Homotopy methods for equations over finite fields

Alan G.B. Lauder *

Computing Laboratory, Oxford University
Oxford OX1 3QD, U.K.
alan.lauder@comlab.ox.ac.uk

Abstract. This paper describes an application of some ideas from homotopy theory to the problem of computing the number of solutions to a multivariate polynomial equation over a finite field. The benefit of the homotopy approach over more direct methods is that the running-time is far less dependent on the number of variables. The method was introduced by the author in another paper, where specific complexity estimates were obtained for certain special cases. Some consequences of these estimates are stated in the present paper.

1 Introduction

A basic problem in computational mathematics is to solve a system of polynomial equations with coefficients in a field. By "solve" one typically means finding a solution, or perhaps all of the solutions if there are known to be only finitely many. When the field admits a non-trivial norm, such as the field of real numbers, then powerful analytic methods can be brought to bear. For example, Newton's method for locating zeros of a real polynomial, and its various generalisations [1]. However, when no such norm exists, one is forced to fall back on algebraic techniques. Consider, for example, a single polynomial equation in one variable over a finite prime field. Using a deterministic algorithm, Berlekamp's root-counting algorithm, one can compute in an efficient manner the number of solutions of the equation in the prime field, see [3, Chapter 14]. By an ingenious application of randomisation, one can actually find all the solutions efficiently. However, finding a solution efficiently using a deterministic algorithm seems a much more difficult, and as yet unsolved, problem. By "efficient" here, and elsewhere, I mean in polynomial-time in the size of the input.

The purpose of the present short paper is to describe a method I have recently been exploring for the related problem of counting, rather than finding, solutions to equations; specifically, for computing the number of rational solutions to a single multivariate polynomial equation over a finite field [6]. This technique involves deforming one polynomial into another. Such deformations

* The author is supported by the EPSRC (Grant GR/N35366/01) and St John's College, Oxford.

only make any intuitive sense over a field, such as the real numbers, which admits a non-trivial norm. However, by the application of some rather deep theory, based upon work of Dwork from the 1960s [2], these deformations also become useful over finite fields. My work is inspired by the paper of Dwork in which he uses deformations to prove the "functional equation of the zeta function of a smooth projective hypersurface". From the point of view of algorithms, the interesting aspect of Dwork's deformation theory is that it can lead to remarkable improvements in computational complexity. The algorithms I will describe bear a passing resemblance to the "homotopy methods" for locating zeros of a system of complex multivariate polynomial equations [1, Section 4.2], whence the title of this paper. In these methods, one begins with an approximate zero of a perturbed system, and then one gradually homes in on a zero of the original system by following a path in some appropriate space. In Dwork's theory, from knowledge of the number of solutions of some perturbed polynomial equation, one can recover the number of solutions of the original polynomial equation by studying the path taken from the original to the perturbed polynomial.

A considerable body of work has appear in the last few years on the problem of counting solutions to equations over finite fields, see for example the references in [7]. To my knowledge, the paper [6] describes the first explicit algorithmic application of a "homotopy method" to this problem. I would be very interested in learning of any other applications of "homotopy methods" to algorithmic, or theoretical, problems on equations over finite fields.

2 The Method

Let $\mathbb{F}_q$ denote the finite field with q elements, where q is a power of a prime p. Let $f \in \mathbb{F}_q[X_1, \ldots, X_n]$ be a homogeneous polynomial of degree d in the variables $X_1, \ldots, X_n$. Assume that p does not divide d, so that Dwork's theory can be applied below. Our aim will be to compute $N(f)$, the number of rational projective solutions to the equation $f = 0$. Specifically, the number $N(f)$ is defined via the equation

$$(q-1)N(f) + 1 = \#\{(x_1, \ldots, x_n) \in \mathbb{F}_q^n \,|\, f(x_1, \ldots, x_n) = 0\}.$$

To compute $N(f)$ in a naive fashion, by substituting in f all rational projective points, would require $q^{n-1} + \ldots + q + 1$ evaluations of f. This is certainly $\Omega(q^{n-1})$ bit operations (ignoring the dependence on d which is not such a concern when just counting rational solutions). However, the dense input size to this problem is about $\binom{n+d}{n}\log(q)$ bits, i.e., $\Theta(\log(q))$ bits. One needs to reduce the dependence on $\log(q)$ from $\Omega(q^{n-1})$ to $\log(q)^{O(1)}$ to get a more practical algorithm. Some progress towards this can be made using p-adic cohomology, as I will now describe.

