THE TOP 10 ALGORITHMS OF THE 20TH CENTURY

M. Bunjamin*

ABSTRAK

SEPULUH ALGORITMA TERBAIK ABAD 20. Berikut disampaikan daftar dan uraian singkat dari "Sepuluh Algoritma Terbaik Abad-20" yang dipilih oleh redaksi majalah Computing in Science & Engineering / CiSE, dan dijadikan topik bahasan utama majalah ini terbitan Jan / Feb 2000. Ini adalah 10 algoritma yang paling besar pengaruhnya pada pengembangan dan praktek sains dan enjiniring di abad-20, dan merupakan rangkuman pengembangan matematika numerik dan sains komputasi sepanjang abad-20. Kita boleh setuju atau tidak dengan pilihan ini, tetapi paling sedikit kita jangan memandang rendah pilihan yang dilakukan oleh para ahli komputasi kelas dunia di negara maju ini. Redaksi majalah CiSE telah mengajukan pertanyaan kepada para pembacanya tentang bagaimana pendapat dan perasaan mereka mengenai pilihan ini. Dari beberapa terbitan CiSE berikutnya tampak bahwa ternyata tidak ada perlawanan berarti dari para pembaca, sehingga dapat disimpulkan bahwa pilihan ini sudah dilakukan dengan baik dan dapat diterima oleh masyarakat ilmiah internasional.

ABSTRACT

THE TOP 10 ALGORITHMS OF THE 20TH CENTURY. The following is a list and a short description of the "Top 10 Algorithms of the 20th Century" selected by the editors of Computing in Science & Engineering / CiSE, and was the main issue of its Jan-Feb. 2000 edition. These were the 10 algorithms with the greatest influence on the development and practice of science & engineering in the 20th century, and was supposed to be a resume of the development of numerical mathematics and computational science throughout the 20th century. We may agree or disagree with this selection, but the least we should do is not to underestimate them, because they are the opinions of highly qualified computational scientists in advanced countries. The editors have asked the readers of the journal regarding their opinions and feelings about this selection. It turned out that in the following issues of CiSE, there were only agreements and no disagreements regarding the selection. It may be concluded, therefore, that CiSE's selection was well done and well received by the international scientific community.

INTRODUCTION

The following is a list and a short description of the "Top 10 Algorithms of the 20th Century" selected by the editors of Computing in Science & Engineering / CiSE, and was the main issue of its Jan-Feb. 2000 edition. These are the 10 algorithms with

* FMIPA-UI and FMIPA-UNAS

the greatest influence on the development and practice of science & engineering in the 20th century, and is supposed to be a resume of the development of numerical mathematics and computational science throughout the 20th century. We may agree or disagree with this selection, but the least we should do is not to underestimate them, because they are the opinions of highly qualified computational scientists in advanced countries. The editors have asked the readers of the journal regarding their opinions and feelings about this selection. It turned out that in the following issues of CiSE, there were only agreements and no disagreements regarding the selection. It may be concluded, therefore, that CiSE's selection was well done well received by the international scientific community.

INTEGER RELATION DETECTION / IRD

For many years, researchers have dreamt of a facility that lets them recognize a numeric constant in terms of the mathematical formula that it satisfies. With the advent of efficient IRD algorithms, that time has arrived. Let $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ be a vector of real or complex numbers. **x** is said to possess an integer relation if there exists integers a_i (not all zero), such that $a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = 0$. An integer relation algorithm is a practical computational scheme that can recover the vector of integers a_i (if it exists), or can produce bounds within which no integer relation exists. These are the activities of computational number theory. The most effective algorithm for IRD is Ferguson's recently discovered PSLQ algorithm (Ferguson et al., Math. of Computation, Vol. 68, No. 225, Jan. 1999). As an example, the following is a formula found by PSLQ in 1997 that enable us to compute the *n*-th hexadecimal digit of π (CiSE Jan-Feb 2000).

