Factorization of polynomials

Arjen K. Lenstra
Centrum voor wiskunde en informatica
Kruislaan 413
1098 SJ Amsterdam
The Netherlands

The last few years there have been a lot of exciting new results in the area of factorization of polynomials. In this note we give an overview of the most important results, and we give some directions for future research.

A long standing open problem was the *Integer linear programming problem with a fixed number of variables.* In 1980 this problem was shown to be polynomially solvable, by an algorithm that relied on methods from *geometry of numbers* [21] (see also [8]). Some of these geometrical methods proved to be useful for the factorization of polynomials as well. In particular an algorithm from [21] to compute reasonably orthogonal bases for integral lattices could be applied to develop a very practical algorithm to factor polynomials over algebraic number fields [13]. Although this algorithm is not polynomial-time, it is the fastest algorithm for this purpose that we know of.

Another application of integral lattices to polynomial factorization consisted of a new algorithm to factor polynomials with integral coefficients, also described in [13]. This is the first algorithm where factors of polynomials are regarded as short vectors in integral lattices, a technique that appeared to have important theoretical consequences.

Unfortunately, this new technique did not yet lead to a polynomial-time algorithm for the factorization of polynomials. The reason for not being polynomial-time was that the algorithm from [21] to reduce bases for integral lattices was not polynomal-time; this reduction algorithm was used to determine approximations of shortest vectors in integral lattices.

In fact, at that time no polynomial-time algorithm to factor polynomials at all was known. Most factorization algorithms were based on the so-called *Berlekamp-Hensel* technique: first compute a sufficiently precise p-adic factorization, then look for suitable combinations of these p-adic factors to find the true factorization. In practice these algorithms perform very well; in theory nothing better than an exponential-time bound can be proved, due to the fact that the p-adic factorization possibly contains considerably more factors than the true factorization (see [5] for a construction of such polynomials).

The situation changed in 1981, when L. Lovász invented a polynomial-time *basis reduction algorithm* for integral lattices. This algorithm makes it possible to compute an approximation of a shortest vector in a lattice in polynomial-time. Combined with the technique from [13], this easily led to a polynomial-time algorithm to factor polynomials with rational coefficients [14].

Together with the results that were obtained by Kaltofen in his Ph.D. thesis [6], this implied polynomial-time factoring algorithms for polynomials in any fixed number of variables, and with integral coefficients. In his method the multivariate problem is reduced in polynomial-time to the univariate case, and the resulting univariate polynomial is factored by means of [14]. Other polynomial-time algorithms for similar polynomial factoring problems that were discovered independently, are described in [4; 11] and in [15; 17; 18; 19]. In the latter papers immediate generalizations of the method from [14] are given (see also [29] in this context).

In theory, and for the dense encoding scheme, the problem of factorization of polynomials is solved by now. For the case of sparsely encoded multivariate polynomials, which is in practice more relevant, the situation is not so very nice. In [28] a (probabilistic) polynomial-time reduction from sparsely encoded multivariate polynomials to polynomials in two variables is given. Clearly, a lot of research should still be carried out in this area.

Also, for polynomials with coefficients in a finite field the problems are not yet solved. Very fast algorihms exist that are expected polynomial-time in the logarithm of the characteristic of the field, even for multivariate polynomials [2; 4; 7; 16; 22]. If we look at the worst case running time, then the best algorithms are at least linear in the characteristic of the field [2].

We now discuss some research areas where L. Lovász' basis reduction algorithm played a crucial role in obtaining important new results. This is to show that the area of *symbolic and algebraic computation* is closely related to a lot of other interesting and active research areas which have not traditionally been involved in computer algebra.

As a first important application of the polynomial-time polynomial factorization algorithm, we want to mention the result by Susan Landau and Gary Miller in [12]. They showed that it can be decided in polynomial-time whether the roots of a polynomial can be expressed by radicals or not. Besides the L^3-algorithm, their algorithm makes use of another recent result which says that the order of a primitive solvable group of degree n is bounded by $c_1 n^{c_2}$, for constant c_1 and c_2 [23].

