Quadratic Isogeny Primes github.com/barinderbanwait/quadratic_isogeny_primes

arxiv.org/abs/2101.02673 - submitted

Barinder Singh Banwait

Harish-Chandra Research Institute

MEGA 2021 Virtually in Tromsø, Norway June 07-11, 2021





Copyright Disclaimer: Photo credits and attribution are given on the final slide.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
●00000000	00000000000	000000	0	0000	000

Isogenies

lsogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0●0000000		000000	O	0000	000
Rational	Isogenies				

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0●0000000	000000000000		O	0000	000
Rational	lsogenies				

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0●0000000	000000000000	000000	O	0000	000
Rational	Isogenies				

Definition

An isogeny $\phi: E_1 \rightarrow E_2$ is a non-constant morphism of curves which

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0●0000000	000000000000	000000	O	0000	000
Rational	lsogenies				

Definition

An isogeny $\phi: E_1 \rightarrow E_2$ is a non-constant morphism of curves which maps O_{E_1} to O_{E_2} ;

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0●0000000	000000000000	000000	O	0000	000
Rational	lsogenies				

Definition

An isogeny $\phi : E_1 \to E_2$ is a non-constant morphism of curves which maps O_{E_1} to O_{E_2} ; \Leftrightarrow induces a group homomorphism from $E_1(\overline{K})$ to $E_2(\overline{K})$;

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0●0000000	000000000000	000000	O	0000	000
Rational	lsogenies				

Definition

An isogeny $\phi: E_1 \to E_2$ is a non-constant morphism of curves which maps O_{E_1} to O_{E_2} ; \Leftrightarrow induces a group homomorphism from $E_1(\overline{K})$ to $E_2(\overline{K})$; \Leftrightarrow has finite kernel.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0●0000000	000000000000	000000	O	0000	000
Rational	lsogenies				

Definition

An isogeny $\phi: E_1 \rightarrow E_2$ is a non-constant morphism of curves which maps O_{E_1} to O_{E_2} ;

- \Leftrightarrow induces a group homomorphism from $E_1(\overline{K})$ to $E_2(\overline{K})$;
- $\Leftrightarrow \text{ has finite kernel.}$

The degree of $\phi = |\ker(\phi)| = [\overline{K}(E_1) : \phi^*\overline{K}(E_2)].$

lsogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0●0000000		000000	O	0000	000
Rational	Isogenies				

Definition

An isogeny $\phi : E_1 \to E_2$ is a non-constant morphism of curves which maps O_{E_1} to O_{E_2} ;

- \Leftrightarrow induces a group homomorphism from $E_1(\overline{K})$ to $E_2(\overline{K})$;
- \Leftrightarrow has finite kernel.

The degree of $\phi = |\ker(\phi)| = [\overline{K}(E_1) : \phi^*\overline{K}(E_2)].$

 ϕ is said to be *K*-rational if it is compatible with the *G_K*-action on *E*₁ and *E*₂; that is, if the following diagram commutes for all $\sigma \in G_K$:

$$\begin{array}{ccc} E_1 & \stackrel{\phi}{\longrightarrow} & E_2 \\ \hline & & & \downarrow^{\sigma} \\ \hline & & & \downarrow^{\sigma} \\ E_1 & \stackrel{\phi}{\longrightarrow} & E_2 \end{array}$$

lsogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00●000000	000000000000	000000	O	0000	000
Isogenies	s = Kernel				

lsogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00●000000		000000	O	0000	000
Isogenies	s = Kernel				

Fact

Let E/K be an elliptic curve over a number field. Then there is a bijection

lsogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00●000000		000000	O	0000	000
Isogenies	= Kernel				

Fact

Let E/K be an elliptic curve over a number field. Then there is a bijection

 $\{K\text{-rational isogenies from } E\} \xrightarrow{\sim} \{G_K\text{-invariant finite subgroups of } E(\overline{K})\}$ $\phi \longmapsto \ker \phi$ $\phi_C : E \to E/C \longleftrightarrow C.$

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000					
Isogenies	s = Kernel				

Fact

Let E/K be an elliptic curve over a number field. Then there is a bijection

$$\{K\text{-rational isogenies from } E\} \xrightarrow{\sim} \{G_K\text{-invariant finite subgroups of } E(\overline{K})\}$$
$$\phi \longmapsto \ker \phi$$
$$\phi_C : E \to E/C \longleftrightarrow C.$$

Slogan

You can identify an isogeny with its kernel.

Isogenies 000●00000	PreTypeOneTwoPrimes	TypeTwoPrimes 000000	Live Demo O	Weeding 0000	WIP 000
The Dre	am				

"Understand rational isogenies."

lsogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000●00000		000000	O	0000	000
The Dre	am				

"Understand rational isogenies."

