What happened to the `flops` command?

James F. Epperson

June 7, 2015

When I first started this book project—long time ago, in the late 1990s—I was determined to make it software-independent. As far as I was concerned, a student or teacher or class could use Fortran or C or Matlab or whatever. My goal was to write a text explaining the mathematics, not a programming manual.

Well. By the time I sat down to write the Second Edition (actually the third version of the book) much had changed. It was impossible to ignore the place that Matlab had taken in the scientific computing marketplace, so I made this edition more "cozy" with Matlab. It caused a few problems—as a non-academic author, I did not have ready access to a modern Matlab installation; the one that I did have was a Student Edition from 1994, which I had managed to install on my laptops up until my newest machine, which uses Windows7. But I thought it was best to acknowledge the way things were.

Then reality attacked me. On Feb. 25, 2015, I got an email from Prof. Kathleen Shannon of Salisbury State University in Delaware, who had written me before about some issues with some exercises. (And she was right, so I paid attention to her new email.) She was unable to find the `flops` command on her University's newest Matlab installation. This was news to me, so I did what any 21st Century scholar would do, and went to Google. The reasons are involved, and are explained at the website:

`http://www.mathworks.com/company/newsletters/articles/matlab-incorporates-lapack.html`

(third paragraph from the bottom). I am most flummoxed to discover that this was done some time ago—in fact, before the First Edition was even published.

There is a lesson here regarding numerical analysis and methods: There has always been a symbiosis between hardware and algorithms—as the hardware got better, what was important about the algorithms changed. Modern hardware is simply *fast*. So fast that the floating point operations are less important than memory access and cache usage. This is especially important in numerical linear algebra, where the notion of "flop count" was traditionally important. However, the author is still of the opinion that a flop count is potentially important for comparison purposes—an algorithm that consistently takes fewer flops for a given accuracy is to be preferred to one that takes more flops. For this (educational) reason I wish the command had been retained.

The impact on the mathematics in the text is minimal, except for the several exercises (especially in Chapter 7) that depended on the `flops` command. I am properly chagrined, and take this as a lesson in the truth of what I say in §1.4: "In short, if you don't like the present state of affairs with regard to computing languages, wait around a little while—it will change."

A third-party `flops` command is available; see:

`http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/`

The author has no experience with this package and does not vouch for it in any way.

I will set to work preparing alternate exercises. I fear it will take a while.