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k}\left[\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right].$$

Another example is the identification of the constant $B_3 = 3.54409035955...$, which is the third bifurcation point of the logistic-map $x_{i+1} = r x_i (1 - x_i)$, which exhibits period doubling shortly before the onset of chaos. To be precise, B_3 is the smallest value of the parameter *r* such that the successive iterates x_i exhibit 8-way periodicity

instead of 4-way periodicity. The constants B_2 and B_1 may be defined analogously. Computations using the predecessor of PSLQ found that B_3 is a root of the equation

$$t^{12} - 12\,t^{11} + 48\,t^{10} - 40\,t^{9} - 193\,t^{8} + 392\,t^{7} +$$

$$44\,t^{6} + 8\,t^{5} - 977\,t^{4} - 604\,t^{3} + 2108\,t^{2} + 4913 = 0\,.$$

Using IRD, researchers have discovered many new facts of mathematics and physics, and these discoveries have in turn led to valuable new insights. This process often called "experimental mathematics" – namely, the utilization of modern computers in the discovery of new mathematical principles – is expected to play a much wider role in both pure and applied mathematics during the 21st century.

THE (DANTZIG) SIMPLEX METHOD FOR LINEAR PROGRAMMING

George Dantzig created a simplex algorithm to solve linear programs for planning and decision-making in large-scale enterprises. The algorithm's success led to a vast array of specializations and generalizations that have dominated practical operations research for half a century.

In one of the simplex method's many forms, the LP problem is:

$$\text{Minimize (w.r.t. } \mathbf{x}) \ \mathbf{c}^{\mathrm{T}}\mathbf{x} \ \text{ subject to } \ \mathbf{A}\mathbf{x} = \mathbf{b} \ \text{ and } \ \mathbf{x} \quad 0.$$

In practice, the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ is partitioned so that the LP problem is written as

$$\begin{aligned} &\text{Minimize (w.r.t. } \mathbf{x}) \ \mathbf{c}^{\mathrm{T}}\mathbf{x}\,, \\ &\quad \text{subject to } \ \mathbf{A}1\mathbf{x} \le \mathbf{b}1 \\ &\qquad\qquad\ \ \mathbf{A}2\mathbf{x} = \mathbf{b}2 \\ &\qquad\qquad\ \ \mathbf{A}3\mathbf{x} \ge \mathbf{b}3 \\ &\qquad\qquad\ \ \mathbf{x} \quad 0 \end{aligned}$$

The use of slack and surplus variables can transform the problems from one to the other form.

Some of the questions when using the simplex method are the following.

- How do we get an initial feasible solution? Or, more importantly, how do we tell if there are no feasible solutions?

- What happens if the constraints don't stop the objective from being reduced? That is, what if the solution is unbounded?
- What happens if several vertices give equal objective values?

Modern textbooks on linear programming solution using simplex method have abandoned the tableaux and the jargons of the earlier works (Gass, Hadley, Hillier & Lieberman, etc).

KRYLOV SUBSPACE ITERATION

Many problems lead to large systems of equations of thousands or even millions of unknowns: electric circuit simulation, magnetic field computation, weather forecast, chemical processes, semiconductor-device simulation, nuclear-reactor safety analysis, mechanical-structure stress, and so on. When we face a large system of linear equations $\mathbf{Ax} = \mathbf{b}$, with the N x N matrix $\mathbf{A}$ and the N vector $\mathbf{b}$ given and the N vector $\mathbf{x}$ unknown, an iterative algorithm such as the successive overrelaxation / SOR method come to our mind. Another iteration method to help solve similar problems was the alternating-direction iterative method, which attempted to solve the discretized partial differential equations over grids in 2-D or 3-D by successively solving 1-D problems in each coordinate direction.

