Improved Divisor Arithmetic on Generic Hyperelliptic Curves

Sebastian Lindner, Laurent Imbert, Michael Jacobson Jr.

Improved Divisor Arithmetic on Generic Hyperelliptic Curves

Sebastian Lindner[1] Laurent Imbert[2] Michael J. Jacobson Jr.[1]

(1) University of Calgary, Calgary, Canada

(2) CNRS, LIRMM, Université de Montpellier, Montpellier, France

1 Introduction

The divisor class group of a hyperelliptic curve defined over a finite field is a finite abelian group at the center of a number of important open questions in algebraic geometry, number theory and cryptography. Many of these problems lend themselves to numerical investigation, and as emphasized by Sutherland [14, 13], fast arithmetic in the divisor class group is crucial for their efficiency. Besides, implementations of these fundamental operations are at the core of the algebraic geometry packages of widely-used computer algebra systems such as Magma and Sage.

Divisor class group arithmetic differs on *ramified* (imaginary) and *split* (real) models of curves. Many efficient algorithms have been proposed for the ramified setting, notably due to its extensive use in cryptographic applications. The more complicated split scenario has received less attention from the research community. Yet, split models exist for a much larger class of hyperelliptic curves than ramified models, so exhaustive computations as in [13] conduct the bulk of their work on split models by necessity. Efficient arithmetic is therefore equally important in both models.

In this note, we present the first results in our overall program aiming to improve the state-of-the-art for general-purpose hyperelliptic curve arithmetic, where the curve may have any genus, be defined over any field, and is given in standard Weierstrass form as opposed to a special non-generic form. Our first contribution is an adaptation of Shanks' NUCOMP algorithm [12] for divisor class group arithmetic on split model hyperelliptic curves of arbitrary genus [9]. Our algorithm works for any split model curve given in general Weierstrass form defined over any field. Our Magma implementations for both ramified and split models over prime finite fields are the method of choice for genus greater than 7. In addition, our split model implementation performs better than Cantor's approach and closes the previously-observed performance gap with the ramified model.

Our second contribution is a series of practical improvements to speed-up addition and doubling operations for generic hyperelliptic curves of genus 2 in Weierstrass form defined over any field. Our algorithms, as is typically the case for small genus, are presented as *explicit formulas*. In general, these very efficient algorithms are typically highly specialized for certain families of curves with efficient computational features. For number theoretic applications, this is usually too restrictive, as curves are generic and cannot be chosen with computationally efficient properties. These specialized algorithms typically also only consider the most common case of input divisors because, under the assumption that the arithmetic is computed over large finite fields, the uncommon cases almost never occur and can be computed using elementary methods (such as Cantor's algorithm) when necessary. Again, this is not ideal for number theoretic applications, as the finite field sizes are relatively much smaller. Our contribution is thus a complete suite of explicit formulas for genus 2 hyperelliptic curve arithmetic that covers all possible cases for both ramified and split models, and requires fewer field operations than all other such formulas given in the literature.

2 Divisor Class Group of Hyperelliptic Curves

A *hyperelliptic curve* C of genus $g > 1$ over a field k is a smooth projective curve that admits an affine equation of the form $y^2 + h(x)y = f(x)$, where $f, h \in k[x]$, and either $\deg f = 2g + 1$, $\deg h \leq g$ (ramified model) or $\deg f \leq 2g + 2$, $\deg h \leq g + 1$ (split model). In both cases, $C(k)$ denotes the k-rational points on C. A *divisor* is a formal sum $D = \sum n_P P$ of points $P \in C(\overline{k})$, with only finitely many $n_P \neq 0$. The *degree* of D is $\deg D = \sum n_P$. The set of all degree zero divisors is denoted $\mathrm{Div}_k^0(C)$. The divisor class group or Picard group of $C(k)$ is denoted $\mathrm{Pic}_k^0(C) = \mathrm{Div}_k^0(C)/\mathrm{Princ}_k^0(C)$, where $\mathrm{Princ}_k^0(C) = \{\sum_P \mathrm{ord}_P(\alpha)P$ for some function $\alpha \in k(C)^*\}$ is the set of *principal divisors* on C. The elements in $\mathrm{Pic}_k^0(C)$ are represented using the Mumford representation $D = [u, v]$, where $u, v \in k[x]$, $\deg v < \deg u \leq g$, u monic, and $u|(v^2 + vh - f)$. The group law in $\mathrm{Pic}_k^0(C)$ can be computed generically using Cantor's algorithm [1], and is expressed in terms of polynomial arithmetic.