Testing a (sufficiently precise rational approximation of a) complex number for algebraicity, can be shown to be equivalent with looking for a relatively short vector in a lattice (see [9; 20] or [24]; in the last paper also an improved version of the basis reduction algorithm is described). Therefore, complex numbers can be tested for algebraicity in polynomial-time, which obviously leads to another polynomial-time algorithm for factoring polynomials with rational coefficients. We can also say that the bits of an algebraic number cannot be used to generate random sequences [9].

Also in cryptography progress has been made. A well-known public-key cryptosystem was the Merkle-

Hellman cryptosystem, which was based on the difficulty of solving knapsack problems. First Adi Shamir had broken the basic Merkle-Hellman cryptosystem [25] (his method used the algorithm from [21]). However, the iterated Merkle-Hellman cryptosystem was not affected by Shamir's attack. Recently, several other papers that have a high probability to break the more complicated iterated cryptosystem were published [1; 3; 10]; they are all based on the basis reduction algorithm. As a result, the situation now is that knapsack-based cryptosystems are considered to be unsafe.

Another subject, which is related to the *Riemann Hypothesis*, is the disproval of Mertens' conjecture. Although nobody in fact expected the Mertens' conjecture to be true (this would have implied the Riemann Hypothesis) it was considered to be too difficult to perform the obvious attacks to disprove it. This changed when the power of vector-computers could be combined with the basis reduction algorithm to compute short vectors in lattices (computations performed by A.M. Odlyzko on the Bell Labs Cray 1). Combined with H. Te Riele's high precision computations of the first 2000 non-trivial zeros of the Riemann zeta function this led to a proof that an infinity of counterexamples to Mertens' conjecture exists [26].

Finally, we want to mention the problem of computing shortest vectors in a lattice. No polynomial-time algorithm for this problem is known yet, and it is widely conjectured that no polynomial-time solution exists. (Notice that the polynomial-time basis reduction algorithm can only be guaranteed to find a reasonable approximation of a shortest vector!) If the L_2-norm is replaced by the L_∞-norm, the problem is known to be NP-hard [27].

Clearly, there is a lot of interaction possible between computer algebra and many other branches of mathematics; more than usually is recognized. The future of computer algebra should therefore be to attract as many disciplines as possible, and to interest mathematicians in the problems that are still open in the field. We should try and look at *more* than what nowadays is considered to be *computer* algebra.