To start an iteration process, we first try to find a good guess for the solution, for instance by solving a much easier nearby (idealized) problem. For this purpose, suppose we approximate the matrix $\mathbf{A}$ of the system $\mathbf{Ax} = \mathbf{b}$ by a simpler matrix $\mathbf{K}$. Then the iteration process may be formulated as follows: in step i+1, solve the new approximation $\mathbf{x}_{i+1}$ for the solution $\mathbf{x}$ of $\mathbf{Ax} = \mathbf{b}$ from $\mathbf{Kx}_{i+1} = \mathbf{Kx}_i + \mathbf{b} - \mathbf{Ax}_i$. In the 1950s Lanczos proposed the idea to keep all approximate computed so far in the iteration process and to recombine them to a better solution. Lanczos recognized that the basic iteration leads to approximate $\mathbf{x}_i$ that are in nicely structured subspaces. These subspaces are spanned by $\mathbf{r}_0$, $\mathbf{Ar}_0$, $\mathbf{A}^2\mathbf{r}_0$, $\cdots$, $\mathbf{A}^{i-1}\mathbf{r}_0$, where $\mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0$. Such a subspace is a Krylov subspace of dimension i for $\mathbf{A}$ and $\mathbf{r}_0$. Using Krylov Subspace Iteration strategy, scientists found several methods such as Conjugate Gradient Method (Hestenes & Stiefel, 1954), GMRES (Saad & Schultz, 1986), and Bi-CGSTAB (van der Vorst, 1992), which are useful for solving $\mathbf{Ax} = \mathbf{b}$ with $N > 1$ million and $\mathbf{A}$ sparse matrix whose elements are mostly zeroes.

THE QR ALGORITHM

It is assumed that we all share the view that the ability for fast computation of matrix eigenvalues is an important tool for modern scientists & engineers. The QR algorithm is a very satisfactory way, but this success does not mean the QR algorithm is the last word on the subject. Matrix theory and eigenvalue problems dramatically increased in importance with the arrival of matrix mechanics and quantum theory, and further in the development of fluid dynamics, nuclear reactor analysis, geophysics, and others.

The basic idea behind the QR algorithm is that any N x N real matrix $\mathbf{A}$ can be decomposed into $\mathbf{A} = \mathbf{A}_1 = \mathbf{Q}_1\mathbf{R}_1$, $\mathbf{Q}_1$ is orthogonal $\Leftrightarrow \mathbf{Q}_1^{\mathrm{T}} = \mathbf{Q}_1^{-1}$, and $\mathbf{R}_1$ is upper-triangular. The next step is to find $\mathbf{A}_2 = \mathbf{R}_1\mathbf{Q}_1 = \mathbf{Q}_1^{-1}\mathbf{A}_1\mathbf{Q}_1$, and then decompose $\mathbf{A}_2 = \mathbf{Q}_2\mathbf{R}_2$, etc. It can be shown that the sequence $\mathbf{A}_1, \mathbf{A}_2, \ldots$ converges to the upper-triangular form of $\mathbf{A}$ with eigenvalues in monotone decreasing order of absolute value down the diagonal. An important property of the QR decomposition is that it preserves the following properties of a matrix: symmetry, tridiagonal form, and Hessenberg form. Because of this property, a general matrix whose eigenvalues have to be found is brought first to a Hessenberg form, say with the aid of Householder method. Once a matrix is in the Hessenberg form, the QR transformation is applied iteratively to take it to converge to a form where the eigenvalues are either isolated on the diagonal or are the eigenvalues of a 2 x 2 submatrix on the diagonal. There are two distinct classes of algorithms for computing the QR decomposition: Gram-Schmidt algorithms and orthogonal triangularization.

SORTING WITH QUICKSORT

Sorting is one of the most studied problems in computer science, because of both its uses in many applications and its intrinsic theoretical importance. The basic sorting problem is the process of rearranging a given collection of items into ascending or descending order. Although researchers have developed and analyzed many sorting algorithms, the Quicksort algorithm stands out. Tony Hoare presented the original algorithm in 1962, and Quicksort is still the best-known practical sorting algorithm.

THE DECOMPOSITIONAL APPROACH TO MATRIX COMPUTATION

There are many matrix decompositions, old and new, and the list of the latter seems to go on growing. Nonetheless, six decompositions hold the center, because they are useful and stable. The six decompositions are described briefly below.