Under the assumption that the zero set of f is "smooth and in general position" (see [4, Page 75]), there is a cohomological formula

$$N(f) = \frac{q^{n-1} - 1}{q - 1} + (-1)^n \frac{\mathrm{Trace}(\alpha)}{q}. \tag{1}$$

Here α is a matrix whose entries lie in the p-adic field obtained by lifting $\mathbb{F}_q$ to characteristic zero and adjoining a primitive pth root of unity. (The primitive pth root is required since Dwork reduces everything to additive character sums, and these are only defined in fields with pth roots.) The matrix α has dimension

$$\frac{1}{d}\{(d-1)^n + (-1)^n(d-1)\} \in \mathbb{Z}.$$

It is the matrix of Frobenius on the primitive middle-dimensional p-adic cohomology space, which we will just call the *Frobenius matrix.* One wishes to construct this Frobenius matrix, or at least find its trace. Dwork's theory is constructive, and using a fairly direct approach the Frobenius matrix can be explicitly computed. This direct approach was first used by Kedlaya [5], and is extremely good for curves where $n = 3$. However, the computational complexity of this direct approach for an equation in n variables appears to be $(pd\log(q))^{\Theta(n)}$ bit operations, at least using the most straightforward generalisation. This is just the same complexity as in the theorem of the author and Wan [7, Theorem 1], but using Kedlaya's method the exponent can be roughly halved. The reason for this complexity is that the computations required involve n-variate polynomials of total degree around $pd\log(q)$, and such polynomials have roughly $(pd\log(q))^n$ terms. (Note that the factor p in the complexity is undesirable, but it is difficult to see how it could be replaced by $\log(p)$ using only p-adic cohomology. Grothendieck's l-adic cohomology theory might achieve this if it could be made constructive in general. This can be done for curves, using methods going back to Weil, but even in this case it only gives good algorithms at present for small genus.) The aim of the homotopy method is to remove the dependence in n from the exponent of $p\log(q)$. Specifically, I believe a complexity of $c_n(pd^n\log(q))^{O(1)}$ bit operations, where c_n depends only on n, should be possible. I have worked out this approach for a special family of equations — the precise results obtained are described in the next section. In the remainder of this section I will sketch the homotopy method and explain why it should be useful.

We write our polynomial f in the form

$$f = \sum_{i=1}^{n} a_i X_i^d + h(X_1, \ldots, X_n)$$

where h is a homogeneous polynomial of degree d with no diagonal terms. Generically $a_1 \ldots a_n \neq 0$, and we shall assume that this is the case. We wish to "deform" f to a diagonal form $\sum_{i=1}^{n} a_i X_i^d$ by making the remainder term h tend to zero. To this end, we introduce an extra variable Y which controls the deformation. Define

$$f_Y = \sum_{i=1}^{n} a_i X_i^d + Y h(X_1, \ldots, X_n) \in \mathbb{F}_q[Y][X_1, \ldots, X_n].$$

Setting $Y = 1$ gives our original polynomial $f = f_1$. Setting $Y = 0$ gives the diagonal form f_0. Intuitively, as Y moves from 1 to 0 our original polynomial is deformed into a diagonal form. Let $N(f_y)$ denote the number of rational projective zeros of the specialised polynomial f_y, where we take $Y = y \in \mathbb{F}_q$. For all but finitely many y in the algebraic closure of $\mathbb{F}_q$, the zero set of f_y will be "smooth and in general position". We will say that such a y defines a *smooth fibre*. If y defines a smooth fibre, a Frobenius matrix α_y is defined, and we have the formula

$$N(f_y) = \frac{q^{n-1}-1}{q-1} + (-1)^n \frac{\mathrm{Trace}(\alpha_y)}{q}.$$

A *generic* Frobenius matrix $\alpha(Y)$ for the polynomial f_Y is also defined. Its entries are p-adic analytic functions in the variable Y which will converge on the Teichmüller lifting of any point $y \in \mathbb{F}_q$ which defines a smooth fibre. For those y at which $\alpha(Y)$ is defined, the generic Frobenius matrix $\alpha(Y)$ converges to the specialised Frobenius matrix α_y. Dwork's deformation theory yields the following factorisation

$$\alpha(Y) = C(Y^q)^{-1}\alpha(0)C(Y). \tag{2}$$

Here $C(Y)$ is a matrix of p-adic analytic functions which need not converge on the Teichmüller lifting of any non-zero point in $\mathbb{F}_q$. One may compute $\alpha(0)$ easily, as it is the Frobenius matrix α_0 of a diagonal form and has a nice Kronecker product decomposition. The matrix $C(Y)$ is the solution matrix of a system of linear differential equations: $dC(Y)/dY = C(Y)B(Y)$ and $C(0) = I$. Here the matrix $B(Y)$ contains entries which are rational functions in Y over the integers, and it can be computed using a method due to Dwork. (The matrix $B(Y)$ for the elliptic curve case is calculated in Dwork's original paper [2, Section 8].) All the theory is now in place to describe the deformation approach for computing the number of projective zeros of the original polynomial f: First, compute the rational matrix $B(Y)$ using Dwork's method. Second, compute an expansion of $C(Y)$ around the origin by solving the differential system numerically. Third, compute $\alpha(0) = \alpha_0$ directly from its Kronecker product decomposition. Fourth, recover an expansion of $\alpha(Y)$ around the origin from Equation (2). Fifth, use this expansion to compute $\alpha = \alpha_1 = \alpha(1)$. Finally, the trace of the matrix α yields the number of projective solutions, as in Equation (1).

