Swarthmore College

Works

Mathematics & Statistics Faculty Works

Mathematics & Statistics

1-1-1985

The Algorithmic Way Of Life Is Best

Stephen B. Maurer , '67
Swarthmore College, smaurer1@swarthmore.edu

Recommended Citation

Stephen B. Maurer , '67. (1985). "The Algorithmic Way Of Life Is Best". *College Mathematics Journal*. Volume 16, Issue 1. 2-5. DOI: 10.2307/2686615
https://works.swarthmore.edu/fac-math-stat/73

The Algorithm Way of Life Is Best
Author(s): Stephen B. Maurer
Source: *The College Mathematics Journal*, Vol. 16, No. 1 (Jan., 1985), pp. 2-5
Published by: Mathematical Association of America
Stable URL: http://www.jstor.org/stable/2686615
Accessed: 19-10-2016 15:32 UTC

The Algorithmic Way of Life is Best

Stephen B. Maurer

Stephen B. Maurer is Associate Professor of Mathematics at Swarthmore College, where he has taught since 1979. He grew up in Silver Spring, Maryland. He received a B.A. from Swarthmore in 1967 and a Ph.D. at Princeton in 1972. He has taught previously at Princeton, the University of Waterloo, Hampshire College, and the Phillips Exeter Academy. His major scholarly interest has been research and curricular development in discrete mathematics, but he has also made forays—sometimes continuous—into mathematical biology, economics, and anthropology. He is an MAA Visting Lecturer, Chairman of the MAA Committee on High School Contests, and a member of CUPM. Professor Maurer is a consultant for the Alfred P. Sloan Foundation, working on projects to increase the role of discrete mathematics in the mathematics curriculum and the role of mathematical and technological "modes of thought" in the general curriculum.

> "Whenever you can settle a question by explicit construction, be not satisfied with purely existential arguments."
>
> Hermann Weyl, 1946

Weyl was referring to the well-known penchant of mathematicians to rigidly separate showing that solutions exist from actually finding them. The former endeavor is viewed as a pure, high-minded occupation, abounding in elegance if abstract methods are used. The latter is seen as an applied, rather disreputable activity, to be done on the side if done at all. Weyl was saying that this schism, and the esthetic behind it, are a mistake.

Weyl's advice has not been heeded much. Yet today, an even stronger admonition deserves our support: ***seek the algorithmic way of life.***

Algorithms go farther than constructions: they involve systematic, mostly iterative procedures that are effective for a wide range of inputs, as opposed to one-time, ad hoc manipulations. Moreover, the algorithmic way of life means much more than merely finding solutions and finding them by algorithms; it means adopting an algorithmic frame of mind towards all aspects of one's mathematical work. It means, as Weyl suggested, trying to fuse computation with proof—developing the theory from the algorithms instead of separately or vice versa. It even means letting one's interest in algorithms suggest the questions one will try to answer.

Thinking with and about algorithms can unify all one's mathematical endeavors. It also extends the range of one's mathematics, and provides as satisfying an esthetic as the old existential esthetic—if one will only let it.

None of this makes much sense without some examples. Let us illustrate the existential and algorithmic outlooks by the manner in which they address the following basic question of linear algebra: what is the structure of the solutions to a system of linear equations $Ax = b$?

Under the old existential esthetic, one might proceed as follows: Consider the preimage of a point under a linear transformation. This is either the kernel or a translate of it. Since the kernel is a vector space, it has a basis and consists of all linear combinations of that basis. In conclusion, the set of solutions consists of a fixed vector (possibly $\mathbf{0}$) plus arbitrary linear combinations of a set of basis vectors. All statements in this argument are basic theorems of pure linear algebra and (under the old regime) have already been proved abstractly.

The only problem is: what if one actually needs to *find* the solutions to a specific system $Ax = b$? The argument above isn't any help. How does one actually find a basis? How does one find the appropriate translate?

