Small Algorithms for Small Systems

James H. Davenport
University of Bath — visiting Symbolic Computation Group
J.H.Davenport@bath.ac.uk

June 13, 2009

Knuth is said to have described Computer Science as "that part of mathematics in which $\log\log n = 3$". In this talk I will consider only some parts of Computer Algebra, and the even more special case when $\log n = 3$, or even less, and where compactness of the algorithm itself, as well as the data structures, is important.

g.c.d. This has been a bugbear of computer algebra for over forty years, and has given rise to many solutions, some of then truly heroic [CGG84, DP85]. Though difficult to *prove*, the subresultant algorithm [Col67] is quite short to *program*, and its intermediate expression swell does not manifest itself on small examples. It may well be worth considering the trial division variant of [Hea79].

Factoring (of univariate polynomials). This has been a challenge for almost as long as the g.c.d. problem, and is still far from being solved, as significant improvements keep on being made [vH02]. Nevertheless, if $\log n = 3$, we can devise a relatively simple algorithm on the following lines.

1. To factor mod p, use Cantor–Zassenhaus [CZ81].
2. Possibly use several primes — see question 1 below.
3. Having decided that there is a viable factorization, we have to lift it p-adically. Again, we note that while an optimal lifting is a very complicated body of code, linear lifting [DST93, p. 168], with imposed leading coefficients [DST93, pp. 174–5], is not.
4. Obviously, any p-adic factor which divides over the integers is a true factor. If this doesn't happen, we have two choices.
   (a) Do appropriate recombinations and trial divisions. The code is not lengthy, but the runnning may well be, since most optimisations ([ABD85] is possibly a counterexample) will substantially lengthen the code.
   (b) Just give up, and declare "I couldn't find any factors, but they may nonetheless exist". In practice, this may well be acceptable on a compact system.

Factoring (of multivariate polynomials). It's not clear to this author that this is worth implementing.

Integration Here the Risch–Norman [NM77] algorithm can be quite short to program, and, while not a full decision procedure, *is* complete on a reasonable range of transcendental integrands [Dav82]. There is a recent extension [Kau08], which looks promising on many cases of algebraic integrands. Here the aim would be to integrate *correctly* many common cases, while *not* guaranteeing that "I can't" is equivalent to "no-one can".

Open Research Questions

Question 1 *How many primes p_i should we factorize modulo in step 2 above before deciding that we have a compatible factorization, and should proceed to Hensel lifting.*

[Mus78] suggests that the answer is 5, though there are heuristic arguments that this should grow as $\log \log d$, where d is the degree of the polynomial to be factored. If d is small, can we get away with less?

References

[ABD85] J.A. Abbott, R.J. Bradford, and J.H. Davenport. A Remark on Factorisation. *SIGSAM Bulletin 2*, 19:31–33, 1985.

[CGG84] B.W. Char, K.O. Geddes, and G.H. Gonnet. GCDHEU: Heuristic Polynomial GCD Algorithm Based on Integer GCD Computation. In J.P. Fitch, editor, *Proceedings EUROSAM 84*, pages 285–296, 1984.

[Col67] G.E. Collins. Subresultants and Reduced Polynomial Remainder Sequences. *J. ACM*, 14:128–142, 1967.

[CZ81] D.G. Cantor and H. Zassenhaus. A New Algorithm for Factoring Polynomials over Finite Fields. *Math. Comp.*, 36:587–592, 1981.

[Dav82] J.H. Davenport. On the Parallel Risch Algorithm (I). In *Proceedings EUROCAM '82 [Springer Lecture Notes in Computer Science 144]*, pages 144–157, 1982.

[DP85] J.H. Davenport and J.A. Padget. HEUGCD: How Elementary Upperbounds Generate Cheaper Data. In *Proceedings EUROCAL 85*, pages 18–28, 1985.

[DST93] J.H. Davenport, Y. Siret, and E. Tournier. Computer Algebra (2nd ed.). *Academic Press*, 1993.

[Hea79] A.C. Hearn. Non-Modular Computation of Polynomial Gcd Using Trial Division. In *Proceedings EUROSAM 79*, pages 227–239, 1979.

[Kau08] M. Kauers. Integration of Algebraic Functions: A Simple Heuristic for Finding the Logarithmic Part. In *Proceedings ISSAC 2008*, pages 133–140, 2008.

[Mus78] D.R. Musser. On the efficiency of a polynomial irreducibility test. *J. ACM*, 25:271–282, 1978.

[NM77] A.C. Norman and P.M.A. Moore. Implementing the New Risch Integration Algorithm. In *Proceedings 4th. Int. Colloquium on Advanced Computing Methods in Theoretical Physics*, pages 99–110, 1977.

[vH02] M. van Hoeij. Factoring polynomials and the knapsack problem. *J. Number Theory*, 95:167–189, 2002.