

## EWD: MAKING IT SIMPLE IS NOT EASY

Francis Sullivan, Editor in Chief



**E**DSGER WYBE DIJKSTRA, WHO DIED 6 AUGUST, SIGNED HIS ELEGANTLY COMPOSED NOTES WITH A SIMPLE “EWD” PLUS A SEQUENCE NUMBER. THE MAIN TOPIC OF THESE NOTES WAS ALGORITHMS—DISCOVERING THEM, EXPLAINING THEM, AND EXPRESSING THEM IN THE SIMPLEST AND MOST

rigorous form. The final expression of the algorithm usually contained a proof of its correctness. Dijkstra’s notes, the bulk of which have not been published (but some of which are available at [www.cs.utexas.edu/users/EWD](http://www.cs.utexas.edu/users/EWD)), ranged over many topics beyond the algorithms themselves. One of the last ones, EWD1310, describes a long trip from Austin, Texas, to several European destinations that combined lectures at major universities with visits to old friends in the Netherlands and elsewhere. Reading his musings on the state of the Dutch railroads, the experience of advancing age and its associated infirmities, memories of the war, and the relationships between Western European and American culture is illuminating and, at the same time, deeply moving. When reading Dijkstra, the impression is that you’re in the company of a vivid personality.

Dijkstra, who sometimes liked to call himself a programmer, was trained initially in mathematics and physics and later received a PhD in computer science. But his wonderful writing style and his unique way of communicating his vision always seemed to me to be that of a philosopher and artist. Like some of the best philosophers, he anticipated objections to his reasoning and refuted the objections in the text itself. This is the essence of his method of proving algorithms correct. Certain objections to his work, however, are merely silly and really can’t be refuted because they’re about style rather than content—for example, some said he made too many personal remarks or that his handwriting was too beautiful. He must be the only scientist ever to have had his hand-

writing turned into a font as part of a joke posted on thousands of bulletin boards.

By now, all computational scientists know some of his creations, such as the algorithms for the shortest path or for the minimum spanning tree. Other algorithms, like the one for computing the length of the longest increasing subsequence, are better known among mathematicians. And still others, such as smoothsort, are not as well known as they should be. I think I learned about smoothsort from some lectures Dijkstra gave at the University of Maryland many years ago. It was this set of lectures that made it clear to me that beauty in algorithms was a good and desirable thing. I also learned that there really is nothing extraneous in Dijkstra’s algorithms.

Smoothsort has the remarkable feature of running in linear time on an almost sorted list and in time  $n \log n$  on a random list. I once used it as part of the neighbors-finding component of a molecular dynamics code, because the order of neighbors shouldn’t change much between time steps or else the time steps are too large. The method worked just as expected, but I’d neglected to include one check that I had decided was not really needed. Several years later, when machines got faster and problems got bigger, a colleague began to have trouble with the code. Of course, the problem was the omitted test, which didn’t kick in until the lists got quite large. I fixed my mistake and returned the code to my collaborator, but he’d already replaced my routine by another sort. “Improve” on Dijkstra’s methods at your peril!

## New Editorial Board Members




**Dietrich Stauffer** is a professor at the Institute for Theoretical Physics, Cologne University, Germany. His technical interests include Monte Carlo simulations of statistical physics and interdisciplinary models. He received a PhD from Munich

Technical University. He has served on many editorial boards, including those for *Computers in Physics*, *European Physical Journal B*, *International Journal of Modern Physics C*, *Journal of Physics A*, *Journal of Statistical Physics*, *Physica A*, *Physics World*, *Theory in Biosciences*, and *Quantitative Finance*. Contact him at the Inst. for Theoretical Physics, Cologne Univ., 50923 Köln, Germany; [stauffer@thp.Uni-Koeln.de](mailto:stauffer@thp.Uni-Koeln.de); [www.thp.uni-koeln.de/personal/stauffer.html](http://www.thp.uni-koeln.de/personal/stauffer.html).



**Stephen G. Nash** is the associate dean for graduate studies and research in the School of Information Technology and Engineering at George Mason University, where he is also a professor of systems engineering and operations research. His research inter-

ests focus on numerical analysis, specifically optimization and linear algebra. He received his BSc in mathematics from the University of Alberta, Canada, and his PhD in computer science from Stanford University. He is a member of SIAM, the ACM, and the American Society for Engineering Education. Contact him at the School of Information Technology and Eng., MS 5C8, George Mason Univ., Fairfax, VA 22030; [snash@gmu.edu](mailto:snash@gmu.edu); <http://iris.gmu.edu/~snash>.

Dijkstra was born in 1930. That puts him in the generation between those who were adults at the start of World War II, called the greatest generation, and those born after January 1943, called the baby boom generation. In the US, Dijkstra's age group is called the silent generation. They're so silent that there hasn't even been a US president elected in their (our!) age bracket. Members of this silent generation are supposed to observe certain unspoken rules of behavior and deportment while never saying exactly what those rules are and what the consequences of breaking them might be. After thinking about Dijkstra's radical originality and great influence on the development of computational science, I've decided that I'd prefer this group to be called the "not noisy" generation or, perhaps even better, the "you'd do well to listen carefully" generation. 



# Stop !

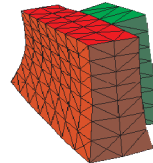
Don't buy a different software product for every modeling problem.

With FlexPDE you can solve whole families of 2D and 3D field problems!

Like electromagnetics, heatflow, deformation, and lots more. Steady-state, time-dependent, time-harmonic or eigenvalues.



Open Tube  
Combustion Reactor



3D Multi-material  
Thermoelasticity



T-Septate Waveguide

With a redesigned user interface, FlexPDE 3.0 is more than ever the indispensable tool for scientists and engineers • FlexPDE 3.0 is a scripted finite element model builder for partial differential equations systems • Linear or Nonlinear • 2D or 3D plus time or eigenvalues • Unlimited number of variables • Windows, Unix, Linux or Mac OSX  
• Download a demo from our website

## PDE Solutions Inc

P.O Box 4217 • Antioch, CA • 94531  
925-776-2407 • FAX 925-776-2406  
[www.pdesolutions.com](http://www.pdesolutions.com)