Counting points on smooth plane quartics

Edgar Costa (MIT)

Simons Collaboration on Arithmetic Geometry, Number Theory, and Computation

August, 2022

Algorithmic Number Theory Symposium XV (ANTS)

Slides available at `edgarcosta.org`

Joint work with David Harvey and Andrew Sutherland.

L-function of a smooth projective curve

$X/\mathbb{Q}$ smooth projective curve of genus g.

$$L(X,s) = \sum_{n \geqslant 1} \frac{a_n}{n^s} = \prod_p \frac{1}{L_p(p^{-s})},$$

where $L_p(T) \in 1 + T\mathbb{Z}[T]$ and $\deg L_p(T) \leqslant 2g$.

$$Z_p(T) := \exp\left(\sum_{r \geqslant 1} \#X(\mathbb{F}_{p^r}) \frac{T^r}{r}\right) = \frac{L_p(T)}{(1-T)(1-pT)}$$

in particular

$$a_p = p + 1 - \#X(\mathbb{F}_p)$$

Understanding $a_n \to$ Birch–Swinnerton-Dyer, Lang–Trotter, Sato–Tate, ...

We need fast algorithms to compute a_p

$$L(X,s) = \sum_{n \geqslant 1} \frac{a_n}{n^s} = \prod_p \frac{1}{L_p(p^{-s})} = \prod_p \frac{1}{1 - a_p p^{-s} + \cdots},$$

To compute a_n for $n \leqslant N$ we only need to compute a_{p^e} for $p^e \leqslant N$.

We need fast algorithms to compute a_p

$$L(X,s) = \sum_{n\geqslant 1} \frac{a_n}{n^s} = \prod_p \frac{1}{L_p(p^{-s})} = \prod_p \frac{1}{1 - a_p p^{-s} + \cdots},$$

To compute a_n for $n \leqslant N$ we only need to compute a_{p^e} for $p^e \leqslant N$.

The cost is dominated by computing $a_p := \mathrm{Tr}(\mathrm{Frob}_p\,|H^1(X)) \in [-2g\sqrt{p}, 2g\sqrt{p}]$

We need fast algorithms to compute a_p

The cost is dominated by computing $a_p := \mathrm{Tr}(\mathrm{Frob}_p \,|H^1(X)) \in [-2g\sqrt{p}, 2g\sqrt{p}]$

We know how to do this efficiently in $N(\log N)^{O(1)}$ for cyclic covers of $\mathbb{P}^1$

$$X : y^m = f(x), \qquad f \in \mathbb{Z}[x]$$

- elliptic: Schoof (1985) — using group structure

We need fast algorithms to compute a_p

The cost is dominated by computing $a_p := \mathrm{Tr}(\mathrm{Frob}_p \,|H^1(X)) \in [-2g\sqrt{p}, 2g\sqrt{p}]$

We know how to do this efficiently in $N(\log N)^{O(1)}$ for cyclic covers of $\mathbb{P}^1$

$$X : y^m = f(x), \qquad f \in \mathbb{Z}[x]$$

- elliptic: Schoof (1985) — using group structure
- hyperelliptic: Harvey (2014), Harvey–Sutherland (2014, 2016)
- geometric hyperelliptic: Harvey–Massierer–Sutherland (2016)
- superelliptic: Sutherland (2020)

Apart from Schoof's, they all work by computing g^2 coefficients of $f(x)^{O(p)}$.

We need fast algorithms to compute a_p

The cost is dominated by computing $a_p := \mathrm{Tr}(\mathrm{Frob}_p \,|H^1(X)) \in [-2g\sqrt{p}, 2g\sqrt{p}]$

We know how to do this efficiently in $N(\log N)^{O(1)}$ for cyclic covers of $\mathbb{P}^1$

$$X : y^m = f(x), \qquad f \in \mathbb{Z}[x]$$

- elliptic: Schoof (1985) — using group structure
- hyperelliptic: Harvey (2014), Harvey–Sutherland (2014, 2016)
- geometric hyperelliptic: Harvey–Massierer–Sutherland (2016)
- superelliptic: Sutherland (2020)

Apart from Schoof's, they all work by computing g^2 coefficients of $f(x)^{O(p)}$.

