Scientific Results of the ANC Associate Team

`http://www.loria.fr/~zimmerma/anc.html`, 2008-2010

Integer Factorization. On October 7-9, 2008, a workshop on integer factorization was organized in Nancy, with joint support from the CADO ANR project, and from the ANC associate team project. In particular, Richard Brent and Paul Leopardi from ANU attended this workshop. The slides from the workshop are available from `http://cado.gforge.inria.fr/workshop/`. During that workshop, Pierrick Gaudry defended his habilitation thesis. In January 2010, Alexander Kruppa defended his PhD thesis, whose subject did exactly match the ANC themes [14]. Together with Peter Montgomery, A. Kruppa has designed a new algorithm for stage 2 of the $P-1$ and $P+1$ methods [16], and has implemented it within GMP-ECM [13].

Sage Days 10 Workshop. The Sage Days 10 workshop, on October 10-15, 2008, attracted in Nancy about 80 participants worldwide, among which about 40 stayed for the "coding sprints". During that workshop a collaboration started with the Sage developers about fast modular composition in GF(2)[x]. This is now implemented within Sage (`http://trac.sagemath.org/sage_trac/ticket/4302`). As part of his contribution to open-source software Paul Leopardi developed a test package for random number generators and discovered statistical flaws in the implementation of some well-known tests. These flaws caused good random number generators to appear to fail the tests [15].

ANTS-IX Conference. On July 19-23, 2010, we have organized the ANTS-IX Conference in Nancy (Algorithmic Number Theory Symposium), whose topics largely cover those of ANC, and which attracted about 150 participants worldwide (`ants9.org`). The Nancy team was in charge of the local organization of ANTS-IX, and ANC partly supported the travel of Shi Bai who attended the conference.

Multiplication in GF(2)[x]**.** During the visit of R. Brent to Nancy in October 2008, a new version of the `gf2x` package was released. This package is an efficient implementation of arithmetic of univariate polynomials over GF(2). The algorithms implemented in `gf2x` are described in the paper presented at the ANTS-VIII conference [5], and go from the base case (one word of 64 bits representing a polynomial of degree < 64) to algorithms using the Fast Fourier Transform for huge degrees. On an Intel Core2 CPU, `gf2x` is about 1.8 times faster than NTL 5.4.1 for degree < 64, and about 5.1 times faster than NTL

for degree < 1048576. Since version 5.5, NTL includes support for `gf2x`, and thus `gf2x` is available from within the SAGE computer algebra system. There is also a wrapper of `gf2x` in the MATHEMAGIX computer algebra system by van der Hoeven, Lecerf *et al.* The `gf2x` package was used by R. Brent and P. Zimmermann to find primitive trinomials of huge degree (see below).

In late 2010, we have extended `gf2x` to use the PCLMULQDQ assembly instruction available on the new Intel processors (Westmere for example). This instruction performs a 64×64 bit multiplication without carries in a latency of 13 cycles, with throughput of 8 cycles, against about 50 cycles latency and throughput for our software implementation [5]. This yields a gain of about a factor of two in polynomial factorization (see below).

Factorization and Irreducibility in GF(2)[x]**.** During an earlier visit of P. Zimmermann at ANU in 2007, R. Brent and P. Zimmermann have found a new "multi-level" algorithm to factor univariate polynomials over GF(2)[x]. The main idea is to replace multiplications by squarings, which are much faster over GF(2)[x]. If the reductions are cheap, for example with trinomials, the speedup is about $\sqrt{M/S}$, where M is the time to multiply two polynomials of degree n, and S is the time to square a polynomial of degree n. This algorithm has been published in Contemporary Mathematics [8] and implemented in the `gf2x` package. During his visit in Nancy in 2008, Joerg Arndt designed a new algorithm to check irreducibility of polynomials over a finite field, which avoids GCDs [2].

Primitive Trinomials. R. Brent and P. Zimmermann have continued their search for primitive trinomials of huge degree, catching up the progress made by the GIMPS (Great Internet Mersenne Prime Search) project. Indeed, if $2^r - 1$ is prime, then the trinomial $x^r + x^s + 1$ over GF(2) is primitive iff it is irreducible. Thus it suffices to check irreducibility, which is done using the algorithm from [8] implemented within the `gf2x` package. For $r = 32582657$, three primitive trinomials have been found; for $r = 43112609$, four primitive trinomials have been found; and for $r = 42643801$, five primitive trinomials have been found. Part of those results has been published in Mathematics of Computation [9]. An invited paper on this search was written for the AMS Notices [12].

Book "Modern Computer Arithmetic". During their visits to Nancy or ANU (R. Brent in October 2008 and April-May 2009, P. Zimmermann in June 2008, July 2009 and May 2010), R. Brent and P. Zimmermann worked on a book "Modern Computer Arithmetic". Several electronic versions were published on the web (the book will remain freely available on the web once the paper version is published). The current version (0.5.7 from September 2010) will be published by Cambridge University Press [10].

Book "Matters Computational". The book of Jörg Arndt "Matters Computational" (previously "Algorithms for Programmers") has also reached its

final stage and should soon be published [1]. J. Arndt also completed his PhD thesis during the period [3].

Fast Jacobi Symbol Computation. Following a question from Steven Galbraith, R. Brent and P. Zimmermann worked out a 2-adic (least significant bit first) algorithm with subquadratic complexity. Note that all the software tools (Magma, Pari/GP, GMP) currently implement a quadratic algorithm to compute the Jacobi symbol. The new algorithm was implemented in GMP and published in the proceedings of the ANTS-IX conference [11].