References

1. L. Adleman, *On breaking generalized knapsack public key cryptosystems,* Proceedings 15th ACM symposium on theory of computing (1983), 402-412.
2. E.R. Berlekamp, *Factoring polynomials over large finite fields,* Math. Comp. **24** (1970), 713-735.
3. E. Brickell, *Are most low density knapsacks solvable in polynomial time?* Proceedings 14th southeastern conference on combinatorics, graph theory, and computing, 1983.
4. A.L. Chistov, D.Y. Grigoryev, *Polynomial-time factoring of the multivariable polynomials over a global field,* LOMI preprint E-5-82, Leningrad 1982.
5. E. Kaltofen, D.R. Musser, B.D. Saunders, *A generalized class of polynomials that are hard to factor,* Proceedings 1981 ACM symposium on symbolic and algebraic computation, 188-194.
6. E. Kaltofen, *On the complexity of factoring polynomials with integer coefficients.* Ph.D. thesis, Rensselaer Polytechnic Institute, August 1982.
7. J. Von zur Gathen, E. Kaltofen, *Polynomial-time factorization of multivariate polynomials over finite fields,* Proceedings 10th international colloquium on automata, languages and programming, LNCS 154, 250-263.
8. R. Kannan, *Improved algorithms for integer programming and related lattice problems,* Proceedings 15th ACM symposium on theory of computing (1983), 193-206.
9. R. Kannan, A.K. Lenstra, L. Lovász, *Predicting bits of algebraic numbers and factorization of polynomials,* to appear in proceedings 16th ACM symposium on theory of computing (1984).
10. J.C. Lagarias, A.M. Odlyzko, *Solving low-density subset sum problems,* Proceedings 24th IEEE symposium on foundations of computer science (1983), 1-10.
11. S. Landau, *Factoring polynomials over algebraic number fields,* manuscript, 1982.
12. S. Landau, G. Miller, *Solvability by radicals is in polynomial time,* Proceedings 15th ACM symposium on theory of computing (1983), 140-151.
13. A.K. Lenstra, *Lattices and factorization of polynomials over algebraic number fields,* Proceedings Eurocam '82 (1982), LNCS 144, 32-39.
14. A.K. Lenstra, H.W. Lenstra, Jr., L. Lovász, *Factoring polynomials with rational coefficients,* Math. Ann. **261** (1982), 515-534.
15. A.K. Lenstra, *Factoring polynomials over algebraic number fields,* Proceedings Eurocal '83 (1983), LNCS 162, 245-254.
16. A.K. Lenstra, *Factoring multivariate polynomials over finite fields,* report IW 221/83, Mathematisch Centrum, Amsterdam; to appear in the special STOC issue of *Computer and system sciences.*
17. A.K. Lenstra, *Factoring multivariate integral polynomials,* report IW 229/83, Mathematisch Centrum, Amsterdam; Proceedings 10th international colloquium on automata, languages and programming, LNCS 154, 458-465; to appear in the special ICALP issue of *Theoretical computer science.*
18. A.K. Lenstra, *Factoring multivariate integral polynomials, II,* report IW 230/83, Mathematisch Centrum, Amsterdam.
19. A.K. Lenstra, *Factoring multivariate polynomials over algebraic number fields,* report IW 233/83, Mathematisch Centrum, Amsterdam.

20. A.K. Lenstra, *Polynomial factorization by root approximation*, report IW 242/83, Mathematisch Centrum, Amsterdam.

21. H.W. Lenstra, Jr., *Integer programming with a fixed number of variables*, Math. Oper. Res. 8 (1983), 538-548.

22. M.O. Rabin, *Probabilistic algorithms in finite fields*, SIAM J. Comput. 9 (1980), 273-280.

23. Palfy, *A polynomial bound for the orders of primitive solvable groups*. J. of Algebra, July 1982, 127-137.

24. A. Schönhage, *Factorization of univariate integer polynomials by diophantine approximation and by an improved basis reduction algorithm*, to appear in proceedings 11th international colloquium on automata, languages and programming, Antwerp 1984.

25. A. Shamir, *A polynomial time algorithm for breaking the Merkle-Hellman cryptosytem*, Proceedings 23th IEEE symposium on foundations of computer science (1982), 145-152.

26. H. Te Riele, *Mertens' conjecture disproved*, CWI Newsletter 1 (1983), 23-24 (Centrum voor Wiskunde en Informatica, Amsterdam).

27. P. Van Emde Boas, *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*, Rep. Dep. Math. 81-04, University of Amsterdam, April 1981.

28. J. Von zur Gathen, *Hensel and Newton methods in valuation rings*, Math. Comp., to appear.

29. J. Von zur Gathen, *Factoring sparse multivariate polynomials*, Proceedings 24th IEEE symposium on foundations of computer science (1983), 172-179.

(Continued from p. 19)

details the numerics of computational geometry - such as the problems caused by braiding straight lines.

The state of the art: what is it, what should it be?

Today's commercially available software in computer graphics and CAD has not yet taken into account the results of computational geometry. Straightforward algorithms are mostly used whose theoretical efficiency is poor as compared to known results. Perhaps the straightforward algorithms are better in practice than theoretically optimal ones, but such difficult questions have hardly been investigated, as CAD systems development today is so labor intensive that all resources are absorbed by just getting the system to work, and algorithm analysis has so far largely restricted itself to theoretically measurable performance.

We know by analogy with numerical analysis what the next step should be in the maturing process of computational geometry: The development of efficient, portable, robust program libraries for the most basic, frequent geometric subroutine library of CAD, thus exposing theoretical results to a severe practical test. The interaction between computational geometry and computer-aided design promises to be mutually beneficial.