One subtlety in this approach should be pointed out: because some y in the algebraic closure of $\mathbb{F}_q$ do not define smooth fibres, the expansion of $\alpha(Y)$ about the origin will not converge on the Teichmüller lifting of non-zero points of $\mathbb{F}_q$. Thus for the final step one needs some method of "continuing" $\alpha(Y)$ to the Teichmüller liftings of the points $y \in \mathbb{F}_q$ which define smooth fibres. In the next section I describe an easier, though non-generic, situation in which the singular fibres do not cause such a complication. I have implemented the algorithm in this simpler situation for some small examples with the help of Frederik Vercauteren.

The reason that the complexity is improved using the above method is that the deformation is always one-dimensional regardless of the number of variables n in the original problem. In practice, this means that one computes with univariate, rather than n-variate, polynomials. Even in the case of curves, in certain special cases the deformation method requires less space than that of Kedlaya although the time complexity is the same.

Note that throughout this section I have glossed over one important point: in all algorithms which exploit p-adic cohomology in some way, it is essential to first compute the "semi-linear Frobenius matrix" rather than the Frobenius matrix itself. This turns the "q" into a "$p \log_p(q)$" in the complexity estimates, but it does make the formulae a bit more involved.

3 Results

I will finish now by stating the results that have been obtained in my paper [6] based upon these ideas. They pertain to certain special equations, namely Artin-Schreier equations with a diagonal leading form. Let $f \in \mathbb{F}_q[X_1, \ldots, X_n]$ be of degree d where p does not divide d (f is not necessarily homogeneous). We will say that f has a diagonal leading form if it can be written as $f = \sum_{i=1}^n a_i X_i^d + h(X_1, \ldots, X_n)$ where $a_1 \ldots a_n \neq 0$ and h has degree strictly less than d. Let $N(f)$ be the number of affine solutions in $\mathbb{F}_q^{n+1}$ to the equation $Z^p - Z = f(X_1, \ldots, X_n)$.

Theorem 1. *There exists an explicit deterministic algorithm with the following input, output and complexity. The input is a polynomial $f \in \mathbb{F}_q[X_1, \ldots, X_n]$ with a diagonal leading form of degree d not divisible by the characteristic p, where $p > 2$. The output is the number $N(f)$ of rational solutions to the affine equation $Z^p - Z = f$. The running time is*

$$\tilde{\mathcal{O}}(c_n d^{\min(4n+1, 3n+3)} \log(q)^3 p^2)$$

bit operations, where c_n depends only on n.

Here the soft-Oh $\tilde{\mathcal{O}}$ notation suppresses logarithmic factors in the parameters d^n, $\log(q)$ and p. The restriction to diagonal leading form is useful as it allows us to avoid the difficulty of "crossing" over singular fibres in the family, i.e., in the fifth step the expansion of $\alpha(Y)$ will converge at the correct points in this case.

A curious corollary of the above is the following result.

Corollary 1. *Let $f \in \mathbb{Z}[X_1, \ldots, X_n]$ have the form*

$$f = \sum_{i=1}^n a_i X_i^d + h(X_1, \ldots, X_n)$$

where $a_1 \ldots a_n \neq 0$ and h has total degree less than d. There exists an explicit deterministic algorithm which takes as input a prime p and outputs the number of solutions to the equation $f = 0 \bmod p$, and runs in $\tilde{\mathcal{O}}(p^2)$ bit operations.

Here the algorithm itself depends upon f, and the hidden constant also depends upon f. The naive bound here would be $\tilde{\mathcal{O}}(p^{n-1})$ using Berlekamp's root-counting algorithm, as alluded to in the introduction.

Acknowledgements

This paper was written to accompany the author's invited talk at AAECC 15, Toulouse, 2003. He wishes to thank the organisers of the conference, most especially Professor Poli, and also Professors Richard Brent and Shuhong Gao for giving helpful comments on the paper.

References

1. J-P. Dedieu, Newton's method and some complexity aspects of the zero-finding problem, in "Foundations of Computational Mathematics", (R.A. DeVore, A. Iserles, E. Suli), LMS Lecture Note Series 284, Cambridge University Press, 2001, 45-67.
2. B. Dwork, On the zeta function of a hypersurface II, Ann. Math. (2) 80, (1964), 227-299.
3. J. von zur Gathen and J. Gerhard, Modern Computer Algebra, Cambridge University Press, 1999.
4. N.M. Katz, On the differential equations satisfied by period matrices, Pub. Math. IHES 35, (1968), 71-106.
5. K. Kedlaya, Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology, Journal of the Ramanujan Mathematical Society, 16 (2001), 323-338.
6. A.G.B. Lauder, Deformation theory and the computation of zeta functions, submitted. Preprint available at:
http://web.comlab.ox.ac.uk/oucl/work/alan.lauder/
7. A.G.B. Lauder and D. Wan, Counting points on varieties over finite fields of small characteristic, to appear in Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography (Mathematical Sciences Research Institute Publications), J.P. Buhler and P. Stevenhagen (eds), Cambridge University Press.