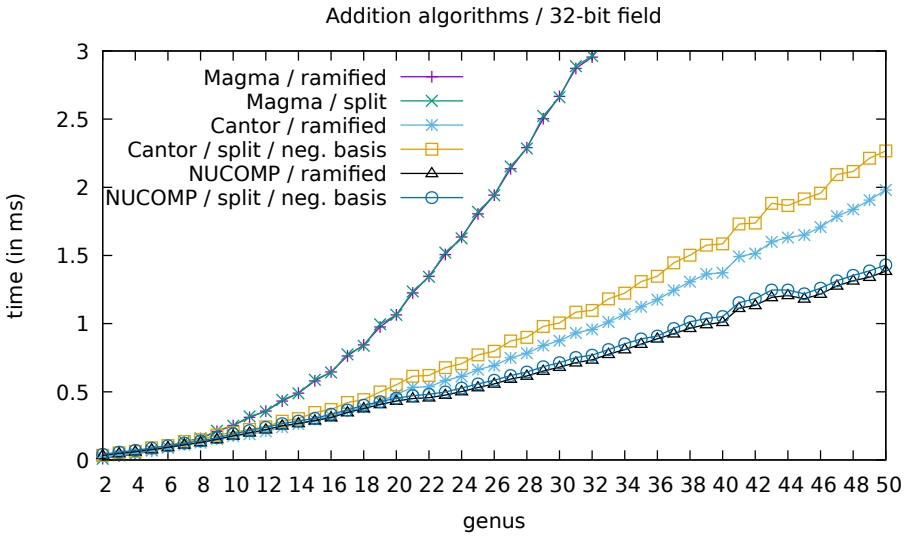
In ramified models, every equivalence class of $\mathrm{Pic}_k^0(C)$ contains a unique, so-called *reduced* representative for which the degrees of u and v are small, bounded by the genus. Unique representatives can also be obtained in split models, but the methods are more complicated, requiring the use of so-called balanced representatives [11] in which an explicit multiple of the two infinite points is tracked and included in the representation to force uniqueness, requiring in some cases extra adjustment steps after reduction.

3 Generic improvements for split models

Various improvements to Cantor's algorithm have been proposed for ramified models, most notably an adaptation of Shank's NUCOMP algorithm [12] for composing binary quadratic forms [7]. The main advantage of NUCOMP is that the sizes of the intermediate operands are reduced via a series of preliminary reduction steps, resulting in better performance in most cases. Improvements to NUCOMP have been proposed, most recently the work of [5], where best practices for computing Cantor's algorithm and NUCOMP are empirically investigated. NUCOMP has also been proposed for arithmetic in the so-called infrastructure of a split model curve [6]. Although Galbraith et. al. [4, 11] have shown that arithmetic in the Jacobian using balanced divisors is more efficient than the closely-related infrastructure, NUCOMP had yet to be applied explicitly to the former.

In [9], we present for the first time an adaptation of NUCOMP designed explicitly for divisor class group arithmetic on split model hyperelliptic curves in the balanced divisor framework. We incorporate all best practices from previous works in the ramified model setting and propose two balanced setting-specific improvements over [6]: we introduce a novel normalization of divisors that eliminates the extra adjustment step required in [6] for curves of odd genus; and we use properties of NUCOMP to save operations in adjustment steps in some cases.

The following graph, taken from [9], illustrates the performance of Magma implementations of our algorithms. We plot the time taken to compute a sequence of divisors additions using $D_{i+1} = D_i + D_{i-1}$, starting from two random divisors using our optimized NUCOMP for ramified models, and our novel version of NUCOMP for split models, as well as Magma's built-in arithmetic, and our optimized implementation of Cantor's algorithm (as described in [9]) for both ramified and split models. We collected timings over a variety of prime finite fields of different sizes, but only show data for 32 bits, as this is fairly representative for all field sizes.



Our data shows that NUCOMP is the method of choice for all but the smallest genera, outperforming our implementation of Cantor's algorithm for genus as low as 5, and outperforming Magma's built-in implementation for $g \geq 7$, with the gaps increasing with the genus. We also observe that, when using balanced NUCOMP, the difference between the best algorithms for split and ramified model curves is negligible for all genus.