1. The Cholesky Decomposition. Given a positive definite matrix **A**, there is a unique upper triangular matrix **R** with positive diagonal elements such that $\mathbf{A} = \mathbf{R}^{\mathrm{T}}\mathbf{R}$. In this form, the decomposition is known as Cholesky decomposition. It is often written in the form $\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}$, where **D** is diagonal and **L** is a lower triangular matrix with 1s on the diagonal. Cholesky decomposition is used primarily to solve positive definite linear systems.
2. The LU Decomposition. Let **A** is an *N* x *N* matrix. There are permutations **P** and **Q** such that $\mathbf{P}^{\mathrm{T}}\mathbf{A}\mathbf{Q} = \mathbf{L}\mathbf{U}$, where **L** is unit lower triangular and **U** is upper triangular. The matrices **P** and **Q** are not unique, and the process of selecting them is known as pivoting. LU decomposition is used primarily for solving linear systems **Ax = b**, where **A** is a general matrix. Cholesky and LU decompositions can be computed with the help of Gaussian elimination.
3. The QR Decomposition. See section IV above.
4. The Spectral Decomposition. Let **A** is an *N* x *N* symmetric matrix. There is an orthogonal matrix **V** such that $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^{\mathrm{T}}$, where $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_N)$. If $\mathbf{v}_i$ denotes the *i*-th column of **V**, then $\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i$. Thus $(\lambda_i, \mathbf{v}_i)$ is an eigenpair of **A**, and the spectral decomposition of **A** shows the eigenvalues of **A** along with complete orthonormal system of eigenvectors.
5. The Schur Decomposition. Let **A** is an *N* x *N* matrix. There is a unitary matrix **U** such that $\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^{\mathrm{H}}$, where **T** is upper triangular and $\mathbf{U}^{\mathrm{H}}$ is the conjugate transpose of **U**. The diagonal elements of **T** are the eigenvalues of **A**, which, by appropriate choice of **U**, can be made to appear in any order. This decomposition is called the Schur decomposition of **A**. A real matrix can have complex eigenvalues and hence a complex Schur form. By allowing **T** to have a real 2 x 2 blocks on its diagonal that contain its complex eigenvalues, the entire decomposition can be made real. This is sometimes called a real Schur form. After a preliminary reduction to Hessenberg form, the Schur form is computed with the QR algorithm.
6. The Singular Value Decomposition / SVD. Let **A** be an *M* x *N* matrix with $M \geq N$. There are orthogonal matrices **U** and **V** such that $\mathbf{U}^{\mathrm{T}}\mathbf{A}\mathbf{V} = \begin{pmatrix} \mathbf{\acute{O}} \\ 0 \end{pmatrix}$, where $\mathbf{\acute{O}} = \mathrm{diag}(\sigma_1, \cdots, \sigma_N)$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N \geq 0$. This is called the SVD of **A**. If

$\mathbf{U}_\mathbf{A}$ consists of the first *n* columns of **U**, we can write $\mathbf{A} = \mathbf{U}_\mathbf{A}\mathbf{ÓV}^T$, which is sometimes called the singular value factorization of **A**. The diagonal elements σ_i of **S** are called the singular values of **A**. The corresponding columns of **U** and **V** are called the left and right singular vectors of **A**. There are three classes of algorithms to compute the spectral decomposition and the SVD: the QR algorithm, the divide and conquer algorithm, and Jacobi's algorithm.

THE FAST FOURIER TRANSFORM

These days, it is almost beyond belief that there was a time before digital technology. It seems almost everyone realizes that the data whizzing over the Internet, bustling through our modems, or crashing into our cell phones is ultimately just a sequence of 0's and 1's, a digital sequence, that magically makes the world the convenient, high-speed place it is today. Much of this magic is due to a family of algorithms that collectively go by the name the fast Fourier transform / FFT. The FFT is perhaps the most ubiquitous algorithm used today to analyze and manipulate digital or discrete data.