Polynomial Selection for the Number Field Sieve. During the visit of Shi Bai in Nancy in July 2010, we worked on understanding and implementing the new polynomial selection algorithm presented by Thorsten Kleinjung at the joint CADO-ANC workshop in 2008. This new algorithm is now implemented in CADO-NFS. First results seem to indicate that this new algorithm generates good polynomials much quicker than the previous algorithm from T. Kleinjung (Mathematics of Computation, 2006). During a previous visit in October-November 2009, Shi Bai had also worked on extending CADO-NFS to the discrete logarithm problem (DLP), following previous work with R. Brent on the subject [4].

Hadamard Maximal Determinant Problem. This project was done in collaboration with Will Orrick (Indiana University). During the visit of R. Brent and J.-A. Osborn in Nancy in April-May 2009, a first version in C of the Gram searching program was written, together with a first prototype in Maple of the Gram decomposition program. During the visit of P. Zimmermann at ANU in July 2009, those programs were improved. We decided to organize a workshop on this problem in May 2010 at ANU, with partial support from the associate team. Our results are not yet published, but the main results so far are the following [6, 7]:

- for size $n = 19$, we have proved that the maximal determinant is $833 \cdot 4^6 \cdot 2^{18}$, which was the best value previously known. The search program, run in parallel on 50 machines, found 9 candidate matrices in 900 cpu hours. From those 9 candidate matrices, only two decompose (in three inequivalent ways); the latter result was obtained with our decomposition program. This result closes the question for $n = 19$.
- similarly for $n = 37$, we have proved that the maximal determinant is $72 \cdot 9^{17} \cdot 2^{36}$. Our search program found 807 matrices with determinant larger or equal to that bound (after 78 hours of computation), from which only one matrix was shown to be decomposable. This closes the search for $n = 37$ too.

Following an idea of Will Orrick at the May 2010 workshop, we were able to generalize the Gram searching program to the case n even. A search was started for $n = 22$, but no significant result found so far.

References

[1] Arndt, J. Matters computational. `http://www.jjj.de/fxt/#fxtbook`. To appear, 966 pages.

[2] Arndt, J. Testing polynomial irreducibility without GCDs. Research Report RR-6542, Institut National de Recherche en Informatique et en Automatique, 2008. `http://hal.inria.fr/inria-00281614`.

[3] Arndt, J. *Generating Random Permutations.* Phd thesis, Australian National University, 2010. `http://wwwmaths.anu.edu.au/~brent/students.html#arndt`.

[4] Bai, S., and Brent, R. P. On the efficiency of Pollard's rho method for discrete logarithms. In *Proceedings of The Australasian Theory Symposium (CATS2008)* (Wollongong, 2008), J. Harland and P. Manyem, Eds., vol. 77 of *Conferences in Research and Practice in Information Technology*, Australian Computer Society, pp. 125–131.

[5] Brent, R., Gaudry, P., Thomé, E., and Zimmermann, P. Faster multiplication in GF(2)[x]. In *Proceedings of the 8th International Symposium on Algorithmic Number Theory (ANTS VIII)* (2008), A. J. van der Poorten and A. Stein, Eds., vol. 5011 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 153–166.

[6] Brent, R. P., Orrick, W. H., Osborn, J. H., and Zimmermann, P. Maximal determinants and saturated D-optimal designs of orders 19 and 37. In preparation, abstract available on `http://wwwmaths.anu.edu.au/~osborn/publications/pubsall.html`.

[7] Brent, R. P., and Osborn, J. H. Minors of maximal determinant matrices. In preparation.

[8] Brent, R. P., and Zimmermann, P. A multi-level blocking distinct degree factorization algorithm. *Contemporary Mathematics 461* (2008), 47–58.

[9] Brent, R. P., and Zimmermann, P. Ten new primitive binary trinomials. *Mathematics of Computation 78*, 266 (2009), 1197–1199.

[10] Brent, R. P., and Zimmermann, P. *Modern Computer Arithmetic.* Cambridge University Press, 2010. Electronic version freely available at `http://www.loria.fr/~zimmerma/mca/pub226.html`.

[11] Brent, R. P., and Zimmermann, P. An $O(M(n)\log n)$ algorithm for the Jacobi symbol. In *Proceedings of the 9th Algorithmic Number Theory Symposium (ANTS-IX)* (Nancy, France, July 19-23, 2010, 2010), G. Hanrot, F. Morain, and E. Thomé, Eds., vol. 6197 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 83–95.

[12] Brent, R. P., and Zimmermann, P. The great trinomial hunt. *Notices of the AMS* (to appear, see `arXiv:1005.1967`).

[13] Gaudry, P., Gladman, B., Fougeron, J., Fousse, L., Kruppa, A., Newman, D., Papadopoulos, J., and Zimmermann, P. *GMP-ECM*, 6.3 ed., 2010. `http://gforge.inria.fr/projects/ecm`.

[14] Kruppa, A. *Améliorations de la multiplication et de la factorisation d'entier*. Phd thesis, University Henri Poincaré Nancy 1, 2010.

[15] Leopardi, P. Testing the tests: using random number generators to improve empirical tests. In *Monte Carlo and Quasi-Monte Carlo Methods 2008* (2009), P. L'Ecuyer and A. B. Owen, Eds., Springer-Verlag, pp. 501–512.

[16] Montgomery, P. L., and Kruppa, A. Improved stage 2 to $P \pm 1$ factoring algorithms. In *Proceedings of the 8th Algorithmic Number Theory Symposium (ANTS VIII)* (2008), A. J. van der Poorten and A. Stein, Eds., vol. 5011 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 180–195.