4 Explicit formulas for genus 2

Explicit formulas describe divisor arithmetic by an optimized sequence of field operations instead of polynomial arithmetic, where unnecessary operations are eliminated. We have developed novel explicit formulas, to be presented in a forthcoming paper, for addition and doubling in genus 2 ramified and split model hyperelliptic curves given in Weierstrass form defined over any field. Our main contributions are:

- a complete set of operations in the sense that explicit formulas are provided for all possible input and output degrees, requiring only one inversion per operation in all cases;
- a novel combination of techniques due to Lange [8] and Costello and Lauter [2], resulting in formulas that require fewer field operations than the state-of-the-art in all cases, most notably, greatly reducing the number of additions in most cases;
- the use, for split model curves, of our balanced NUCOMP algorithm to cases where extra adjustment steps are required.

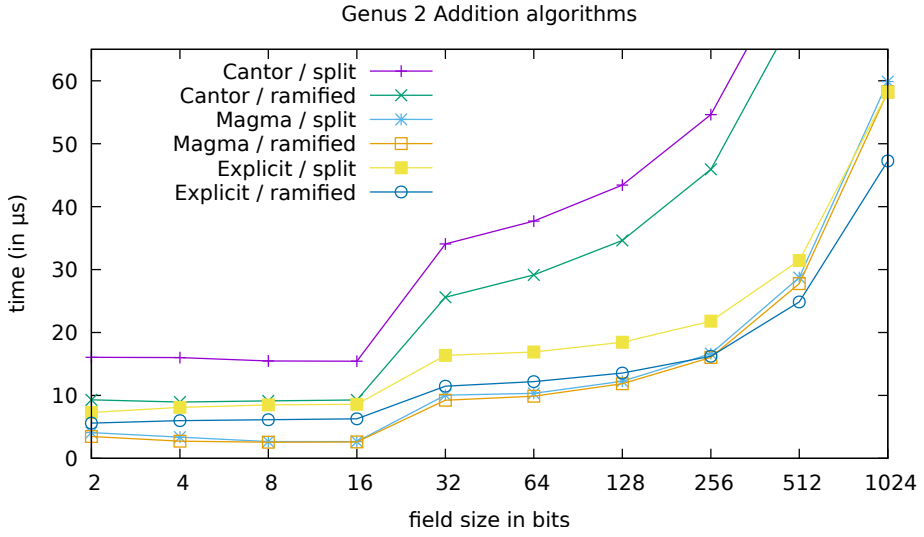
Comparisons of our explicit formulas to the previous best over both ramified and split model curves are presented in the following table. We present operation counts for doubling degree 2 and degree 1 divisors (2DB and 1DB), adding degree 2 divisors, degree 1 with 2, and degree 1 with 1 (2AD, 12AD, 1AD) for ramified models. For split models, we present all of the operations over ramified models except degree 1 doubling, as that operation always requires either a "down" or an "up" adjustment step; counts for those operations are listed (1DDW, 1DUP). We also include counts for all other possible cases, specifically adjustments steps of degree 2 and degree 1 divisors (2DW, 1DW, 2UP, 1UP), degree 1 with 1 addition with up or down adjustments (1ADW,1AUP) and degree 1 with 2 addition with up adjustment (12AUP.) Note that the operations from [3] denoted with (2I) are combinations of two operations, and thus require two inversions. We count field multiplications and squarings (denoted by M) and additions (denoted by A).

We compare the cost of our formulas to [3] for split models, and to [8] and [2] for ramified models. For brevity we only give a sample for curves defined over fields of characteristic $\neq 2$; complete details along with our Magma implementation and testing framework, can be found in [10].

