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

siam news

Volume 51/ Issue 10
December 2018

Special Issue on Computational Science and Engineering

Read about various applications related to computational science and engineering in this month's ***special issue****!*

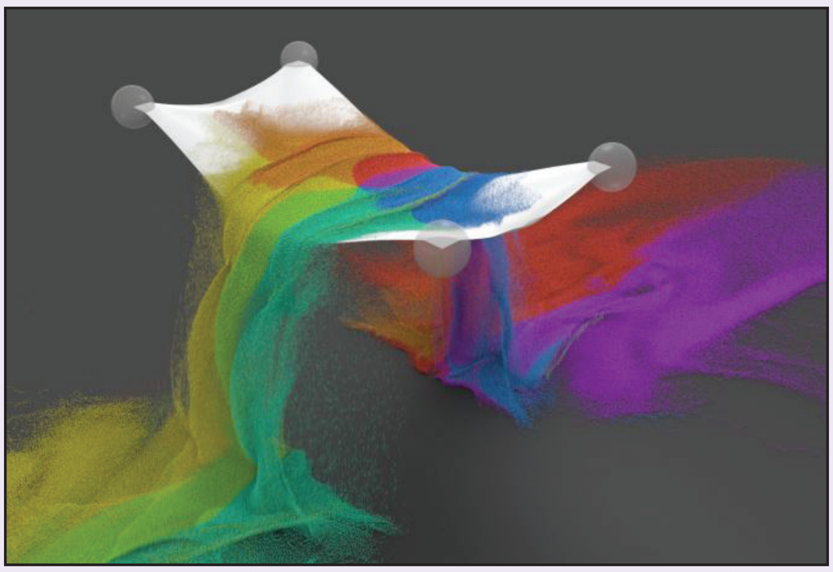


Figure 1. *The coupling of elastic cloth with seven million colored grains of sand displays dazzling flow patterns. Image courtesy of [1].*

In an article titled "The Serious Mathematics of Digital Animation" on page 3, Matthew R. Francis recaps Joseph Teran's invited presentation on the use of mathematical models for computer-generated imagery from the 2018 SIAM Annual Meeting.

The Functions of Deep Learning

By Gilbert Strang

Suppose we draw one of the digits $0, 1, \ldots, 9$. How does a human recognize which digit it is? That neuroscience question is not answered here. How can a computer recognize which digit it is? This is a machine learning question. Both answers probably begin with the same idea: *learn from examples*.

So we start with M different images (the training set). An image is a set of p small pixels — or a vector $\boldsymbol{v}=(v_1,\ldots,v_p)$. The component v_i tells us the "grayscale" of the $i\mathrm{th}$ pixel in the image: how dark or light it is. We now have M images, each with p features: M vectors $\boldsymbol{v}$ in p-dimensional space. For every $\boldsymbol{v}$ in that training set, we know the digit it represents.

In a way, we know a function. We have M inputs in $\mathbf{R}^p$, each with an output from 0 to 9. But we don't have a "rule." We are helpless with a new input. Machine learning proposes to create a rule that succeeds on (most of) the training images. But "succeed" means much more than that: the rule should give the correct digit for a much wider set of test images taken from the same population. This essential requirement is called *generalization*.

What form shall the rule take? Here we meet the fundamental question. Our first answer might be: $F(\boldsymbol{v})$ could be a linear function from $\mathbf{R}^p$ to $\mathbf{R}^{10}$ (a 10 by p matrix). The 10 outputs would be probabilities of the numbers 0 to 9. We would have $10p$ entries and M training samples to get mostly right.

The difficulty is that linearity is far too limited. Artistically, two 0s could make an 8. 1 and 0 could combine into a handwritten 9 or possibly a 6. Images don't add. Recognizing faces instead of numbers requires a great many pixels — and the input-output rule is nowhere near linear.

Artificial intelligence languished for a generation, waiting for new ideas. There is no claim that the absolute best class of functions has now been found. That class needs to allow a great many parameters (called weights). And it must remain feasible to compute all those weights (in a reasonable time) from knowledge of the training set.

The choice that has succeeded beyond expectation—and transformed shallow learning into deep learning—is *continuous piecewise linear (CPL) functions*. Linear for simplicity, continuous to model an unknown but reasonable rule, and piecewise to achieve the nonlinearity that is an absolute requirement for real images and data.

See **Deep Learning** *on page 4*

Designing Algorithms to Increase Fairness in Artificial Intelligence

By Anil Aswani and Matt Olfat

The increasing role of artificial intelligence (AI) to automate decision-making sparks concern about potential AI-based discrimination. Such bias is undesirable in the quest to minimize inequality, and legal regulations require that AI not discriminate against protected classes on the basis of gender, race, age, and the like. Exclusion of data on protected classes when training AI is one strategy to minimize prejudice. However, this is not only a naive approach to fairness, but also insufficient because AI systems can learn to use information on protected classes via other data [2]; for example, race can often be inferred from a home address. Consequently, more sophisticated algorithmic and computational approaches are necessary to ensure that AI behaves impartially.

Such concerns about AI fairness are not merely theoretical. Researchers have observed several instances of biased data and prejudiced AI. When discriminatory behavior influences the data that trains AI, the resulting AI output often perpetuates this bias. For instance, doctors frequently undertreat pain in women as compared to men [3]. AI systems for pain management trained using such biased data can algorithmically preserve this gender discrimination. Social media algorithms provide further examples; in one situation, LinkedIn disproportionately advertised high-paying jobs to men. In another, Facebook's algorithms displayed considerable racial prejudice in censorship [1].

The first step towards developing fair AI is to quantify fairness, for which investigators have proposed several definitions for supervised learning to date. Put simply, these definitions require that an individual's membership in a protected class (e.g., gender) will not impact AI's outcome (e.g., approval or disapproval of a loan application). One can tailor this process for the purpose of *classification*, where the goal is to construct a function $h:\mathbb{R}^p \rightarrow \{-1,+1\}$ (known as a classifier) that uses a vector of descriptive features $x \in \mathbb{R}^p$, characterizing each individual to predict a binary outcome $y \in \{-1,+1\}$ for him/herself. When an individual's protected class $z \in \{-1,+1\}$ is binary, the notion of *demographic parity* [4] at level Δ requires that $|\mathbb{P}[h(x)=+1|z=+1]-\mathbb{P}[h(x)=+1|z=-1]| \leq \Delta$. Intuitively, the probability of receiving a positive outcome from the classifier is independent of the protected class's value. Other definitions of fairness—such as *equal opportunity*—also exist [5].

While there are many approaches for constructing accurate classifiers from training data, researchers have only recently begun

See **Artificial Intelligence** *on page 3*

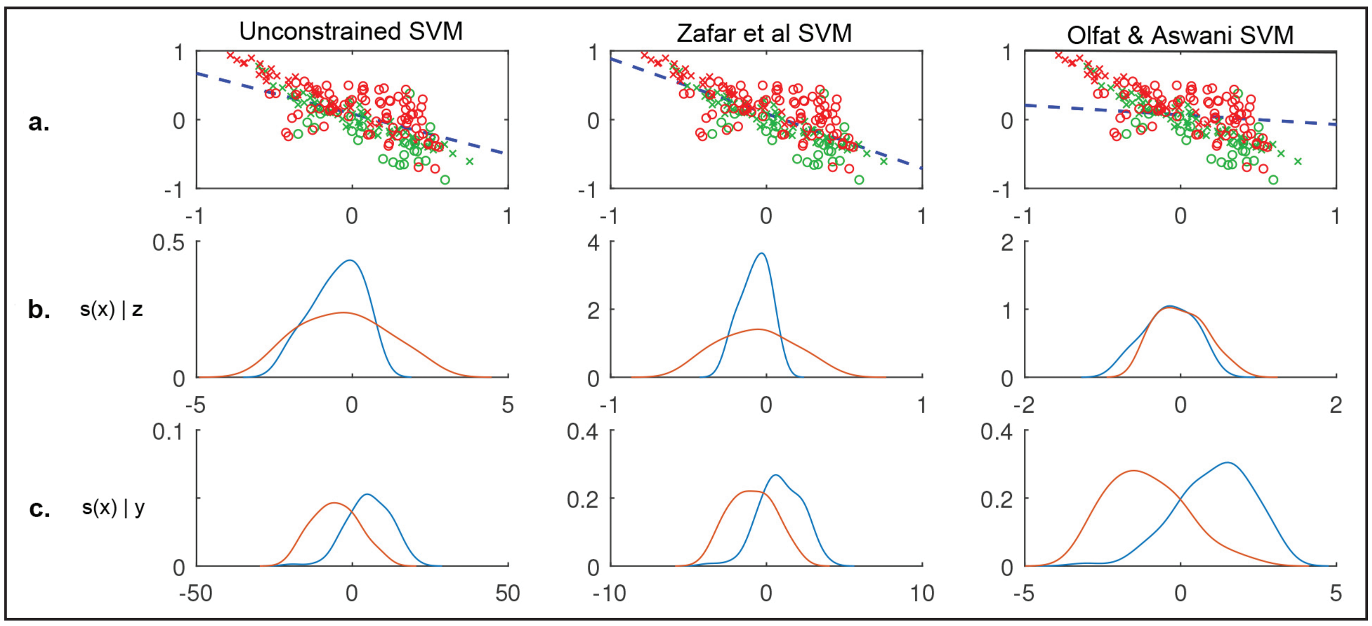


Figure 1. *From left to right: comparison of a linear support vector machine (SVM), linear SVM of [9], and linear SVM of [7].* **1a.** *We separate red and green points while remaining fair between "x"s and "o"s.* **1b.** *The distribution of the linear score $s(x)$ conditioned on the protected class z.* **1c.** *The distribution of the linear score $s(x)$ conditioned on outcome y. Fairness occurs when conditional distributions in 1b are similar, and accuracy occurs when conditional distributions in 1c are dissimilar. Figure courtesy of [7].*

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

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

siam news
Volume 51/ Issue 10/ December 2018

4 **A Near-perfect Heat Exchange**
Thermal contact between two objects results in convergence of their temperatures, which are incapable of overshooting or reversing because temperature lacks inertia. In this month's column, Mark Levi describes how nature circumvents this limitation engendered by the second law of thermodynamics, thus allowing two objects to nearly exchange temperatures. Through discrete approximation and a setup with two liquids flowing in opposite directions separated by a heat-conducting membrane, Levi draws parallels with arteries in our arms that run along deep veins.