We need fast algorithms to compute a_p

The cost is dominated by computing $a_p := \mathrm{Tr}(\mathrm{Frob}_p \,|H^1(X)) \in [-2g\sqrt{p}, 2g\sqrt{p}]$

We know how to do this efficiently in $N(\log N)^{O(1)}$ for cyclic covers of $\mathbb{P}^1$

$$X : y^m = f(x), \qquad f \in \mathbb{Z}[x]$$

- elliptic: Schoof (1985) — using group structure
- hyperelliptic: Harvey (2014), Harvey–Sutherland (2014, 2016)
- geometric hyperelliptic: Harvey–Massierer–Sutherland (2016)
- superelliptic: Sutherland (2020)

Apart from Schoof's, they all work by computing g^2 coefficients of $f(x)^{O(p)}$.

Today: smooth plane quartics

Smooth plane quartics

Smooth plane quartics are generic $g = 3$ curves given as

$$X : f(x_0, x_1, x_2) = 0, \qquad f \in \mathbb{Z}[x_0, x_1, x_2], \quad \deg f = 4$$

and computing $a_p := \mathrm{Tr}(\mathrm{Frob}_p \,|H^1(X))$ for $p \leqslant N$ in $N(\log N)^{O(1)}$

We will present three algorithms to do this.

We will in fact compute the Cartier–Manin matrix $C_p \in \mathbb{F}_p^{3\times 3}$.

Suffices for p large enough, as $\mathrm{Tr}\, C_p \equiv a_p \bmod p$ and $a_p \in [-6\sqrt{p}, 6\sqrt{p}]$.

Smooth plane quartics

Smooth plane quartics are generic $g = 3$ curves given as

$$X : f(x_0, x_1, x_2) = 0, \qquad f \in \mathbb{Z}[x_0, x_1, x_2], \quad \deg f = 4$$

and computing $a_p := \mathrm{Tr}(\mathrm{Frob}_p \,|H^1(X))$ for $p \leqslant N$ in $N(\log N)^{O(1)}$

We will present three algorithms to do this.

We will in fact compute the Cartier–Manin matrix $C_p \in \mathbb{F}_p^{3\times 3}$.

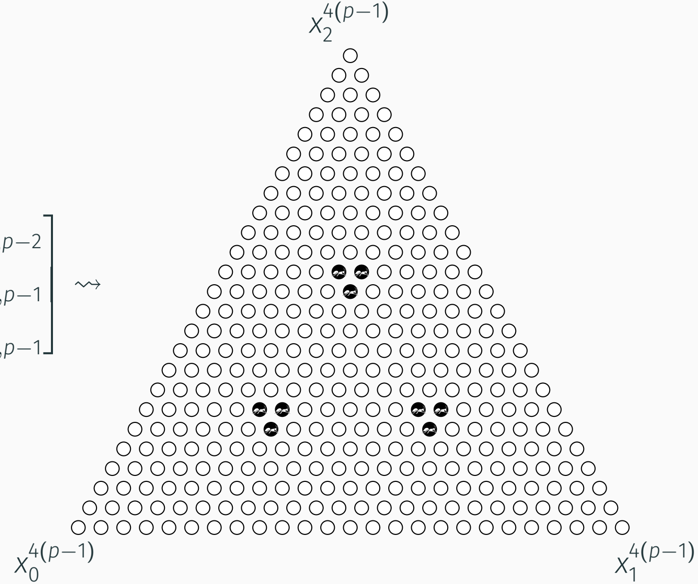
Suffices for p large enough, as $\mathrm{Tr}\, C_p \equiv a_p \bmod p$ and $a_p \in [-6\sqrt{p}, 6\sqrt{p}]$.

$$C_p := \begin{bmatrix} f^{p-1}_{p-1,\,p-1,\,2p-2} & f^{p-1}_{2p-1,\,p-1,\,p-2} & f^{p-1}_{p-1,\,2p-1,\,p-2} \\ f^{p-1}_{p-2,\,p-1,\,2p-1} & f^{p-1}_{2p-2,\,p-1,\,p-1} & f^{p-1}_{p-2,\,2p-1,\,p-1} \\ f^{p-1}_{p-1,\,p-2,\,2p-1} & f^{p-1}_{2p-1,\,p-2,\,p-1} & f^{p-1}_{p-1,\,2p-2,\,p-1} \end{bmatrix},$$

where $f^{p-1}_{i,j,k}$ denotes the coefficient of the term $x_0^i x_1^j x_2^k$ in $f(x_0, x_1, x_2)^{p-1}$.