Split Model (M A)																
	2DB		2AD		12AD		1AD		2DW		1DW		2UP		1UP	
[3]	32	46	28	37	19	29	3	5	6	15	6	12	6	15	6	10
This	32	39	28	36	16	22	3	5	6	12	6	9	6	12	6	7
	1DDW		1DUP		1ADW		1AUP		12AUP							
[3](2I)	15	39	15	39	9	20	9	20	25	44						
This	16	23	14	19	12	16	14	17	17	26						

Ramified Model (M A)										
	2DB		1DB		2AD		12AD		1AD	
[8]	27	36	-	-	25	32	11	18	-	-
[2]	26	41	-	-	22	32	-	-	-	-
This	26	25	5	9	23	23	10	15	3	4

Average timings for one addition operation computed within a Fibonacci-like addition sequence with random divisors, ranging over several field sizes, are presented in the following figure. We compare our Magma implementation of genus 2 explicit formulas to the fastest generic algorithms from Section 3 and Magma's corresponding built-in operations. Our data shows that our explicit formulas are faster than our implementation of generic divisor arithmetic, but that it only outperforms Magma's built-in arithmetic when the field is sufficiently large. We suspect, again, that this is due to the fact that our implementation does not have direct access to Magma's internals and is forced to use the generic interface for field arithmetic.



5 Conclusions

Our Magma implementations are state-of-the-art for genus greater than 7, and we are continuing to work on improvements for smaller genera. We continue to optimize our implementation for genus 2, including porting to Sage (where we can use internal primitives) and C/C++. Preliminary results on developing a complete suite of explicit formulas for genus 3 show that NUCOMP yields significant improvements over prior work. Finally, we are working on methods to automatically generate near-optimal explicit formulas for any genus, in order to provide further improvements for $g > 3$ where developing formulas by hand is far too tedious. We hope that our results will yield algorithms and implementations that are the fastest available for all $g \geq 2$.

References

[1] D. Cantor. Computing in the jacobian of a hyperelliptic curve. *Mathematics of Computation*, 48(177):95–101, 1987.

[2] C. Costello and K. Lauter. Group law computations on jacobians of hyperelliptic curves. In Miri and S. Vaudenay, editors, *Selected areas in Cryptography*, volume 7118 of Lecture Notes in Computer Science, pages 92–117. Springer, 2011.

[3] S. Erickson, M. J. Jacobson, Jr., and A. Stein. Explicit formulas for real hyperelliptic curves of genus 2 in affine representation. *Advances in Mathematics of Communication*, 5(4):623–666, 2011.

[4] S. D. Galbraith, M. Harrison, and D. J. Mireles-Morales. Efficient hyperelliptic arithmetic using balanced representation for divisors. In *Algorithmic Number Theory Symposium, ANTS 2008*, volume 5011 of *Lecture Notes in Computer Science*, pages 342–356. Springer, 2008.

[5] L. Imbert and M. J. Jacobson, Jr. Empirical optimization of divisor arithmetic on hyperelliptic curves over $\mathbb{F}_{2^m}$. *Advances in Mathematics of Communication*, 7(4):485–502, 2013.

[6] M. J. Jacobson, Jr., R. Scheidler, and A. Stein. Fast arithmetic on hyperelliptic curves via continued fraction expansions. In *Advances in coding theory and cryptography*, pages 200–243. World Scientific, 2007.

[7] M. J. Jacobson, Jr. and A. J. van der Poorten. Computational aspects of NUCOMP. In *Algorithmic Number Theory - ANTS-V*, volume 2369 of *Lecture Notes in Computer Science*, pages 120–133, Sydney, Australia, 2002. Springer-Verlag, Berlin.

[8] T. Lange. Formulae for arithmetic on genus 2 hyperelliptic cruves. *Applicable Algebra in Engineering, Communications and Computing*, 15:295–328, 2005.

[9] S. A. Lindner, L. Imbert, and M. J. Jacobson, Jr. Balanced NUCOMP. Submitted, preprint available at `https://github.com/salindne/divisorArithmetic/blob/master/Balanced_NUCOMP.pdf`.

[10] S.A. Lindner. Supporting material. Available at `https://github.com/salindne/divisorArithmetic`, 2020.

[11] D.J.M. Mireles-Morales. *Efficient arithmetic on hyperelliptic curves with real representation.* PhD thesis, University of London, 2009.

[12] D. Shanks. On Gauss and composition I, II. In R.A. Mollin, editor, *Proc. NATO ASI on Number Theory and Applications*, pages 163–179. Kluwer Academic Press, 1989.

[13] A.V. Sutherland. Sato-Tate Distributions. *arXiv e-prints*, page arXiv:1604.01256, Apr 2016.

[14] A.V. Sutherland. Fast jacobian arithmetic for hyperelliptic curves of genus 3. *ANTS XIII*, 2:425–442, 2019.