6 **Randomized Projection Methods in Linear Algebra and Data Analysis**
Large complex datasets require analysis and processing that give rise to mathematical and algorithmic challenges. Finding ways to depict such datasets using fewer parameters allows for their efficient storage, transmission, and interpretation. Per-Gunnar Martinsson describes two randomized algorithms that can handle large datasets in high-dimensional spaces. These algorithms are typically more accurate and less stringent—in terms of random number generation—than their traditional counterparts in scientific computing.

7 **Progress by Accident: Some Reflections on My Career**
On the occasion of his 90th birthday last year, Walter Gautschi—a leading mathematician in the areas of approximation theory, orthogonal polynomials, special functions, and numerical analysis—reflects on his research and career trajectory. From requests for his insight on new company projects to invitations to give lectures at conferences, Gautschi recognizes the role of chance and opportunity in piquing his interest in various research areas.

7 **Professional Opportunities and Announcements**

Rhapsodizing About Bohemian Matrices

FROM THE SIAM PRESIDENT

By Nicholas Higham

In the November 1987 issue of *SIAM News*, Eric Grosse and Cleve Moler reported a 9×9 symmetric matrix of 1s and -1s for which EISPACK failed to accurately compute one of the eigenvalues on a particular machine [5]. In my January/February 2018 column, I described how surprising results can occur when computing the determinant of a matrix of small integers in floating-point arithmetic [6]. And as is well known, a matrix of 0s, 1s, and -1s devised by Wilkinson is a matrix that achieves the worst-case growth factor for Gaussian elimination with partial pivoting.

Matrices of small integers—innocuous as they may seem—can clearly provoke interesting behavior. In fact, such matrices have long been a subject of study, not least because analytical formulas can be obtained for the eigenvalues, inverse, determinant, etc. in many cases. For this reason, and the fact that they are stored exactly in floating-point arithmetic, they make good test matrices. A matrix that was popular in the early days of digital computing is

$$W = \begin{bmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{bmatrix},$$

whose inverse has moderately large integer entries. John Todd described it as "the notorious matrix W of T. S. Wilson" [9], and Moler recently investigated its properties and history [7]. A favorite of mine is the Pascal matrix, with (i,j) entry $\binom{i+j-2}{j-1}$. A great deal is known about this positive definite, totally positive matrix. A scaled and rotated version of the Cholesky factor of the matrix (`pascal(n,2)` in MATLAB) has the intriguing property that it is a cube root of the identity, shown here for $n=4$:

$$\begin{bmatrix} -1 & -1 & -1 & -1 \\ 3 & 2 & 1 & 0 \\ -3 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^3 = I.$$

Deep connections exist between integer matrices—particularly positive definite ones—and number theory, as Olga Taussky-Todd demonstrated in her work; see [8] and an appendix to Harvey Cohn's 1978 book on algebraic number theory [3]. In a 1991 letter, Taussky-Todd told me that "we [she and her husband, John Todd] have been really interested in batteries of test matrices." Indeed, the Hilbert matrix was a favorite of theirs.

In scientific computing, researchers often employ increases in computing power to solve larger problems (rather than a greater number of small problems), but a systematic search of small problems can offer new insights. Recent work on Bohemian matrices combines the latter approach with theoretical analysis. Rob Corless and Steven Thornton (both of the University of Western Ontario) coined the term "Bohemian matrices"—a contraction of BOunded HEight Matrix of Integers—to denote families of matrices with entries drawn from a fixed discrete set of small integers (or some other discrete set) [4].

Thornton has been investigating the distribution of eigenvalues and characteristic polynomials of Bohemian matrices of dimensions up to about 10, with elements in sets such as $\{-1,0,1\}$, using a mix of numerical and symbolic computation. The number of matrices grows rapidly with the dimension, and Thornton uses random sampling when exhaustive computation is not possible. His plots show remarkable structures and make wonderful pieces of art (see Figure 1). A collection of images is available on his website.[1]

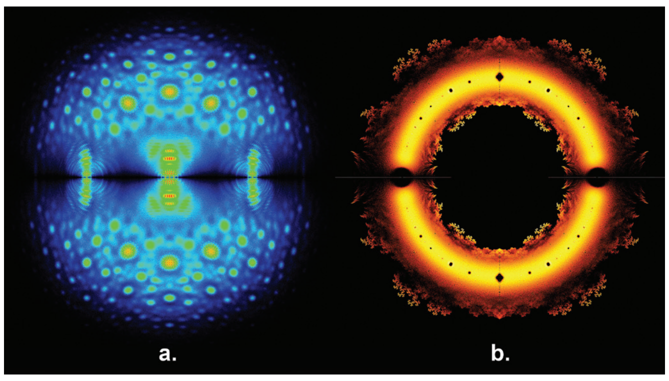
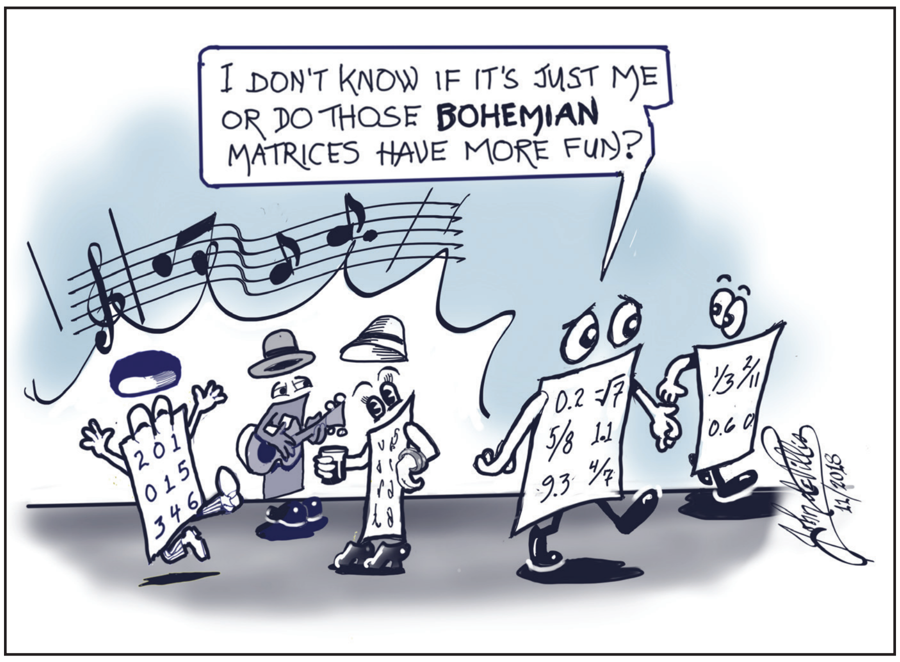


Figure 1. 1a. *A density plot in the complex plane of the eigenvalues of a sample of 73 million* 5×5 *matrices with entries sampled from the set* $\{-20,-1,0,1,20\}$. *Color represents the eigenvalue density.* **1b.** *A density plot in the complex plane of the eigenvalues of all* 19×19 *(Frobenius) companion matrices with entries in* $\{-1,1\}$. *Color again represents the eigenvalue density. See [1-2] and the references therein for discussions of the equivalent polynomial root problem for Littlewood polynomials. Figure courtesy of Steven Thornton.*

Various open questions have been identified concerning eigenvalue distributions, ranks, and the number of distinct characteristic polynomials in particular classes of Bohemian matrices. Some of these are listed in Thornton's characteristic polynomial database,[2] which contains more than 10^9 polynomials from a variety of Bohemian matrix families.

Last June, Corless and I organized a workshop entitled *Bohemian Matrices and Applications* that brought together researchers with expertise in matrix theory, numerical linear algebra, computer algebra, algebraic geometry, and number theory. Some open problems were solved and many others formulated. Videos of the introductory talks (by Corless, Thornton, and myself) and slides from the other talks are available online.[3,4]

This is my last column as SIAM president. I wish Lisa Fauci all the best as she takes over the presidency in January 2019.


I DON'T KNOW IF IT'S JUST ME OR DO THOSE BOHEMIAN MATRICES HAVE MORE FUN?
2 0 1
0 1 5
3 4 6
0.2 √7
5/8 1.1
9.3 4/7
1/3 2/11
0.6 0
11/2018

Cartoon created by mathematician John de Pillis.

References

[1] Baez, J. (2011, December 15). *The Beauty of Roots*. Retrieved from http://math.ucr.edu/home/baez/roots/.

[2] Borwein, P., & Jörgenson, L. (2001). Visible Structures in Number Theory. *Amer. Math. Monthly*, *108*, 897-910.

[3] Cohn, H. (1978). *A Classical Invitation to Algebraic Numbers and Class Fields* (with two appendices by Olga Taussky-Todd). New York, NY: Springer-Verlag.

[4] Corless, R.M., & Thornton, S.E. (2016). The Bohemian Eigenvalue Project. *ACM Comm. Comput. Alg., 50*, 158-160.

[5] Grosse, E., & Moler, C. (1987). Underflow Can Hurt. *SIAM News, 20*(6), p. 1.

[6] Higham, N.J. (2018). Snap to Structure. *SIAM News, 51*(1), p. 2.

[7] Moler, C. (2018, August 20). *Reviving Wilson's Matrix*. Retrieved from https://blogs.mathworks.com/cleve/2018/08/20/reviving-wilsons-matrix/.

[8] Taussky, O. (1960). Matrices of Rational Integers. *Bull. Amer. Math. Soc.*, *66*, 327-346.

[9] Todd, J. (1950). The Condition of a Certain Matrix. *Proc. Camb. Philos. Soc., 46*, 116-118.

Nicholas Higham is Royal Society Research Professor and Richardson Professor of Applied Mathematics at the University of Manchester. He is the current president of SIAM.

[1] http://www.bohemianmatrices.com/
[2] http://www.bohemianmatrices.com/cpdb
[3] http://www.maths.manchester.ac.uk/~higham/conferences/bohemian.php
[4] https://nickhigham.wordpress.com/2018/10/15/bohemian-matrices-in-manchester/

siam news

ISSN 1557–9573. Copyright 2018, 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 2017: 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.

