it relates to MDS.

There are many reasons to be excited about a trip to San Diego this fall, though participants can enjoy MDS22 remotely if they prefer. Early registration[6] opens in July and will be available until September 1, after which standard registration rates apply. SIAM is also proud to continue its collaboration with the Gesellschaft für Angewandte Mathematik und Mechanik (GAMM) in Germany and provide reduced registration for GAMM members. We're looking forward to seeing you online or in San Diego!

The 2022 SIAM Conference on Mathematics of Data Science (MDS22) will take place in a hybrid format from September 26-30, 2022, with the in-person component at the Town and Country Resort in San Diego, Calif. Image courtesy of the Town and Country Resort.

Lior Horesh, Lars Ruthotto, and Karen E. Willcox are co-chairs of the Organizing Committee for the 2022 SIAM Conference on Mathematics of Data Science. Lior Horesh is a Senior Manager at IBM Research. He manages the Mathematics of AI group, which approaches some of the big challenges that the field of artificial intelligence faces from a principled mathematical angle. He also serves as an adjunct associate professor at Columbia University, where he teaches quantum computing and machine learning. Lars Ruthotto is an applied mathematician who develops computational methods for machine learning and inverse problems. He is an associate professor in the Department of Mathematics and Department of Computer Science at Emory University. Karen E. Willcox is director of the Oden Institute for Computational Engineering and Sciences and a professor of aerospace engineering and engineering mechanics at the University of Texas at Austin. She is also external faculty at the Santa Fe Institute.

[6] https://www.siam.org/conferences/cm/registration/mds22-registration

Julia

Continued from page 5

we make Zygote aware of our insight? We define a custom differentiation rule for `reduce` and inject it into our code (see Figure 1, on page 5).

With this special case for product reduction, we now have a 17-line, fast implementation of `speelpenning` on the GPU without ever interacting with the developers of CUDA.jl, Zyote.jl, or ChainRules.jl. Imagine the effort of doing this in a computing language that lacks macros and abstract syntax tree manipulation, multiple dispatch, or JIT compilation. These features allow us to implement the fully differentiable power flow solver ExaPF.jl[17] that runs efficiently on both GPU and central processing unit architectures — all while providing first and second-order derivatives of the power flow. Still, a cautionary remark is appropriate. Because Julia is developing quickly, all of these features come with a certain fragility and incompleteness. Nevertheless, we are confident that we can achieve exascale with Julia on Aurora and Frontier.

[17] https://github.com/exanauts/ExaPF.jl

Is It Fast?

Julia utilizes LLVM, a leading compiler backend for `C`/`C++`. In theory, it should permit Julia implementations to achieve `C`-like performance. However, LLVM also requires a considerable low-level understanding of the Julia language implementation. This balance of development cost versus performance is an ongoing debate in the scientific community. How desirable is a 2x speedup at 10 times the financial development cost? Julia enables an insightful preliminary performance assessment for novel algorithm implementations. Although not required, one can even create competitive BLAS implementations in pure Julia.[18]

ExaTron.jl[19] solves the optimal power flow subproblem and is crucial for the performance of our ExaSGD software stack. Incidentally, we also implemented a `C++`/CUDA solution to compare performance (see Figure 2, on page 5).

We do not claim that both versions of the code are fully optimized. But given the law of marginal gains, a Julia implementation requires much less effort to develop while simultaneously providing huge benefits for expression transformations like differentiable programming and portability.

[18] https://github.com/JuliaLinearAlgebra/Octavian.jl
[19] https://github.com/exanauts/ExaTron.jl

References

[1] Anitescu, M., Maldonado, D.A., Pacaud, F., Kim, K., Kim, Y., Rao, V., … Subramanyam, A. (2021). Targeting exascale with Julia on GPUs for multiperiod optimization with scenario constraints. *SIAG/OPT Views and News*, *29*(1), p. 1.

[2] Besard, T., Churavy, V., Edelman, A., & De Sutter, B. (2019). Rapid software prototyping for heterogeneous and distributed platforms. *Adv. Engin. Soft.*, *132*, 29-46.

[3] Dunning, I., Huchette, J., & Lubin, M. (2017). JuMP: A modeling language for mathematical optimization. *SIAM Rev.*, *59*(2), 295-320.

[4] Innes, M. (2018). Don't unroll adjoint: Differentiating SSA-form programs. Preprint, *arXiv:1810.07951*.

[5] Perkel, J.M. (2019). Julia: Come for the syntax, stay for the speed. *Nature*, *572*, 141-143.

[6] Speelpenning, B. (1980). *Compiling fast partial derivatives of functions given by algorithms* [Doctoral dissertation, University of Illinois at Urbana-Champaign]. U.S. Department of Energy Office of Scientific and Technical Information.

Michel Schanen is an assistant computer scientist at Argonne National Laboratory (ANL) with a background in automatic differentiation and high-performance computing. He is currently investigating differentiable programming and rapid prototyping for large-scale computing in Julia. Valentin Churavy is a Ph.D. student at the Massachusetts Institute of Technology's Julia Lab, where he works on accelerated and high-performance computing. Youngdae Kim holds a Ph.D. in computer sciences from the University of Wisconsin-Madison with a focus on nonlinear programming. He is presently a postdoctoral appointee at ANL, where his research is driven by graphical processing unit accelerated solvers for nonlinear programming. Mihai Anitescu is a senior mathematician at ANL and a professor of statistics at the University of Chicago. He works in the areas of numerical optimization and uncertainty quantification, which he applies to electrical power networks.