At this point, in the classical scheme, Gaussian elimination is introduced, somewhat testily, and one or two examples are worked out to show that one can obtain answers this way if one must.

This is being a bit unfair. While it is true that many of the older books don't introduce Gaussian elimination until after all the theory is done, today almost all books do a chapter on solving linear equations first, as motivation for the theory that follows.

But the schism is still there. Most of these books go right back to existential methods to derive the theory. They don't show that the same Gaussian elimination algorithm also allows you to *prove* those theorems.

For instance, why is the solution set a translate of a vector space? One who has carefully studied Gaussian elimination as an algorithm will be very familiar with the structure of the final matrix and the resulting form in which the solutions are written. That form is: a fixed vector plus arbitrary linear combinations of some other vectors. Ergo, the solutions are a translate of a vector space.

What about theorems which are not directly related to the structure of solutions? For instance, consider the theorem that any two bases of a (finite dimensional) vector space have the same cardinality. One can prove this by Gaussian elimination too. Suppose the first basis had fewer vectors than the second. If the second basis is written as columns in terms of its coefficients relative to the first basis, one gets a coefficient matrix B. Now imagine doing Gaussian elimination on B. Since there are more columns than rows, at least one of the columns does not become a unit column (one 1 and the rest 0's) in the final form. Since dependencies among the columns are not affected by row reduction, it follows that the columns of B are dependent.

Fleshed out, this argument may be longer than the classical existential argument, but it is very natural if one is thoroughly familiar with Gaussian elimination and if one is accustomed to using algorithms to try to answer questions. It also has the esthetic value of killing two birds with one stone—the same tool which allows one to do computations also allows one to answer theoretical questions.

Thinking in terms of algorithms leads to questions about linear algebra that one would never have thought to ask before. How many steps does it take to solve a problem by Gaussian elimination? Is this more, or less, than it takes by Gauss–Jordan reduction? Can one prove that the better of these is a best possible method for general linear equations? Are there better methods for special matrices? For instance, one can change the order in which one clears out columns; for a sparse matrix, this might make a big difference. Can one determine the optimal order in advance? These last questions, by the way, are matters of current research.

One reason Gaussian elimination is rarely used in texts for proving things is because it's never explained very clearly. Rather, it is merely illustrated by examples. This is not surprising, since traditional notation is not very good for describing algorithms. But if algorithms are going to be used as the building blocks of theory, they must be defined precisely and careful proofs must be given that they terminate and work. For this, one needs algorithmic language and mathematical induction, the latter used in some unfamiliar ways. Thus, the algorithmic approach gives some old building blocks new prominence.

Of course, one can't do everything by algorithms. For instance, one can't select an infinite basis with a finite algorithm; hence one can't use such algorithms to prove that such bases exist. Nevertheless, thinking about things in terms of algorithmic steps can aid understanding in the infinite domain as well. We offer some examples below. Moreover, we now have nonstandard analysis, which provides new ways to do proofs in infinite settings using finitistic methods. This suggests even further use for algorithmic thinking.

A good amount of continuous mathematics can be approached algorithmically by standard methods, if one uses limits. Take Brouwer's Fixed Point Theorem. The traditional proof is existential and seems like magic: the appropriate homology group wouldn't map right if there isn't a fixed point. But there is another proof. It uses a discrete analog called Sperner's Lemma: if a simplex is triangulated and labeled with very minimal restrictions, there must always be a fully labeled subsimplex. By associating a labeling with any continuous function, and going to the limit, one gets a fixed point. Moreover, Sperner's Lemma has a proof by algorithm. There is a simple algorithm which terminates only if a fully labeled simplex is found and which does always terminate. If one is familiar with how the algorithm works, termination is obvious; one can prove it simply by following what the algorithm does. This algorithm (and various generalizations) are actually used when fixed points need to be found. This too is a subject of current research.