Printed in the USA.
siam *is a registered trademark.*

The Serious Mathematics of Digital Animation

By Matthew R. Francis

While computer simulations have a wide range of uses, their goals are generally similar: find the simplest model that recreates the properties of the system under investigation. For scientific systems, this involves matching observed or experimental phenomena as precisely as necessary.

But what about movie simulations? Should they match the processes they replicate so closely? Computer-generated imagery (CGI) is a common feature in both animated and live-action films. For these CGI systems, creating visuals that *look* right is an important task. However, Joseph Teran of the University of California, Los Angeles believes that starting from physical models is still a good idea.

During his invited address at the 2018 SIAM Annual Meeting, held in Portland, Ore., this July, Teran pointed out that beginning with a mathematical system is often easier than drawing from real life. Many movies model a system's various forces and internal structures with partial differential equations (PDEs) for this reason. While solving these equations to produce CGI is computationally expensive, such methods have become powerful tools for creating realistic visual cinematic effects.

Teran and his collaborators utilized a general physical model for a wide range of movie phenomena, such as smoke, sand, snow, water and other fluids, and even clothing (see Figure 1, on page 1). Teran noted that modeling everyday occurrences—such as pouring coffee or the behavior of clothes on a human body—in a convincing manner is much more difficult than simulating exotic things like exploding spaceships. The very familiarity of ordinary systems frequently exposes inconsistencies; this is in contrast to esoteric things, akin to the "uncanny valley effect"[1] wherein attempts at realistic human faces fall short.

From Jell-O to Snow

During his presentation, Teran focused on a particular model known as "elastoplasticity," which allows animators to treat a wide range of visual phenomena with a few equations, governed by a reasonable number of parameters that can be adjusted until things look right. Elastoplastic theory describes materials that both spring back when deformed (hence, elastic) and retain some of their altered shape (plastic).

For example, snow is granular on one level because it comprises small crystals that are visible to the human eye. However, a large-scale view shows that it is an elastoplastic material, as anyone who has ever made a snowball knows. How well a snowball holds together depends on its texture and "wetness," among other things. And how well the initial handful packs together partly depends on snow's plasticity. The crumbliness of "dry" snow—which renders it unsuitable for snowballs—also means that it blows more readily in the wind, making for easier cleanup. The varying elastoplastic properties of snow dictate whether or not it flows, thus determining the manner in which it drifts and the dangers of possible avalanches.

Based on this theoretical framework, Teran and his colleagues consulted with Walt Disney Animation Studios to generate realistic-looking snow for the computer-animated film *Frozen*. Animators must create movie special effects without having to produce simulations of various phenomena from scratch. This is when PDEs become useful, as does reduction of the physical model's parameters, which can be adjusted based on a film's visual needs.

See **Digital Animation** *on page 6*

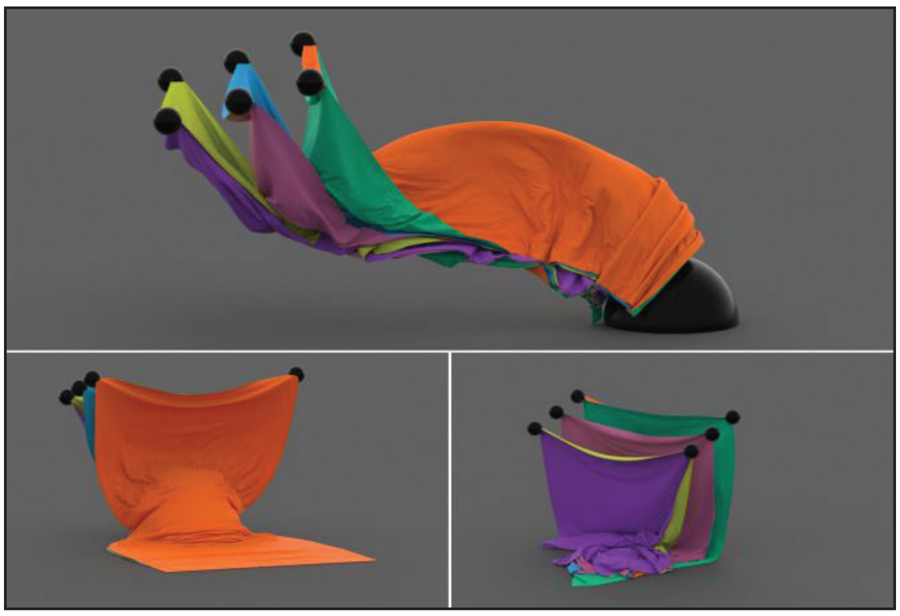


Figure 2. *A sphere pushes three pieces of cloth—with approximately 1.4 million triangles—back and forth. This yields complex folds and contact. Image courtesy of [1].*

[1] The uncanny valley effect is the unsettling feeling that people experience upon encountering faces on robots or in digital art that are very nearly human in appearance but not quite convincingly realistic.

Artificial Intelligence

Continued from page 1

developing methods that also ensure fairness. Linear classifiers $h(x) = \operatorname{sign}(x^{\mathsf{T}}\beta - t)$ compare a linear score $s(x) = x^{\mathsf{T}}\beta$ (dependent upon a vector of parameters $\beta \in \mathbb{R}^p$) to a threshold $t \in \mathbb{R}$. One can classically compute the β parameters from training data by solving a convex optimization problem corresponding to either logistic regression or a support vector machine (SVM). In contrast, designing a fair linear classifier requires choosing the β parameters to satisfy the fairness definition for a small value of Δ while still maintaining high accuracy in predicting outcome y. This can be accomplished by adding appropriate constraints to the original convex optimization problem and developing algorithms to solve the resulting optimization problem. Here we describe these steps in more detail for the special case $\mathbb{E}(x) = 0$, where we assume the features have a 0 mean.

One approach augments the optimization problems that underlie logistic regression or SVM with an additional constraint $-\delta \le \mathbb{E}[x^{\mathsf{T}}\beta \mid z = +1] - \mathbb{E}[x^{\mathsf{T}}\beta \mid z = -1] \le \delta$, which ensures that the means of the score function—when conditioned on the protected class—are approximately equal [9]. Adding another constraint, $-\delta \le \mathbb{E}[\beta^{\mathsf{T}}xx^{\mathsf{T}}\beta \mid z = +1] - \mathbb{E}[\beta^{\mathsf{T}}xx^{\mathsf{T}}\beta^{\mathsf{T}} | z = -1] \le \delta$, to the optimization problem can further improve fairness of the linear classifier. This substantiates that the variances of the score function—when conditioned on the protected class—are approximately equal. However, training occurs with data (x_i, y_i, z_i) for $i = 1,\dots,n$. Therefore, the actual constraints in the optimization problem correspond to their sample average approximations

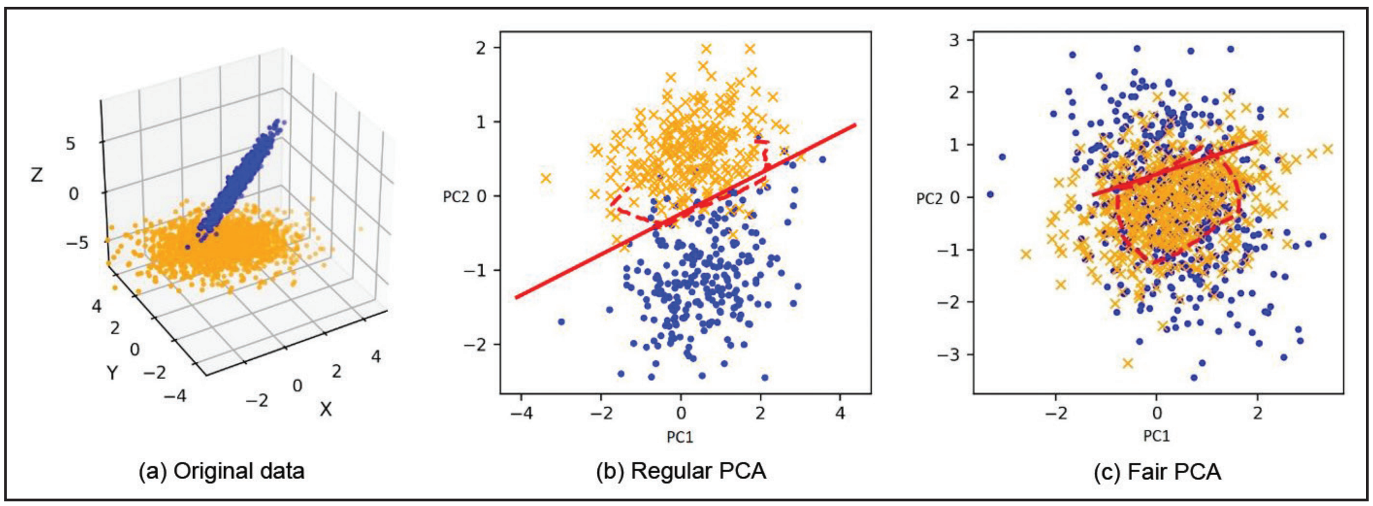
$$-\delta \le \Big(\sum_{i:z_i=+1} x_i - \sum_{i:z_i=-1} x_i\Big)^{\mathsf{T}}\beta \le \delta$$

$$-\delta \le \beta^{\mathsf{T}}\Big(\sum_{i:z_i=+1} x_i^{\mathsf{T}}x_i - \sum_{i:z_i=-1} x_i^{\mathsf{T}}x_i\Big)\beta \le \delta.$$

Adding only the first (linear) fairness constraint results in a convex optimization problem, while including the second (indefinite quadratic) fairness constraint yields a nonconvex optimization problem. Consequently, attaining good numerical solutions that satisfy the second fairness constraint necessitates careful algorithmic design [7]. Because this second constraint is an indefinite quadratic, a spectral decomposition can separate it into a difference of two convex quadratic functions. Once decomposed in this manner, the problem is solvable via algorithms from difference-of-convex programming [8]. Figure 1 (on page 1) depicts the improved fairness with high accuracy of a linear classifier trained by this method.