Visualization of the Cartier–Manin matrix for $p = 7$

$$\begin{bmatrix} f^{p-1}_{p-1,\,p-1,\,2p-2} & f^{p-1}_{2p-1,\,p-1,\,p-2} & f^{p-1}_{p-1,\,2p-1,p-2} \\ f^{p-1}_{p-2,\,p-1,\,2p-1} & f^{p-1}_{2p-2,\,p-1,\,p-1} & f^{p-1}_{p-2,\,2p-1,p-1} \\ f^{p-1}_{p-1,\,p-2,\,2p-1} & f^{p-1}_{2p-1,\,p-2,\,p-1} & f^{p-1}_{p-1,\,2p-2,p-1} \end{bmatrix} \rightsquigarrow$$



Average polynomial time algorithms

These algorithms work via the computation of partial products of $r \times r$ matrices

$$M_0, \ldots, M_{N-1} \in \mathbb{Z}^{r \times r}$$

reduced modulo coprime integers

$$m_0, \ldots, m_{N-1} \in \mathbb{Z}$$

This can be accomplished in $O(r^2 N \log^3 N)$ time using $O(r^2 N \log N)$ space via an accumulating remainder tree.

In a simplified way, how small can we take r for smooth plane quartics?

Making r smaller will have other side effects, but in our 3 scenarios these are less significant.

We present three possibilities for $r \in \{66, 28, 16\}$

Old algorithm: (Optimized) Harvey – Computing zeta func. of arithmetic schemes

$$X : f(x_0, x_1, x_2) = 0, \qquad f \in \mathbb{Z}[x_0, x_1, x_2]$$

Consider the auxiliary polynomial $g = x_0^4 + x_1^4 + x_2^4$.

By looking at the binomial expansion of $(f + tg)^{p-1}$, where t is an auxiliary parameter, for certain sets of monomials S, we can construct $M_i \in \mathbb{Z}^{66\times 66}$ so

$$M_i \cdot \left(g^{(p-1)-i} f^i\right)\Big|_S = \left(g^{(p-1)-i-1} f^{i+1}\right)\Big|_S \bmod p.$$

Using these matrices we can reduce the problem of computing C_p to a single accumulating remainder tree, and we only need 3 rows of the end result.

Computing

$$V_0 M_0 M_0 \cdots M_k \bmod m_k,$$

with $V_0 \in \{0,1\}^{3\times 66}$ instead of $M_0 \cdots M_k \bmod m_k$.

New algorithm

Key idea: there are relations between the neighbouring coefficients of f^m.

In particular, f^m satisfies the following system of equations:

$$\partial_i(fg) = (m+1)(\partial_i f)g, \qquad i = 0, \dots, 2,$$

where $\partial_i := x_i \partial / \partial x_i$ and g a polynomial of degree $4m$.

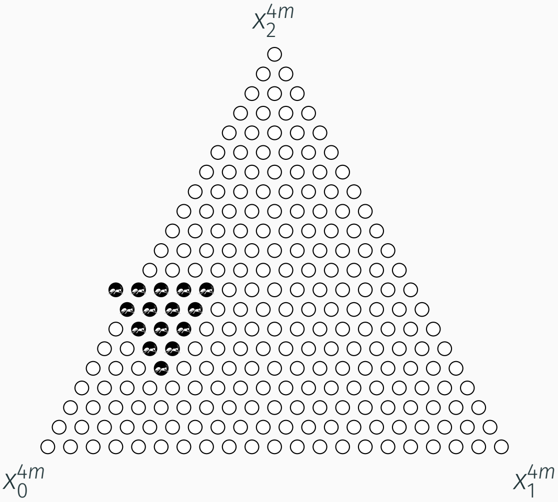
Looking at the coefficient of x^w gives rise to a system linear equations

$$w_i \sum_{\deg x^t = d} f_t g_{w-t} = (m+1) \sum_{\deg x^t = d} t_i f_t g_{w-t}, \qquad i = 0, \dots, 2.$$

(The Euler identity implies that one of these 3 equations is redundant.)

