HOW RIGHT YOU ARE!

Francis Sullivan, Editor in Chief

DO YOU BELIEVE IN YOUR SOFTWARE'S OUTPUT? HOW MUCH DO YOU TRUST IT? THESE QUESTIONS FORCE US TO THINK ABOUT COMPUTATION IN A DIFFERENT AND IMPORTANT WAY. IN FACT, UNDERSTANDING HOW MUCH TO TRUST THE RESULTS IS AS IMPORTANT AS THE RESULTS THEMSELVES.

Here are four different "degrees of belief" that I might have for results of my own computations, depending on the purpose and kind of calculation.

(a) I'm Certain It's Right

I reserve this category for cases in which an absolute check of the result is available and carrying out the check is much easier than the calculation itself. The standard example is sorting a set of integers. Checking that the resulting list is non-decreasing is easy. To take another example, I am certain that $8 * 7$ is equal to 1, modulo 11 simply because $8 * 7 = 56$. Less obvious but just as certain is the fact that $523 * 955 = 1$ modulo 991. Here, I generate the solution to $523 * x = 1 \bmod 991$ using the Euclidean GCD algorithm and then check it by multiplying. I can even solve an integer-linear system $Ax = b \bmod 991$ exactly because the results are integers, so there's no round off, which means that the matrix multiplication is exact.

More common for scientific computing are floating-point calculations, where round off clearly matters but where I also can estimate how much error will be generated. This brings us to class...

(b) I'm Confident This Result Is as Right as Available Computational Precision Can Make It

Here, I include things like solving a linear system of modest size, in which the "modest size" of today was known as "huge" three years ago. This category also covers other extremely well-understood numerical techniques, such as computation of square roots using Newton's method or evaluation of well-understood functions such as sin and log.

(c) This Looks Good

This means that I can believe in the gross features of the output but perhaps not in all the details. Almost all physical modeling falls in this category. Here, "looks good" means that I'm willing to fly in an aircraft that has been designed computationally, without any wind-tunnel testing, but it doesn't mean that I believe every detail of the shedding of vortices from the wing tips.

(d) Who Knows?

This is a very large class. It includes attempts at computational simulation of all complex systems that actually evolve faster than we can compute or are governed by nonlinear rules that offer no intuitive hints about what's going on.

A nice example from Marc Kac is the following: Suppose you place a set of n balls at uniform intervals around the circumference of a circle and then add m other marks at random. Color the n balls green initially and assume that the random marks remain fixed while the n balls rotate clockwise. Every time a uniform ball passes a random mark, the ball changes color: if the ball is green, it turns red and if red, it turns green. If you take a snapshot at any time, the colors appear to be random. However, because you're just rotating a circle, you know that the sequence of color changes must be periodic. So, if you look at just some of the output, you might think you're looking at random changes, but if you look at still more of the output, it's clearly periodic. Nonetheless, knowing it's periodic doesn't tell you how it was generated and whether the generation algorithm really does implement Kac's idea.

 Copublished by the IEEE CS and the AIP 1521-9615/03/$17.00 © 2003 IEEE

Opinions on which are the most interesting categories depend on your viewpoint. Someone doing computational modeling of physical systems might be most interested in (b) and (c) whereas a specialist in the uses of computing in "pure" mathematics might like (a) or (d). However, the boundaries between these classes are not sharp. The final example based on "Kac's Ring" crops up in studies of chaos and nonequilibrium statistical mechanics.

I'd modify Davy Crockett's motto, "Be sure you're right and then go ahead," to, "Know how right you are and then go ahead!"

How to Reach CiSE

Writers

For detailed information on submitting articles, write to cise@computer.org or visit http://computer.org/cise/edguide.htm.

Letters to the Editors

Send letters to Jenny Ferrero, Contact Editor, jferrero@computer.org. Please provide an email address or daytime phone number with your letter.

On the Web

Access http://computer.org/cise or http://ojps.aip.org/cise for information about CiSE.

Subscription Change of Address (IEEE/CS)

Send change-of-address requests for magazine subscriptions to address.change@ieee.org. Be sure to specify CiSE.

Subscription Change of Address (AIP)

Send general subscription and refund inquiries to subs@aip.org.

Subscribe

Visit http://ojps.aip.org/cise/subscrib.html or http://computer.org/subscribe.

Missing or Damaged Copies

If you are missing an issue or you received a damaged copy (IEEE/CS), contact membership@computer.org. For AIP subscribers, contact kgentilie@aip.org.

Reprints of Articles

For price information or to order reprints, send email to cise@computer.org or fax +1 714 821 4010.

Reprint Permission

To obtain permission to reprint an article, contact William Hagen, IEEE Copyrights and Trademarks Manager, at whagen@ieee.org.