The application of dimensionality reduction algorithms, like the classic principal component analysis (PCA), also raises the question of fairness. This issue is less well-studied because the lack of outcome data makes past fairness definitions inapplicable. Recent work has moved toward developing a fair version of PCA [6]. Let $x \in \mathbb{R}^p$ be a vector of features describing an individual, and suppose that the protected class $z \in \{-1, +1\}$ of an individual is binary. In this case, a dimensionality-reducing map $\Pi : \mathbb{R}^p \to \mathbb{R}^d$ with $d < p$ is Δ-fair with respect to a family $\mathcal{F}$ of classifiers if $|\mathbb{P}[h(\Pi(x)) = +1 | z = +1] - \mathbb{P}[h(\Pi(x)) = +1 \mid z = -1]| \le \Delta$ for all classifiers $h \in \mathcal{F}$. The intuition behind this definition is that a dimensionality reduction is fair if one cannot accurately predict the protected class of any single point using the lower-dimensional data computed after applying map Π. The classic PCA framework can be extended using semidefinite programming to construct an algorithm for *fair PCA* performance [6]. Figure 2 shows that fair PCA yields an intuitively fairer outcome.

Most research on impartial AI has focused on the learning problem, and much work remains in the development of algorithms for unbiased decision-making. The fundamental difficulty is that fairness is inherently nonconvex. One can solve typical learning and decision-making problems by minimizing a convex loss function; however, fairness essentially asks the opposite — maximizing the independence of predictions and decisions from protected classes. Thus, the development of effective and fair AI techniques will require direct confrontation of the underlying nonconvexity in various instances of the problem.



(a) Original data

(b) Regular PCA

(c) Fair PCA

Figure 2. *Motivation for fairness in unsupervised learning. Dimensionality reduction via fair principal component analysis (PCA) diminishes opportunities for discrimination. The thick red line in 2b and 2c is the optimal linear support vector machine (SVM) that separates by color, and the dotted line is the optimal Gaussian kernel SVM. Figure courtesy of [6].*

References

[1] Angwin, J., & Grassegger, H. (2017). Facebook's secret censorship rules protect white men from hate speech but not black children. *ProPublica*. Retrieved from https://www.propublica.org/article/facebook-hate-speech-censorship-internal-documents-algorithms.

[2] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference* (pp. 214-226). Cambridge, MA: Association for Computing Machinery.

[3] Edwards, L. (2013, March 16). The gender gap in pain. *The New York Times*, p. SR8.

[4] Goh, G., Cotter, A., Gupta, M., & Friedlander, M.P. (2016). Satisfying real-world goals with dataset constraints. In *Advances in Neural Information Processing Systems 29* (*NIPS 2016*) (pp. 2415-2423). Barcelona, Spain.

[5] Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems 29* (*NIPS 2016*) (pp. 3315-3323). Barcelona, Spain.

[6] Olfat, M., & Aswani, A. (2018). Convex formulations for fair principal component analysis. In *33rd AAAI Conference on Artificial Intelligence*. Honolulu, Hawaii. To appear.

[7] Olfat, M., & Aswani, A. (2018). Spectral algorithms for computing fair support vector machines. In *21st International Conference on Artificial Intelligence and Statistics*. Playa Blanca, Lanzarote.

[8] Tuy, H. (1998). *Convex Analysis and Global Optimization*. In *Advances in Natural and Technological Hazards Research* (Vol. 22). Boston, MA: Springer.

[9] Zafar, M.B., Valera, I., Rodriguez, M.G., & Gummadi, K.P. (2017). Fairness constraints: Mechanisms for fair classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Fort Lauderdale, FL.

Anil Aswani is an assistant professor in Industrial Engineering and Operations Research (IEOR) at the University of California, Berkeley. His current research focuses on developing methodologies for robust data-driven decision-making, particularly for applications in healthcare. Matt Olfat is a Ph.D. candidate in IEOR at UC Berkeley. He holds a B.S. in systems engineering and mathematics from the University of Virginia, and his research interests include fairness and robustness in machine learning and decompositions of high-dimensional datasets.

A Near-perfect Heat Exchange

MATHEMATICAL CURIOSITIES

By Mark Levi

When two objects are brought into thermal contact, their temperatures converge. And because temperatures have no inertia, they cannot "overshoot" and reverse. But nature found a remarkable way to overcome this constraint of the second law of thermodynamics and (nearly) exchange the two substances' temperatures (see Figure 1).

The Construction

The two liquids flowing in opposite directions are separated by a heat-conducting membrane. I claim that the temperatures of the two fluids will nearly reverse: 0° water will heat to $100^\circ - \varepsilon$, while 100° water will cool to ε degrees, with an arbitrarily-small ε ($\varepsilon = 1^\circ$ in Figure 1).

How it Works

Figure 1 depicts a discrete approximation: we imagine each fluid split into n cells and replace the continuous motion with a jerky one. We first allow temperatures in adjacent cells across the membrane to settle to the common one (these will be slightly different in practice, contributing to ε in the preceding paragraph), and then we let each fluid quickly advance by one cell. This brings T_{k-1} and T_{k+1} into contact with one another. They settle to the common new temperature

$$T_k^+ = \frac{1}{2}(T_{k-1} + T_{k+1}) \qquad (1)$$

(the top will actually be slightly colder but we ignore this; we also ignore heat exchange between cells in the same pipe). In short, (1) is a discretization (in space and time) of the heat equation. Indeed, we can write it as

$$T_k^+ - T_k = \frac{1}{2}(T_{k-1} - 2T_k + T_{k+1}),$$
$$k = 1, \ldots n,$$

with $T_0 = 0$ (a new cell at 0° enters from the left) and $T_{n+1} = 100$ (a new cell at 100° enters from the right). The temperature will advance towards a linear profile regardless of initial temperature distribution; the first cell will thus approach $T_1 = 100/(n+1)$ while the last cell will approach $T_n = 100 - 100/(n+1)$. A larger n means a more perfect temperature swap.

The key to the operation of the heat exchanger is that *the temperature differences are small in every heat exchange*: $T_{k+1} \approx T_{k-1}$. This proximity of temperatures makes for a small entropy increase. When heat Q passes from an object at temperature T_a to an object at temperature $T_b < T_a$, the entropy of the system consisting of these two objects increases by

$$Q\Big(\frac{1}{T_b} - \frac{1}{T_a}\Big) = Q\,\frac{T_a - T_b}{T_a T_b},$$

a small amount if $T_a \approx T_b$, even if Q is not small (here, T is the absolute temperature, *not* the centigrade). Since $T_{k+1} \approx T_{k-1}$, the heat exchanger increases entropy by less than an unintelligent design would.

Speaking of intelligent design, biological evolution showed intelligence in "inventing" the heat exchanger. For example, deep veins in our arms run along arteries, just like the two tubes in Figure 1 — the top tube represents a vein and the bottom one represents an artery. In cold weather, the outgoing arterial blood warms the cold venous blood coming from the hands; this helps maintain core body temperature.

The entropy's near-constancy signals the near-reversibility of the process [1]. The latter is also directly apparent from our ability to run Figure 1's outputs through another exchanger and nearly return to the original temperatures (e.g., to 2° and 98°).

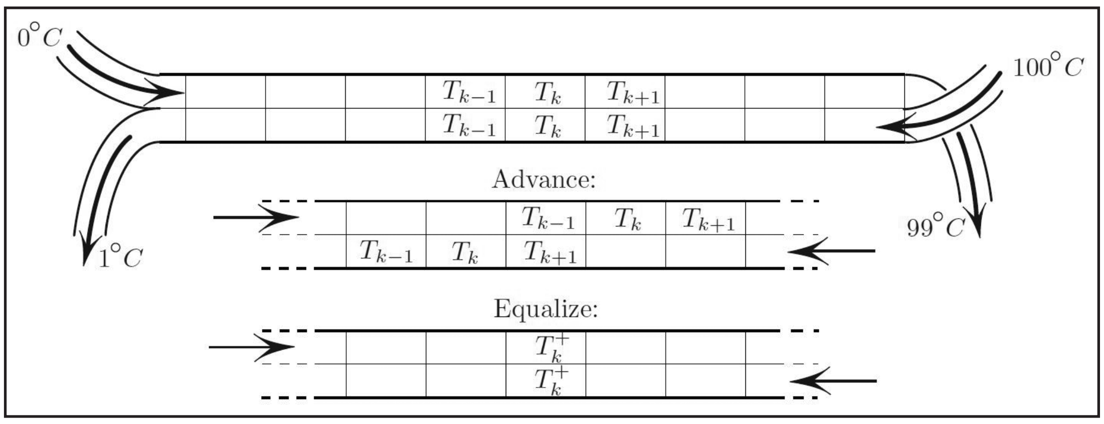


Figure 1. *Beginning with equalized temperatures in adjacent cells (top), the cells advance (middle) and the temperatures of adjacent cells (nearly) equalize. This completes the cycle. Figure courtesy of Mark Levi.*

References

[1] Feynman, R., Leighton, R., & Sands, M. (1964). Entropy. In *The Feynman Lectures on Physics* (Vol. 1). The California Institute of Technology.

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

Deep Learning

Continued from page 1

This leaves the crucial question of computability. What parameters will quickly describe a large family of CPL functions? Linear finite elements start with a triangular mesh. But specifying many individual nodes in $\mathbf{R}^p$ is expensive. It would be better if those nodes are the *intersections* of a smaller number of lines (or hyperplanes). Please note that a regular grid is too simple.

Figure 1 is a first construction of a piecewise linear function of the data vector $\boldsymbol{v}$. Choose a matrix A and vector $\boldsymbol{b}$. Then set to 0 (this is the nonlinear step) all negative components of $A\boldsymbol{v} + \boldsymbol{b}$. Then multiply by a matrix C to produce the output $\boldsymbol{w} = F(\boldsymbol{v}) = C(A\boldsymbol{v} + \boldsymbol{b})_+$. That vector $(A\boldsymbol{v} + \boldsymbol{b})_+$ forms a "hidden layer" between the input $\boldsymbol{v}$ and the output $\boldsymbol{w}$.

The nonlinear function called $\mathrm{ReLU}(x) = x_+ = \max(x, 0)$ was originally smoothed into a logistic curve like $1/(1+e^{-x})$. It was reasonable to think that continuous derivatives would help in optimizing the weights A, $\boldsymbol{b}$, C. That proved to be wrong.