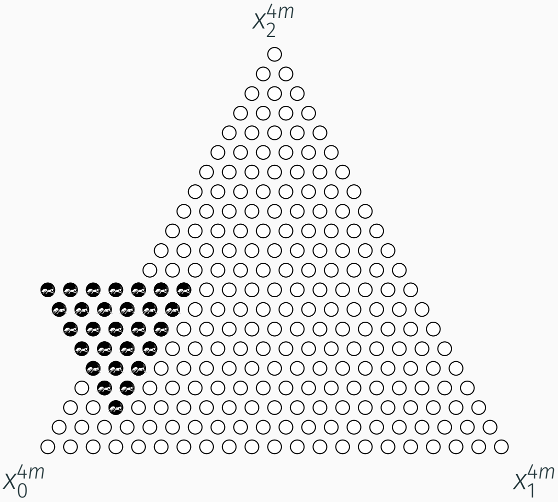
Example $d = 4$ and $m = 4$

We expect there to be two independent relations involving the dots.



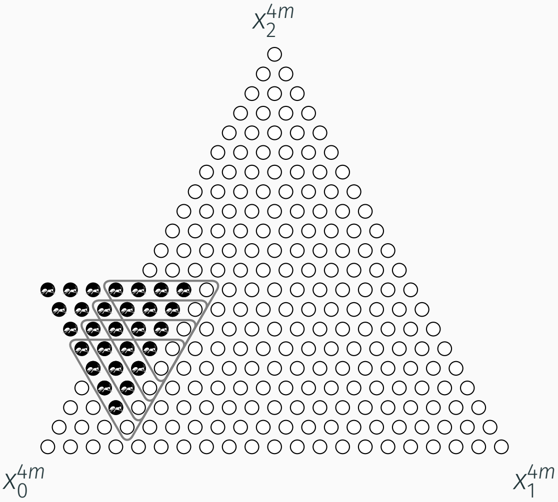
Example $d = 4$ and $m = 4$

Combining enough relations we can to move a larger triangle.



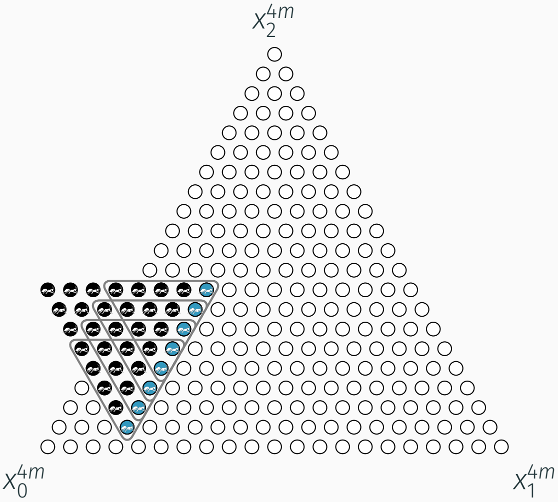
Example $d = 4$ and $m = 4$

Combining enough relations we can to move a larger triangle.



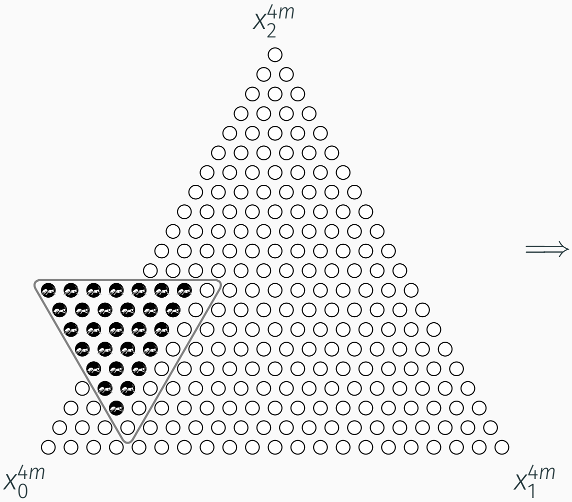
Example $d = 4$ and $m = 4$

Combining enough relations we can to move a larger triangle.

