

[Nick Higham](#)

The Top 10 Algorithms in Applied Mathematics

In the January / February 2000 issue of *Computing in Science and Engineering*, Jack Dongarra and Francis Sullivan chose the “10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century” and presented a **group of articles** (<http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=17639&punumber=5992>) on them that they had commissioned and edited. (A **SIAM News article** (<http://www.siam.org/news/news.php?id=637>) by Barry Cipra gives a summary for anyone who does not have access to the original articles). This top ten list has attracted a lot of interest.

Sixteen years later, I thought it would be interesting to produce such a list in a different way and see how it compares with the original top ten. My unscientific—but well defined— way of doing so is to determine which algorithms have the most page locators in the index of **The Princeton Companion to Applied Mathematics** (<http://www.maths.manchester.ac.uk/~higham/pcam/index.php>) (PCAM). This is a flawed measure for several reasons. First, the book focuses on applied mathematics, so some algorithms included in the original list may be outside its scope, though the book takes a broad view of the subject and includes many articles about applications and about topics on the interface with other areas. Second, the content is selective and the book does not attempt to cover all of applied mathematics. Third, the number of page locators is not necessarily a good measure of importance. However, the index was prepared by a professional indexer, so it should reflect the content of the book fairly objectively.

A problem facing anyone who compiles such a list is to define what is meant by “algorithm”. Where does one draw the line between an algorithm and a technique? For a simple example, is putting a rational function in partial fraction form an algorithm? In compiling the following list I have erred on the side of inclusion. This top ten list is in decreasing order of the number of page locators.

1. Newton and quasi-Newton methods
2. Matrix factorizations (LU, Cholesky, QR)
3. Singular value decomposition, QR and QZ algorithms
4. Monte-Carlo methods
5. Fast Fourier transform

Algorithm 3 (Newton’s method).

```

Given initial iterate  $u^0$ 
for  $k = 1, 2, \dots$  until convergence do
  Solve  $J(u^{k-1})\delta u = -F(u^{k-1})$ 
  Update  $u^k = u^{k-1} + \delta u$ 
end for

```

From “Computational Science” by David E. Keyes in Princeton Companion to Applied Mathematics

6. Krylov subspace methods (conjugate gradients, Lanczos, GMRES, minres)
7. JPEG
8. PageRank
9. Simplex algorithm
10. Kalman filter

Note that JPEG (1992) and PageRank (1998) were youngsters in 2000, but all the other algorithms date back at least to the 1960s.

By comparison, the 2000 list is, in chronological order (no other ordering was given)

- Metropolis algorithm for Monte Carlo
- Simplex method for linear programming
- Krylov subspace iteration methods
- The decompositional approach to matrix computations
- The Fortran optimizing compiler
- QR algorithm for computing eigenvalues
- Quicksort algorithm for sorting
- Fast Fourier transform
- Integer relation detection
- Fast multipole method

The two lists agree in 7 of their entries. The differences are:

PCAM list	2000 list
Newton and quasi-Newton methods	The Fortran Optimizing Compiler
Jpeg	Quicksort algorithm for sorting
PageRank	Integer relation detection
Kalman filter	Fast multipole method

Of those in the right-hand column, Fortran is in the index of PCAM and would have made the list, but so would C, MATLAB, etc., and I draw the line at including languages and compilers; the fast multipole method nearly made the PCAM table; and quicksort and integer relation detection both have one page locator in the PCAM index.

There is a remarkable agreement between the two lists! Dongarra and Sullivan say they knew that “whatever we came up with in the end, it would be controversial”. Their top ten has certainly stimulated some debate, but I don’t think it has been too controversial. This comparison suggests that Dongarra and Sullivan did a pretty good job, and one that has stood the test of time well.

Finally, I point readers to a talk **Who invented the great numerical algorithms?** (<https://people.maths.ox.ac.uk/trefethen/talks.html>) by Nick Trefethen for a historical perspective on algorithms, including most of those mentioned above.

Posted on **March 29, 2016** by **Nick Higham** Posted in **Princeton Companion**

One thought on “The Top 10 Algorithms in Applied Mathematics”

1. **DRPARTHA SAYS:**

April 10, 2016 at 5:07 am

Thanks a lot for this wonderfully informative post.

↳ Reply

Website Powered by WordPress.com.