The graph of each component of $(A\boldsymbol{v} + \boldsymbol{b})_+$ has two half-planes (one is flat, from the 0s where $A\boldsymbol{v} + \boldsymbol{b}$ is negative). If A is q by p, the input space $\mathbf{R}^p$ is sliced by q hyperplanes into r pieces. We can count those pieces! This measures the "expressivity" of the overall function $F(\boldsymbol{v})$. The formula from combinatorics is

$$r(q, p) = \binom{q}{0} + \binom{q}{1} + \cdots + \binom{q}{p}.$$

This number gives an impression of the graph of F. But our function is not yet sufficiently expressive, and one more idea is needed.

Here is the indispensable ingredient in the learning function F. The best way to create complex functions from simple functions is by *composition*. Each F_i is linear (or affine), followed by the nonlinear ReLU : $F_i(\boldsymbol{v}) = (A_i\boldsymbol{v} + \boldsymbol{b}_i)_+$. Their composition is $F(\boldsymbol{v}) = F_L(F_{L-1}(\ldots F_2(F_1(\boldsymbol{v}))))$. We now have $L-1$ hidden layers before the final output layer. The network becomes deeper as L increases. That depth can grow quickly for convolutional nets (with banded Toeplitz matrices A).

The great optimization problem of deep learning is to compute weights A_i and $\boldsymbol{b}_i$ that will make the outputs $F(\boldsymbol{v})$ nearly correct — close to the digit $w(\boldsymbol{v})$ that the image $\boldsymbol{v}$ represents. This problem of minimizing some measure of $F(\boldsymbol{v}) - w(\boldsymbol{v})$ is solved by following a gradient downhill. The gradient of this complicated function is computed by *backpropagation*: the workhorse of deep learning that executes the chain rule.

A historic competition in 2012 was to identify the 1.2 million images collected in ImageNet. The breakthrough neural network in AlexNet had 60 million weights in eight layers. Its accuracy (after five days of stochastic gradient descent) cut in half the next best error rate. Deep learning had arrived.

Our goal here was to identify continuous piecewise linear functions as powerful approximators. That family is also convenient — closed under addition and maximization and composition. The magic is that the learning function $F(A_i, \boldsymbol{b}_i, \boldsymbol{v})$ gives accurate results on images $\boldsymbol{v}$ that F has never seen.

This article is published with very light edits.

Gilbert Strang teaches linear algebra at the Massachusetts Institute of Technology. A description of the January 2019 textbook "Linear Algebra and Learning from Data" is available at math.mit.edu/learningfromdata.

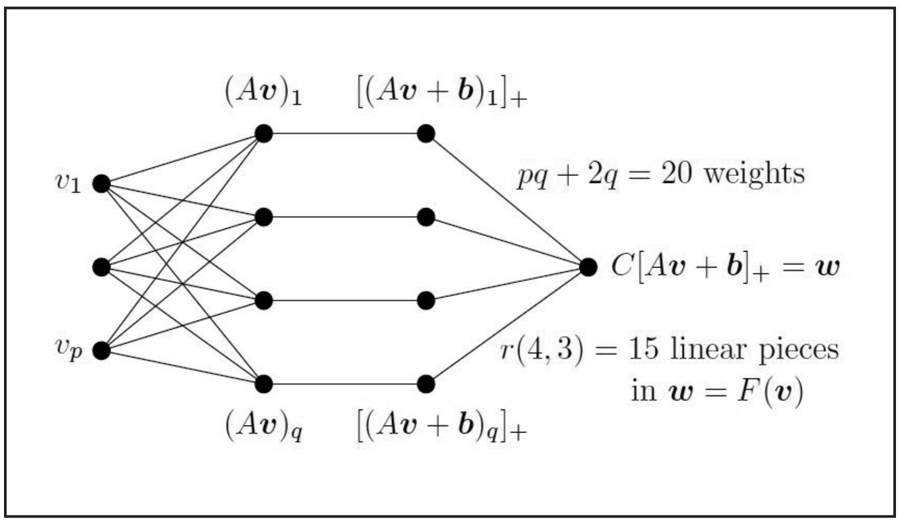
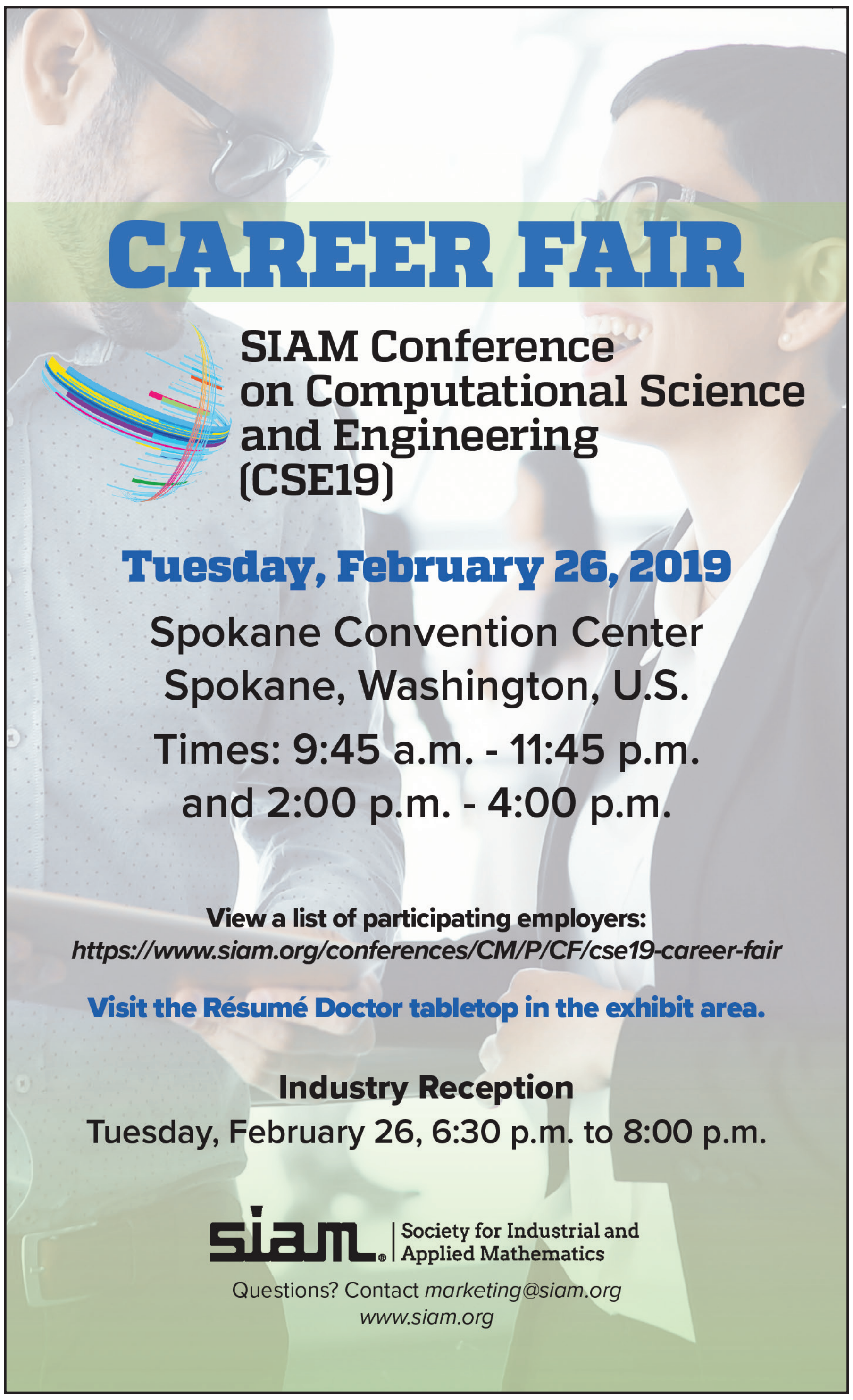
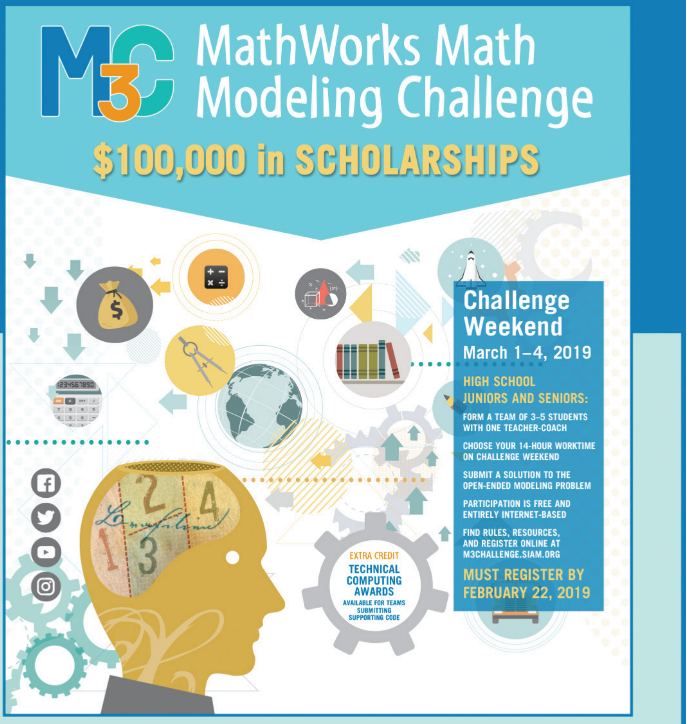

$(A\boldsymbol{v})_1$
$[(A\boldsymbol{v}+\boldsymbol{b})_1]_+$
v_1
v_p
$pq + 2q = 20$ weights
$C[A\boldsymbol{v}+\boldsymbol{b}]_+ = \boldsymbol{w}$
$r(4,3) = 15$ linear pieces
in $\boldsymbol{w} = F(\boldsymbol{v})$
$(A\boldsymbol{v})_q$
$[(A\boldsymbol{v}+\boldsymbol{b})_q]_+$

Figure 1. *Neural net construction of a piecewise linear function of the data vector* $\boldsymbol{v}$.



GOT A PROBLEM?