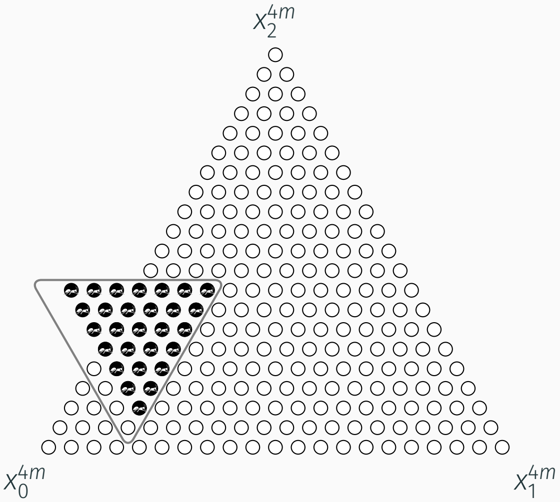


Example $d = 4$ and $m = 4$

Combining enough relations we can to move a larger triangle.



$\Longrightarrow$



Nondegeneracy condition

One is able to move if we assume some nondegeneracy conditions about X:

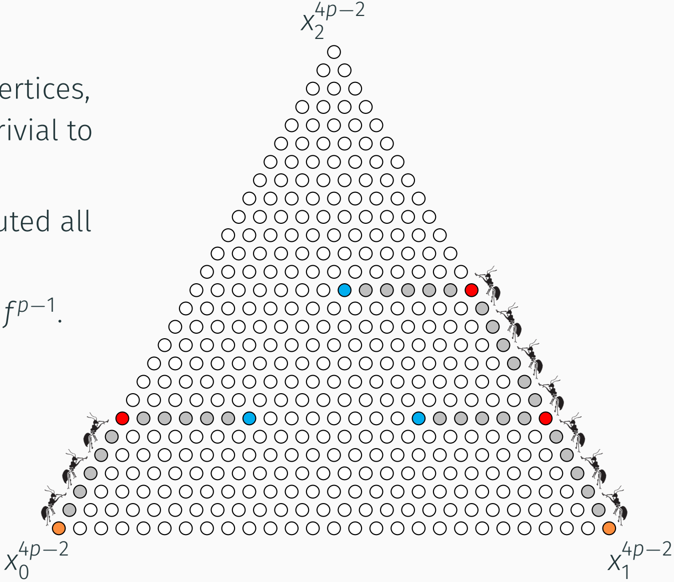
- $f(1,0,0)f(0,1,0)f(0,0,1) \neq 0$
  $\Leftrightarrow X$ does not pass through the points $(1,0,0), (0,1,0)$, and $(0,0,1)$.
- $f(0,x_1,x_2), f(x_0,0,x_2), f(x_0,x_1,0)$ are square free
  $\Leftrightarrow X$ intersects the coordinate axes transversally.

Nondegeneracy is a very mild condition.
Almost every smooth plane quartic has a nondegenerate model.

If we are given an equation that is not nondegenerate, a random coordinate change will likely produce a nondegenerate one (provided p is not too small), and this does not change a_p or $\#X(\mathbb{F}_p)$.

New algorithm

1. Start with a triangle at one of the vertices, where the coefficients of f^{p-2} are trivial to compute.
2. Walk it around until we have computed all the target coefficients of f^{p-2}.
3. Deduce the relevant coefficients of f^{p-1}.



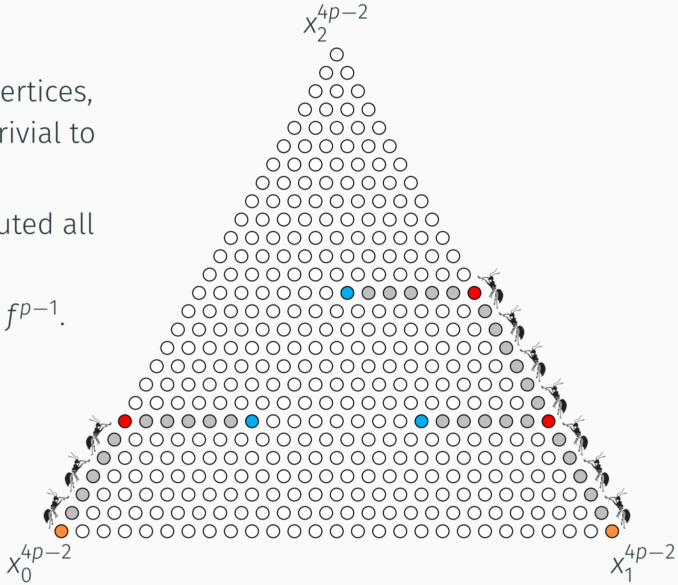
New algorithm

1. Start with a triangle at one of the vertices, where the coefficients of f^{p-2} are trivial to compute.
2. Walk it around until we have computed all the target coefficients of f^{p-2}.
3. Deduce the relevant coefficients of f^{p-1}.