THE METROPOLIS ALGORITHM

The Metropolis algorithm has been the most successful and influential of all the members of the computational species that used to be called the "Monte Carlo Method." Traditionally, the Monte Carlo method was applied using the sampling method called rejection method proposed by Von Neumann. Metropolis et al. in 1953 proposed another method of sampling which is supposed to be much more productive than the rejection method.

THE FORTRAN I COMPILER

The Fortran I compiler was the first demonstration that it is possible to automatically generate efficient machine code from high-level languages. It has thus been enormously influential. A very good technique were used in the Fortran I compiler for the parsing of expressions, loop optimization, and register allocation.

FAST MULTIPOLE ALGORITHM

Problems that involve computing the mutual interactions within large number of particles pervade many branches of science & engineering. When the particles interact through the electrostatic / Coulomb or gravitational / Newton potentials, the long range of the resulting forces creates a computational headache. This is because forces arise for all pairs of charges (or masses) in such a system. Because the number of pairs increases quadratically with the number of particles N included in a simulation, the computational complexity of simulations carried out without a well-designed computational strategy is said to be of order $O(N^2)$. This quickly renders simulation studies of the so called *N*-body problem practically impossible as systems increase in size to levels relevant for realistic problems. This is where the fast multipole algorithm helps.

CLOSING REMARKS

1. Out of the 10 algorithms described above, four of them involve properties and applications of matrices, namely the (Dantzig) simplex method of linear programming, the Krylov subspace iterations, the QR algorithm, and the decomposition approach to matrix computation. This fact shows that the role played by matrices in scientific & engineering computation in the last century was very important. This role includes the solution of a system of linear equations: $\mathbf{Ax} = \mathbf{b}$, and the eigenvalue problem: $\mathbf{Ax} = \lambda\mathbf{x}$, and this seemed to be getting more and more important in the future, especially to meet the future demands to solve linear equations or eigenvalue problems involving "monster" matrices with N in the millions (see for example: Davidson, Monster Matrices, Their Eigenvalues and Eigenvectors, Computers in Physics, 1993, p. 519). The importance of matrix theory may also be seen from the fact that there is a scientific journal especially devoted to it: SIAM Journal of Matrix Analysis and Applications.
2. The mathematical models used in scientific & engineering problem solving consists mostly of linear ODEs, PDEs and integral equation or integrodifferential equation. When we solve these problems using finite-difference or finite-element methods, they are converted into a system of linear equations involving matrices.
3. Mathematics, which in the past used to be highly theoretical in character, and relies completely on deductive reasoning, is now becoming more "experimental" as shown by the Integer Relation Detection algorithms. Therefore, inductive reasoning in mathematics is now quite acceptable by most mathematicians.

4. Methods for solving constrained optimization problems such as Dantzig's simplex algorithm for linear problems are becoming more and more important because of the following fact. In order to survive in a very dynamic world with free global competition, it is demanded from the producers that production and operations processes be executed with maximum efficiency and productivity, of course without sacrificing quality of products.
5. Many scientific problems are multidimensional in character, and therefore deterministic simulations of these systems are not easy. This is where the Monte Carlo simulation came to the rescue, such as for multidimensional integrations. Monte Carlo simulation is also absolutely needed to simulate systems, which is stochastic in character. In all these applications, the Metropolis method is needed to increase the efficiency of the simulation, without sacrificing the validity of the conclusions.
6. Molecular dynamics and molecular mechanics simulations are now becoming more and more important in computational physics, computational chemistry, computational biology, and computational medicine. To make sure that the simulation be as close to reality as possible, the scientists demanded that the number of particles used in the simulation be as big as those occurring in nature. Therefore the demand for algorithms with higher and higher efficiency is practically insatiable. This is where the Fast Multipole algorithm plays its important role.
7. Out of the total of 142 papers cited by the 10 articles, almost 50% were published after 1990, another almost 50% were published between 1950-1990, and the remaining 3% were published before 1950. These facts proved that the top 10 algorithms of the 20^{th} century were mostly developed after 1950, and highly developed and widely applied in the 1990s. These developments were mostly in parallel with the development of high-performance computing systems.