SIAM is Seeking Problem Ideas for National High School Math Modeling Competition



M3 MathWorks Math Modeling Challenge

$100,000 in SCHOLARSHIPS

Challenge Weekend

March 1–4, 2019

HIGH SCHOOL JUNIORS AND SENIORS:

FORM A TEAM OF 3–5 STUDENTS WITH ONE TEACHER-COACH

CHOOSE YOUR 14-HOUR WORKTIME ON CHALLENGE WEEKEND

SUBMIT A SOLUTION TO THE OPEN-ENDED MODELING PROBLEM

PARTICIPATION IS FREE AND ENTIRELY INTERNET-BASED

FIND RULES, RESOURCES, AND REGISTER ONLINE AT M3CHALLENGE.SIAM.ORG

MUST REGISTER BY FEBRUARY 22, 2019

EXTRA CREDIT
TECHNICAL COMPUTING AWARDS
AVAILABLE FOR TEAMS SUBMITTING SUPPORTING CODE

MathWorks Math Modeling (M3) Challenge is an Internet-based, applied mathematics contest for high school juniors and seniors. M3 Challenge takes place each year in March. Teams of 3–5 students are given 14 hours to solve an open-ended, applied math-modeling problem related to a real-world issue. Winners will receive college scholarships totaling $100,000. Registration and participation are free.

The goal of the Challenge is to motivate students to study and pursue careers in STEM disciplines, especially applied mathematics, computational science, economics, and finance. The problem is revealed to students only after they login on their selected Challenge day. Solutions are judged on the approach and methods used and the creativity displayed in problem solving and mathematical modeling.

Problem structure

Within the problem statement, there should be three questions for teams to answer:

- Question One: A warm up — every serious team can answer.
- Question Two: The guts — framed so that every team can have some success and many teams will cover it well.
- Question Three: The discriminator — many teams will do something, while only a few will have striking results.
- Data — data that is provided or easily found is desirable to encourage students to use coding and technical computing in solution papers.

Honoraria

- $50 for problems found suitable to add to the M3 Challenge problem reserve "bank"
- $500 for problems selected from the reserve bank to be used as "the" Challenge problem

Required problem characteristics

- Accessibility to 11th and 12th graders
- Suitability for solution in 14 hours
- Possibility for significant mathematical modeling
- Topic of current interest involving interdisciplinary problem solving and critical thinking skills
- Availability of enough data for a variety of approaches and depth of solutions (but no easily found answers)
- References identified that will be helpful for getting students started
- Submitted problem idea in the format of previous Challenge problems

Submit your ideas: *M3Challenge.siam.org/suggest-problems*

View previous problem statements: M3Challenge.siam.org/resources/sample-problems

Contact SIAM for more information: m3challenge@siam.org

M3 MathWorks Math Modeling Challenge
A CONTEST FOR HIGH SCHOOL STUDENTS

M3Challenge.siam.org

Organized by siam. Society for Industrial and Applied Mathematics

Sponsored by MathWorks®

The National Association of Secondary School Principals has placed this program on the NASSP National Advisory List of Student Contests and Activities since 2010.

APPROVED FOR LISTING NASSP 2018–2019 STUDENT CONTESTS, PROGRAMS & ACTIVITIES

Randomized Projection Methods in Linear Algebra and Data Analysis

By Per-Gunnar Martinsson

The management and analysis of large complex datasets give rise to mathematical and algorithmic challenges that have intrigued applied mathematicians in recent years. A common situation is a dataset that apparently lives in high-dimensional space, even though evidence suggests that it is interpretable as having a lower "intrinsic" dimension. Finding a way to represent such a dataset using fewer parameters enables its efficient storage, transmission, and analysis to uncover additional structures.

Informally, one might say that the difficulty that arises as a problem's dimensionality grows is that there are just too many places to look. The unit cube in $\mathbb{R}^n$ has 2^n corners, which is a lot even for, say, $n = 100$. Curiously, one's best bet when facing a vast search space such as this is often to conduct the search completely randomly. I will illustrate how this works in practice by reviewing two recently developed algorithms — one for factorizing matrices and the other for finding approximate nearest neighbors.

Linear Approximation (Principal Component Analysis)

A classical and well-understood dimension reduction problem—given a set of points $\{\mathbf{a}_j\}_{j=1}^n$ in $\mathbb{R}^m$—is to find a linear subspace L that to some degree of approximation contains all points. A common technique for finding L is to form the $m \times n$ matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ and compute its singular value decomposition (SVD)

$$\mathbf{A} = \sum_{j=1}^{p} \sigma_j \mathbf{u}_j \mathbf{v}_j^*,$$

where $p = \min(m,n)$, $\{\sigma_j\}_{j=1}^p$ are the singular values of $\mathbf{A}$ (ordered so $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$), and $\{\mathbf{u}_j\}_{j=1}^p$ and $\{\mathbf{v}_j\}_{j=1}^p$ are the left and right singular vectors of $\mathbf{A}$. For any $k \in \{1,2,\ldots,p\}$, the subspace L spanned by the first k left singular vectors $\{\mathbf{u}_j\}_{j=1}^k$ is in certain regards "optimal."

The aforementioned problem arises in many applications. An example occurs in statistics, when one attempts to fit observed high-dimensional data to an assumption that such data represent samples drawn from a multivariate normal distribution driven by a small number of significant components.

Randomized SVD

The problem of finding an optimal linear subspace has a clean and simple mathematical solution, but a user must determine how to compute the dominant k singular values and singular vectors of the given data matrix $\mathbf{A}$. Standard linear algebra routines, such as those in LAPACK, often return the *full* decomposition. This computation's flop count, $O(mn\min(m,n))$, is typically unaffordable for large matrices. Alternatively, one could compute a partial factorization using an algorithm like column pivoted QR or Arnoldi. While this would reduce the asymptotic flop count to $O(mnk)$, these algorithms can be difficult to efficiently implement on modern communication-constrained hardware.

Another option, which is often highly competitive, is the recently-developed *randomized SVD* (RSVD) [1], which draws a Gaussian random matrix $\mathbf{G}$ of size $n \times k$ to create a *sampling matrix* $\mathbf{Y} = \mathbf{AG}$. In many situations, the columns of matrix $\mathbf{Y}$ form a good approximate basis for the column space of $\mathbf{A}$. To identify approximations to the dominant modes of an SVD of $\mathbf{A}$, one must orthonormalize $\mathbf{Y}$'s columns to find a basis for its column space before projecting the matrix $\mathbf{A}$ down to this subspace and using deterministic methods to compute a full SVD of the resulting small matrix.

To attain accuracy comparable to the best possible approximation of rank k, the RSVD must draw a few extra samples. To this end, let us select an oversampling parameter $p\,(p = 5$ or $p = 10$ are generally good) and proceed as follows:

(1) Draw an $n \times (k+p)$ Gaussian random matrix $\mathbf{G}$

(2) Form the $m \times (k+p)$ sampling matrix $\mathbf{Y} = \mathbf{AG}$

(3) Form an $m \times (k+p)$ orthonormal matrix $\mathbf{Q}$, such that $\mathbf{Y} = \mathbf{QR}$

(4) Form the $(k+p) \times n$ matrix $\mathbf{B} = \mathbf{Q}^*\mathbf{A}$

(5) Compute a (small) SVD $\mathbf{B} = \widehat{\mathbf{U}}\,\mathbf{D}\,\mathbf{V}^*$

(6) Form the matrix $\mathbf{U} = \mathbf{Q}\,\widehat{\mathbf{U}}$.

The result is an approximate factorization $\mathbf{A} \approx \mathbf{UDV}^*$, where the diagonal matrix $\mathbf{D}$ holds approximations to the singular vectors of $\mathbf{A}$ and the tall, thin matrices $\mathbf{U}$ and $\mathbf{V}$ have orthonormal columns. The RSVD has the same $O(mnk)$ complexity as existing techniques for partial SVD computation. However, its interaction with the matrix $\mathbf{A}$ via two matrix-matrix multiplications offers compelling advantages, as it often leads to high practical execution speed. Furthermore, it eases the method's use in situations where $\mathbf{A}$ may be stored on slow memory, such as a hard drive, or across a distributed computing system.

The RSVD produces a factorization with near-optimal accuracy in scenarios where $\mathbf{A}$'s singular values decay reasonably rapidly. In the case of matrices whose singular values decay slowly, such as measured statistical data, combining the RSVD with a small number of steps of classical subspace iteration is imperative. In practice, we would then replace the computation on the aforementioned line (2) with something along the lines of $\mathbf{Y} = \mathbf{A}(\mathbf{A}^*(\mathbf{AG}))$ [1].

Variations of the RSVD

The basic algorithmic template of the RSVD is easily adaptable to specialized environments. For instance, one can reorganize the computation in such a way that the algorithm only needs to see each entry of the matrix *once*; this enables the processing of huge matrices resistant to storage [1]. Replacing the Gaussian random matrix with a structured random projection, such as a subsampled and randomized FFT [1], can reduce the flop count from $O(mnk)$ to $O(mn\log k)$.

See **Randomized Projection** *on page 8*

Digital Animation

Continued from page 3

The general conservation laws for mass and momentum govern these physics-based models. For materials, these equations are PDEs that describe changes in the materials' velocity vector field $v(x,t)$ and density $\rho(x,t)$:

$$\frac{Dv}{Dt} = \frac{1}{\rho}\nabla\cdot\underline{\sigma} + g\,\frac{D\rho}{Dt} + \rho\nabla\cdot v = 0,$$

where $\mathbf{g}$ is the gravitational force vector, $\underline{\sigma}(x,t)$ is the material's stress tensor, and

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + v\cdot\nabla$$

is the convective derivative operator. The choice of stress tensor determines which specific physical system is described.

To simulate snow, Teran and collaborators animated cubes of a Jell-O-like substance, adjusting elastic and plastic parameters to visualize how the cubes bounced or stuck together. These cubes—though very unlike snow in a broad sense—formed the basis of the mathematical description of snow's flow, incorporating frictional forces between snow grains.

Once a software PDE solver fast enough for animation became available after several years of development, the elastoplastic formulation reduced rendering time by a massive amount. Frames of the film that would have previously entailed 40 minutes of generation time with other methods required only three to four minutes using PDEs.

Elastoplastic models are general enough to describe other materials that are useful for CGI. Teran showed his audience simulations of water interacting with sand to demonstrate how the water gradually wears away a sand barrier until it collapses. Like snow, sand is granular and exhibits small-scale behavior governed by moisture content, grain size, and frictional interactions between grains. Wet sand can also be packed (into sandcastles, for instance), though less durably than snow.