Complexity:

-  $\log^{2+o(1)} p$ time
-  on average $\log p^{3+o(1)}$ time

one ART involving 28×28 matrices



New algorithm (compressed version)

Despite computing the Cartier–Manin matrix we have not yet used smoothness.

Under the smoothness assumption one observes:

- the 28×28 matrices have rank 16.
- 28/36 coefficients in the previous ∇'s belong to 16 dimensional vector spaces.

This allows us to replace 28×28 matrices with 16×16 matrices.

Compressing isn't free!
The size of the coefficients in the matrices increase.
We can amortize this cost and still obtain a speedup factor at least $3 \sim (28/16)^2$.

Timings: average polynomial time versions

	16 × 16		28 × 28		Harvey (optimized)	
N	seconds	ms/p	seconds	ms/p	seconds	ms/p
2^{10}	0.060	0.355	0.151	0.903	0.092	0.550
2^{12}	0.280	0.500	1.12	2.01	0.592	1.06
2^{14}	1.47	0.774	7.00	3.69	6.66	3.34
2^{16}	8.08	1.24	36.9	5.65	74.4	11.4
2^{17}	19.2	1.57	85.2	6.96	252	20.5
2^{18}	44.8	1.95	192	8.37	676	29.4
2^{19}	106	2.44	437	10.1	1680	38.6
2^{20}	241	2.94	991	12.1	4100	50.0
2^{21}	543	3.49	2230	14.3	10800	69.3
2^{22}	1260	4.26	5040	17.0	29900	101
2^{23}	2950	5.23	11400	20.3	88200	156

Timings: quasilinear methods

p	Cartier–Manin matrix 16×16	Cartier–Manin matrix 28×28	Cartier–Manin matrix Harvey (opt.)	Cartier–Manin matrix Costa	point counting `smalljac`	point counting `magma`
$2^{10}+7$	0.001	0.000	0.001	0.014	0.000	0.000
$2^{12}+3$	0.002	0.000	0.006	0.023	0.001	0.020
$2^{14}+27$	0.009	0.002	0.023	0.058	0.004	0.070
$2^{16}+1$	0.033	0.006	0.089	0.192	0.023	0.300
$2^{18}+3$	0.130	0.024	0.368	0.718	0.078	1.23
$2^{20}+7$	0.527	0.092	1.41	2.84	0.324	5.50
$2^{22}+15$	2.11	0.370	5.65	11.3	1.47	23.9
$2^{24}+43$	8.43	1.46	23.4	44.9	6.44	99.3
$2^{26}+15$	33.7	5.83	90.4	180	26.9	723
$2^{28}+3$	135	23.4	361	719	114	3080
$2^{30}+3$	539	93.1	1480	3130	465	13600

Timings: against other genus 3 methods

N	plane quartic	geometrically hyperelliptic	rationally hyperelliptic	2-cover of a genus 1 curve	3-cover of $\mathbb{P}^1$	4-cover of $\mathbb{P}^1$
2^{10}	0.058	0.053	0.007	0.021	0.006	0.006
2^{12}	0.281	0.126	0.011	0.070	0.008	0.008
2^{14}	1.49	0.724	0.065	0.326	0.030	0.028
2^{16}	8.00	5.42	0.829	1.77	0.333	0.285
2^{18}	44.6	29.6	10.0	10.1	2.38	2.15
2^{20}	241	168	55.6	57.2	15.3	12.2
2^{21}	543	388	133	133	36.1	29.6
2^{22}	1260	921	320	315	87.6	72.0
2^{23}	2950	2160	746	748	214	173
2^{24}	6840	4860	1760	1750	514	410
2^{25}	15600	11200	4120	4050	1220	975
2^{26}	35600	26000	9560	9370	2880	2350

counter

- Harvey (2018) < 100
- Costa (2022) = 210

and a 15^2 safety net