Can the algorithmic frame of mind affect calculus? Of course. Consider how to define the definite integral. One common way is to show there is a unique function which satisfies certain axioms of area—various inequalities and the usual formula for rectangles. The definite integral is defined to be this function. Another approach is to define it as the limit of Riemann sums. The former is slicker. But is it better? Using the latter means that the way most integrals are actually computed is also the way they are defined. (Remember, most functions which actually appear in the world can't be integrated in closed form, or even stated that way.) Therefore, Riemann sums become both a computational and a theoretical tool.

Or consider the definition of a real number. Dedekind cuts are charming, but try to use them to locate a number one doesn't already thoroughly understand. Convergent sequences, on the other hand, can actually be used to compute numbers to any desired accuracy.

Now look at the definition of derivative. Epsilons and deltas have so befuddled students that instructors shirk from teaching this approach anymore. How does one algorithmically approximate the derivative? Not by considering all epsilons and deltas, but with a convergent sequence of difference quotients. Why not use the difference calculus, too long neglected at the freshman level, to develop differentiation theory as well as to do computations?

Or look at differentiation formally—as a symbolic operation on closed-form expressions. From the algorithmic point of view, all the theorems about differentiating sums, products, etc., are really all parts of one theorem: there is an algorithm for differentiating every elementary function. Viewed this way, all sorts of new, interesting questions arise. One can't apply those rules without recognizing how to "parse" the elementary functions. How can this be done systematically? What is the order of complexity of the task?

My suggestions about definitions in calculus probably seem objectionable. The definitions I have called "slick" are, according to current tastes, short and sweet. Also, developing theory through algorithms has not been part of our training, and thus seems unnatural. However, this unnaturalness is easily overcome by practice. As for current tastes, recall that computer programmers used to pride themselves in finding the shortest possible program for solving a problem. The ideal was a solution in one line. Now it's been realized that this is folly; those one-line solutions are too slick for people to understand or extend with ease. Similarly, algorithmic definitions, though longer than traditional definitions, often allow for ease of understanding and use. This is just one more reason why the algorithmic way of life offers so much, both esthetically and operationally.

Let us heed Hermann Weyl, in the strongest algorithmic terms.

One Needs More than the Algorithmic Approach . . .

R. G. Douglas, *SUNY at Stony Brook, Stony Brook, NY*

I share Maurer's impatience with existence proofs when more informative constructive proofs are available. We should deplore the slick axiomatic definition which, while "clearer," has nothing to do with the form or context in which the notion was first grappled with or understood. And I agree that the esthetics of the professional mathematician should not dictate the content or form of undergraduate mathematics courses. However, I believe Maurer has pushed his argument too far.

I take issue with his claim that the algorithmic way is the *only* way for teaching, for understanding, and even for doing research mathematics. There is more, much more, and to leave out that part which doesn't yield to this approach would be worse than the present situation.

For example, in calculus one should certainly learn algorithmic approaches to calculating derivatives and integrals. But, in addition, one should learn that linear approximation is the central idea of the differential calculus, and one should understand the central role played by the notion of area in the integral calculus. Moreover, one should realize "discovery" of the calculus does not date from the understanding of the differential or of the integral calculus, but rather of understanding the relation between them. Finally, one should learn something about the behavior of "nice functions," both analytically and geometrically, and of the role of differential equations in describing systems that change. All of this is difficult to approach algorithmically; it involves the development and enhancement of the native intuition present in all students. Being able to calculate derivatives and integrals means little if the student has no understanding of their purpose. ***While it may be true that an algorithmic definition can help in understanding the definition of the derivative, mathematicians thought about the notion of the derivative for a very long time before giving any definition, algorithmic or otherwise.***

Although perhaps not as obvious, a similar argument can be made for linear algebra. ***The ideas and notions of linear algebra are even more subtle than those of the calculus. That it took several hundred years more for these ideas to be understood and explicated than it did for those of the calculus is not accidental.*** Anyone who has taught linear algebra has encountered students who can determine if a set of vectors is linearly independent or forms a basis without having any idea what either means. Obviously one wants both, and the algorithmic approach can be coupled effectively