Cloth and Deformed Potatoes

Frozen aside, most movies do not require many snow scenes. However, the majority of animated films have human characters who sport hair and wear clothing. These systems are both extremely familiar and very complicated to visually simulate (compare the characters' blocky hair in early animated films like *Toy Story* or *Shrek* to modern movies like *Moana* that use advanced physics models). Elastoplastic models can also visually describe these phenomena, despite hair and clothing's dissimilarity to snow or sand.

Teran noted that these systems can employ the same PDE solver as snow simulations. Unfortunately, clothes are not intrinsically granular, which makes them computationally much more expensive. If one treats them as a mesh of particles, the fabric texture constrains the relative positions of those particles. In addition, the external forces change constantly as different patches of cloth come in contact with other cloth, skin, and various objects. From a modeling perspective, cloth is almost *always* deformed; it creases, flaps in the wind, and clings when wet (see Figure 2, on page 3).

The total mathematical approach of Teran and his colleagues is a mixture of the Lagrangian picture—treating the forces on the material's individual particles or fibers—and the Eulerian fluid continuum view. The low-level perspective addresses stresses, various frictional forces, and plasticity with simple physical models, similar to those taught in introductory physics classes. The high-level fluid perspective provides the bulk elastic and plastic properties of the material.

The mathematical model for the material's deformation involves a flow map operator ϕ (which is invertible) taking the fabric from its original configuration $\Omega_0 = \{X\}$ to its final configuration $\Omega_t = \{x\}$, where $\mathbf{x}$ and X are the set of initial and final particle positions:

$$\phi(\cdot,t) : \Omega_0 \to \Omega_t \text{ and } x = \phi(X,t).$$

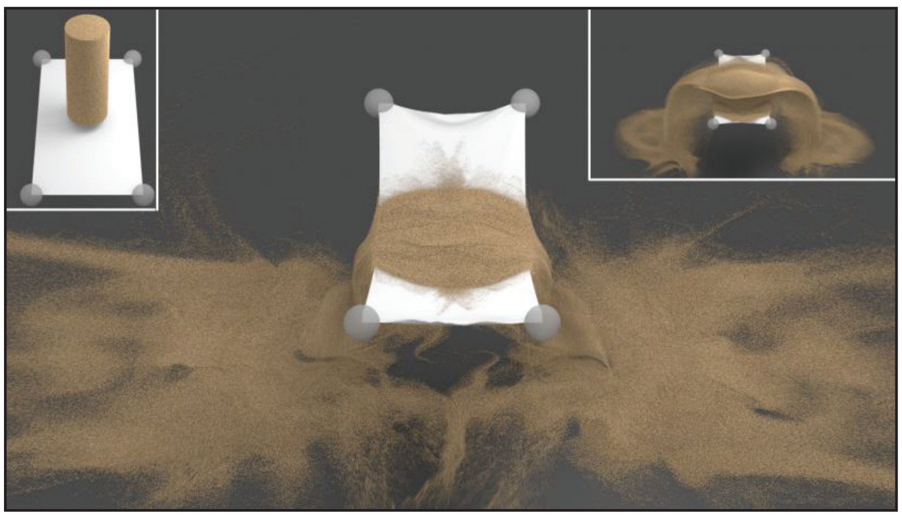


Figure 3. *Two-way coupling between a piece of elastic cloth and seven million grains of sand. Image courtesy of [1].*

Teran described the geometrical process as "mapping a potato onto a deformed potato." The system's physics is encapsulated in the Jacobian or "deformation gradient" F and its determinant J:

$$F(X,t) = \frac{\partial\phi}{\partial X}\,J(X,t) = \det\big(F(X,t)\big).$$

The conservation laws and material properties in the elastoplastic model are connected to this mapping. One can linearize the system to simplify the math during each step of the deformation.

The model applies all external and internal forces acting on the cloth, mapping the motion and constraints on each grain. If the forces acting on a particle are physically unreasonable, the calculation employs constraints to restore the particle to an allowable configuration. In other words, every particle that begins in the fabric must end in the fabric in more or less the same position relative to its neighbors; this prevents unphysical deformations to the material.

The resulting elastoplastic model is amazingly powerful, allowing realistic simulation of fabrics from heavy carpets and cable-knit sweaters to light silk cloth. Teran displayed animations of sand pouring over fabric that used the elastoplastic model for both materials (see Figure 3).

As with many numerical approximations to continuous systems, the accuracy of the elastoplastic simulation depends on the coarseness of the mesh that models the fabric. If the mesh is too coarse or too fine, the simulated fabric behaves incorrectly. Similarly, the types of constraints necessary to make the fabric behave appropriately are similar to the unrealistic imaginary springs that some simulations utilize for similar tasks in other animations.

Nevertheless, the ability of real physics to produce more realistic animations with lower computational costs, even when the particular physics does not naïvely seem to describe the system at hand, is intriguing. With future advances in graphics processing, animators will have an even greater ability to simulate the world, paving the way for increasingly imaginative stories.

References

[1] Jiang, C., Gast, T., & Teran, J. (2017). Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Trans. Graph., 36*(4), 152:1-152:14

Matthew R. Francis is a physicist, science writer, public speaker, educator, and frequent wearer of jaunty hats. His website is BowlerHatScience.org.

Progress by Accident: Some Reflections on My Career

By Walter Gautschi

Walter Gautschi, professor emeritus at Purdue University and a leading mathematician in the areas of approximation theory, orthogonal polynomials, special functions, and numerical analysis, celebrated his 90th birthday in December 2017. A conference honoring this occasion was held at Purdue University earlier this year.[1] In the following article, Gautschi describes how different research areas sparked his interest. An extended version of this article is available online.[2]

Often in my career, my interest in the mathematical areas in which I was active came about by chance occurrences that at the time seemed rather insignificant but were reinforced by later events.

Ordinary Differential Equations

Numerical ordinary differential equations (ODEs) piqued my interest during my first semester at the University of Basel. I enrolled in a course on "Wissenschaftliches Rechnen" (scientific computation), in which Professor Eduard Batschelet mentioned a graphical method for solving ODEs courtesy of Richard Grammel. The method uses polar coordinates: the argument x of the solution y serving as the polar angle and the reciprocal of the solution $1/y(x)$ plotted on the radius vector with angle x. It struck me as odd that the reciprocal of the solution was being approximated. Why not the solution itself?

It turned out that a geometric construction similar to the one used by Grammel indeed exists to approximate the solution. I mentioned this to Batschelet, who was pleased by my observation. He must have mentioned this to Alexander Ostrowski, who encouraged me to expand my work on Grammel's method into a Ph.D. thesis. I was not thrilled with this suggestion, knowing that the digital computer era—which was just beginning—would demand numerical methods rather than graphical ones. But I made the most of it and developed techniques for analyzing the error of Grammel's method.

CAREERS IN MATHEMATICAL SCIENCES

Aside from my thesis work, I also studied numerical methods; two early events stimulated my interest. One was the appearance of Lothar Collatz's book on ODEs, which I eagerly studied from cover to cover. The other pertained to Ludwig Bieberbach's visit to our university. In his 1923 book on ODEs, Bieberbach had stated without proof an estimate for the local truncation error of the classical Runge-Kutta method for a single first-order differential equation. He worked out the full proof—also for systems of differential equations—during his visit and published it in an early issue of *Zeitschrift für Angewandte Mathematik und Physik* (*ZAMP*) in 1951. A few years earlier, Rudolf Zurmühl had published Runge-Kutta methods that directly integrate single differential equations of nth order, i.e., without first decomposing into a system of first-order equations. I decided to apply Bieberbach's techniques to Runge-Kutta-Zurmühl methods to obtain local error bounds for all derivatives of order $< n$. It was a laborious undertaking—a real tour de force—but I persisted and published the results in *ZAMP* in 1955.

Throughout my years at Oak Ridge National Laboratory (ORNL)—and still later—I was teaching myself Russian and reading Russian books and papers about approximation and computation (recall that these were the years after the Sputnik launch). This inspired me to examine numerical methods for ODEs based on trigonometric, rather than algebraic, polynomials with the expectation that they could possibly be used to solve differential equations with oscillatory solutions. I published a paper on this work in 1961, but it did not immediately have the resonance that I had hoped it would. It took some 40 years until the paper was recognized as anticipating what in the meantime had been called "exponentially fitted" methods.

Special Functions

When I arrived at the National Bureau of Standards in Washington, D.C. (now the National Institute of Standards and Technology) in 1956, a major project entailed the preparation of the *Handbook of Mathematical Functions*. Milton Abramowitz, who was directing the project, asked me whether I would be interested in writing the as-yet unassigned chapter on the error function. I accepted, despite my complete lack of experience with special functions; I felt that my background in classical analysis was strong enough for me to be up to the task. I diligently began to study the literature on special functions, particularly the confluent hypergeometric function. At Milton's request, I also helped with the chapter on the exponential integral.

Orthogonal Polynomials and Gaussian Quadrature

During my time at ORNL in the late 1950s and early 1960s, an ORNL chemist asked Alston Householder if a member of his group could help with the computation of a definite integral that resisted accurate evaluation. Householder felt that I was the best person for the job. The integral in question turned out to be an integral over $[-1, 1]$ with a logarithmically singular factor in the denominator, something like π^2 plus the square of $\log(1+x)(1-x)$. That seemed easy enough: use Gaussian quadrature with the reciprocal of this singular factor as a weight function and take n, the number

See **Progress by Accident** *on page 8*

[1] https://www.cs.purdue.edu/sca/

[2] https://sinews.siam.org/Details-Page/progress-by-accident-some-reflections-on-my-career

Professional Opportunities and Announcements

Send copy for classified advertisements and announcements to ***marketing@siam.org****.*
For rates, deadlines, and ad specifications, visit ***www.siam.org/advertising****.*

*Students (and others) in search of information about careers in the mathematical sciences can click on "Careers" at the SIAM website (****www.siam.org****) or proceed directly to* ***www.siam.org/careers****.*

The Chinese University of Hong Kong, Shenzhen

The School of Science and Engineering

Located in the Longgang District of Shenzhen, the Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen) is a research-intensive university, established in 2014 through a Mainland-Hong Kong collaboration with generous support from the Shenzhen Municipal Government. It inherits the fine academic traditions of the Chinese University of Hong Kong, and will develop its academic programmes in phases and offer courses in the School of Science and Engineering, the School of Management and Economics, and the School of Humanities and Social Science. English is the main language for course instruction, and students will receive degrees from the Chinese University of Hong Kong. At present, several research centers have been established in the School of Science and Engineering, including the Arieh Warshel Institute of Computational Biology, Kobilka Institute of Innovative Drug Discovery, Hopcroft Institute for Advanced Study in Information Sciences, and Shenzhen Key Laboratory of Semiconductor Lasers.

Post Specifications: The School of Science and Engineering (**http://sse.cuhk.edu.cn/en**) invites applications for faculty positions—professor/associate professor/assistant professor/lecturer—in focused areas of **statistics**, **data science**, **mathematics**, **financial engineering and quantitative finance**, and **bioinformatics**, though excellent applicants in all related areas will be considered.

Junior applicants should have (i) a Ph.D. degree (by the time of reporting for duty) in related fields and (ii) high potential in teaching and research. Candidates for senior posts (associate and full professor) are expected to have demonstrated academic leadership and strong commitment to excellence in teaching, research, and services. Junior appointments will normally be made on contract basis for up to three years initially, leading to longer-term appointment or tenure later — subject to review. Exceptional appointments with tenure will be considered for candidates of proven excellence. Applicants are encouraged to check out details about the university at **http://www.cuhk.edu.cn/en**.

Salary and Fringe Benefits: Salary will be comparable to international standards, commensurate with experience and accomplishments. Appointments will be made under the establishment of CUHK-Shenzhen, and employee benefits will be provided according to the relevant labor laws of Mainland China as well as CUHK-Shenzhen regulations. Subsidies from various government-sponsored talent programs will also be made available for eligible candidates: **http://www.cuhk.edu.cn/UploadFiles/talentsprogramoutline.pdf**.

An application package, including a CV and personal statements in teaching and research, as well as contact information of three references who will write recommendation letters on behalf of the candidate, should be sent by email to **Talents4SSE@cuhk.edu.cn**. All applicants need to specify the rank(s) of the position being applied to in their application cover letters. Upon submission of applications, applicants should request three recommendation letters to be directly sent to **Talents4SSE@cuhk.edu.cn**.

Randomized Projection

Continued from page 6

The Johnson-Lindenstrauss Lemma and Applications to Nonlinear Problems

While linear approximation is a tremendously powerful tool, many applications require techniques that are inherently nonlinear. Let us again consider a cloud of points $\{\mathbf{a}_j\}_{j=1}^n$ in $\mathbb{R}^m$, where m is large but information suggests that the points belong to some nonlinear manifold of lower dimension. To further complicate things, we must typically also account for noise in the model, meaning that each point may shift slightly away from the true lower-dimensional structure. The problem of discovering this structure and representing it in a user-friendly way is a subject of vigorous research. Key challenges include determining how to computationally resolve certain recurring subtasks—such as clustering the points—or identifying each point's nearest neighbors to form a connectivity graph. Standard algorithms for resolving these tasks scale unfavorably as the dimension m of the ambient space increases, which naturally leads to a search for embeddings of the points into lower-dimensional spaces that preserve important geometric properties. A classical result in this direction is the Johnson-Lindenstrauss lemma, which states that for any $\epsilon > 0$ and any dimension d satisfying

$$d \geq 4\left(\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}\right)^{-1} \log(n),$$

there exists a map $f : \mathbb{R}^m \rightarrow \mathbb{R}^d$ that approximately preserves distances in the sense that

$$(1-\epsilon)\|\mathbf{a}_i - \mathbf{a}_j\|^2 \leq \|f(\mathbf{a}_i) - f(\mathbf{a}_j)\|^2$$
$$\leq (1+\epsilon)\|\mathbf{a}_i - \mathbf{a}_j\|^2$$

for every pair $\{\mathbf{a}_i, \mathbf{a}_j\}$ of points. The map f is often built via stochastic techniques (e.g., $f(\mathbf{x}) = d^{-1/2}\mathbf{G}\mathbf{x}$, where $\mathbf{G}$ is again a Gaussian random matrix, now of size $d \times n$).

The Johnson-Lindenstrauss lemma is compelling from a theoretical point of view; the reduced dimension d depends exclusively on the number of points n (not the dimension m of the ambient space), and the dependence is only logarithmic with respect to n. But from a *practical* point of view, the lemma is by itself of limited use — the scaling with respect to the distortion parameter ϵ is ghastly. For instance, if $\epsilon = 0.1$ then $d \gtrapprox 800\log(n)$, which is far too high a dimension for most classical algorithms to be practicable and still leaves substantial distortions of 10 percent. While it is certainly possible to improve the original Johnson-Lindenstrauss results, most known randomized dimension reduction techniques either exhibit very sizable distortions or an image space of unacceptably high dimension.

Randomized Nearest Neighbor Search and a Two-stage Sampling Strategy

A delightful algorithm proposed by Peter Jones, Andrei Osipov, and Vladimir Rokhlin [2]—designed to solve the aforementioned approximate nearest neighbor problem—illustrates how to harness the power of randomized projections without badly sacrificing accuracy. Given a small integer k and a set of points $\{\mathbf{a}_j\}_{j=1}^n$ in $\mathbb{R}^m$, where m is large, let us find each point's k nearest neighbors. This seems ideally suited for a Johnson-Lindenstrauss projection-type approach. However, as we have seen, if d is small enough for classical methods, the introduced distortions would produce many mistaken nearest neighbors. While a single instance of the outlined procedure is exceedingly unreliable, the situation improves dramatically with several repetitions of the experiment [2]. Say we draw 10 different random projections and perform a low-dimensional search for each one. Every search yields a list of candidates for any given point's nearest neighbors. Now we simply form for each point the union of these lists. Finally, we return to the original dataset to compute the actual distances in $\mathbb{R}^m$, keep the k best results, and discard the rest.

Both the RSVD and the randomized nearest neighbor search manage to find accurate results despite randomized projections' introduction of substantial distortions. Perhaps the trick is a two-stage recipe:

(1) Use randomized projections to inexpensively develop a rough sketch that tells us where to look

(2) Use high-precision deterministic methods to compute a precise answer.

The second step allows one to aggressively oversample in the first stage, worrying only about minimizing the risk of missing important information. Any unnecessary information is then filtered out in the second stage.

Here I have described two randomized algorithms designed to help process large datasets in high-dimensional spaces. These algorithms are typically more accurate than traditional randomized algorithms in scientific computing such as Monte Carlo. One may perhaps argue that they are more closely related to methods like randomized QuickSort, in that the algorithm's randomized aspect primarily affects the runtime. Algorithms of this type reliably produce highly accurate answers and are forgiving in terms of random number generation.

References

[1] Halko, N., Martinsson, P.-G., & Tropp, J.A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev., 53*(2), 217-288.

[2] Jones, P.W., Osipov, A., & Rokhlin, V. (2013). A randomized approximate nearest neighbors algorithm. *Appl. Comp. Harm. Anal., 34*(3), 415-444.

Per-Gunnar Martinsson holds the W.A. "Tex" Moncrief, Jr. Chair in Simulation-Based Engineering and Sciences at the University of Texas at Austin, where he is also a professor of mathematics. He has previously held faculty positions at the University of Oxford, the University of Colorado at Boulder, and Yale University.

Progress by Accident

Continued from page 7

of Gauss points, large enough to yield any desired accuracy. Having carefully studied Francis Hildebrand's book on numerical analysis, I knew how the required orthogonal polynomials could be generated from the moments of the weight function and was able to compute them to any order. With full confidence, I wrote the necessary short program and ran it on ORACLE, the world's fastest computer in 1954. I failed miserably! Investigating the underlying reason for my failure—ill conditioning—gave rise to many papers on the constructive theory of orthogonal polynomials.

The 150th anniversary of Elwin Christoffel's birth in 1979 greatly intensified my preoccupation with orthogonal polynomials — especially Gaussian quadrature. Christoffel was instrumental in generalizing the Gaussian quadrature rule to arbitrary weight functions, and developed the underlying theoretical machinery involving orthogonal polynomials. The speaker slated to give the plenary talk on Christoffel's contributions to numerical integration for the occasion had to withdraw unexpectedly, and I was asked to step in. Against all odds I pulled it off and presented the lecture, which was published in 1981.

History

I have continually enjoyed reading about the masters of centuries past, and whenever I suspected that some contemporary results may have been realized much earlier, perhaps in the 19th century, I eagerly dug into the older literature to confirm my hunches. The first time this happened was in connection with a result in Oskar Perron's book on computing solutions to three-term recurrences using continued fractions, which I speculated was much older. Despite perusing many books on difference equations, I could not find any mention of this result. But I did find many references to Italian mathematician Salvatore Pincherle, so I browsed through Pincherle's collected works. There, in an 1894 paper on hypergeometric functions, I found exactly the result that Perron stated in his book. I attributed this result to Pincherle, and it became known as Pincherle's theorem.

I have always admired Leonhard Euler. During a visit to Basel in the early 2000s, Emil Fellmann, a well-known Euler biographer, handed me a copy of a letter that Euler had written to his close friend Daniel Bernoulli. It dealt with the somewhat bizarre (and hence failed) attempt to interpolate the common logarithm at all powers of 10. It took me a while to figure out what was being described in the letter, but I was eventually able to explain the matter both in a 2008 paper and a short commentary in a correspondence volume. A year earlier, I was invited to speak about Euler on the 300th anniversary of his birth at the 2007 International Congress on Industrial and Applied Mathematics, which took place in Zurich. It took me—not really an expert on Euler's work and life—a whole year to prepare the talk, an expanded version of which appeared in *SIAM Review*[3] in 2008.

Acknowledgments: Thanks to Alex Pothen for his help in editing this article.

Walter Gautschi is professor emeritus of computer science and mathematics at Purdue University.

[3] https://epubs.siam.org/doi/pdf/10.1137/070702710

Walter Gautschi celebrates his 90th birthday at the Purdue Conference on Scientific Computing and Approximation, which was held in his honor earlier this year. Image credit: Kristyn R. Childres.