Since we can identify isogenies with their kernels,

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000					
The Dre	am				

"Understand rational isogenies."

Since we can identify isogenies with their kernels, which are finite abelian groups,

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000●00000		000000	O	0000	000
The Drea	am				

"Understand rational isogenies."

Since we can identify isogenies with their kernels, which are finite abelian groups, which break up as a direct sum of cyclic groups,

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000					
The Dre	am				

"Understand rational isogenies."

Since we can identify isogenies with their kernels, which are finite abelian groups, which break up as a direct sum of cyclic groups, the above goal reduces to

Reduced Goal

"Understand rational isogenies with cyclic kernel."

Call these cyclic K-isogenies.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	00000000000	000000		0000	000

Question

For a number field K, what possible degrees arise as the degree of a K-rational cyclic isogeny between elliptic curves over K?

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000					

Question

For a number field K, what possible degrees arise as the degree of a K-rational cyclic isogeny between elliptic curves over K?

Let's call this set of possible degrees lsogCyclicDeg(K).

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000					

Question

For a number field K, what possible degrees arise as the degree of a K-rational cyclic isogeny between elliptic curves over K?

Let's call this set of possible degrees lsogCyclicDeg(K).

We write lsogPrimeDeg(K) for the primes in this set, and call them isogeny primes for K.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000					

Question

For a number field K, what possible degrees arise as the degree of a K-rational cyclic isogeny between elliptic curves over K?

Let's call this set of possible degrees lsogCyclicDeg(K).

We write lsogPrimeDeg(K) for the primes in this set, and call them isogeny primes for K.

A priori these could be infinite sets.

The Theorems of Mazur and Kenku



Barry C. Mazur



Monsur A. Kenku

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000				0000	000
	.				

The Theorems of Mazur and Kenku

Theorem (Mazur, 1978)

$\mathsf{IsogPrimeDeg}(\mathbb{Q}) = \{2, 3, 5, 7, 11, 13, 17, 19, 37, 43, 67, 163\}\,.$



Barry C. Mazur



Monsur A. Kenku

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000					

The Theorems of Mazur and Kenku

Theorem (Mazur, 1978)

 $\mathsf{IsogPrimeDeg}(\mathbb{Q}) = \{2, 3, 5, 7, 11, 13, 17, 19, 37, 43, 67, 163\}\,.$

Theorem (Kenku, 1982)

 $\mathsf{lsogCyclicDeg}(\mathbb{Q}) = \{1 \le N \le 19\} \cup \{21, 25, 27, 37, 43, 67, 163\}.$



Barry C. Mazur



Monsur A. Kenku

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0000000000	00000000000	000000		0000	000
—					

Beyond Mazur's Theorem

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP		
000000000	00000000000	000000		0000	000		
Devend Meruy's Theorem							

Beyond Mazur's Theorem

Question

Can one write down lsogPrimeDeg(K) for any other number field K?

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000					
	· · · · · ·				

Beyond Mazur's Theorem

Question

Can one write down IsogPrimeDeg(K) for any other number field K?

Theorem (B., 2021)

Assuming GRH, we have the following.

$$\begin{split} & \mathsf{IsogPrimeDeg}(\mathbb{Q}(\sqrt{7})) = \mathsf{IsogPrimeDeg}(\mathbb{Q}) \\ & \mathsf{IsogPrimeDeg}(\mathbb{Q}(\sqrt{-10})) = \mathsf{IsogPrimeDeg}(\mathbb{Q}) \\ & \mathsf{IsogPrimeDeg}(\mathbb{Q}(\sqrt{5})) = \mathsf{IsogPrimeDeg}(\mathbb{Q}) \cup \{23,47\} \end{split}$$

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0000000●0		000000	O	0000	000
Algorithm	for Quadratic	lsogeny Prin	nes		

Actually this is a corollary of the following.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0000000●0		000000	O	0000	000
Algorithm	for Quadratic	Isogeny Prin	nes		

Actually this is a corollary of the following.

Algorithm (B., 2021)

Let K be a quadratic field which is not imaginary quadratic of class number 1. Then there is an algorithm which computes a superset of $lsogPrimeDeg(K)^*$ as the union of three sets:

(*: With these assumptions, this is a finite set, as explained in next section)

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0000000€0		000000	O	0000	000
Algorithm	for Quadratic	Isogeny Prin	nes		

Actually this is a corollary of the following.

Algorithm (B., 2021)

Let K be a quadratic field which is not imaginary quadratic of class number 1. Then there is an algorithm which computes a superset of $lsogPrimeDeg(K)^*$ as the union of three sets:

 $\begin{aligned} \mathsf{IsogPrimeDeg}(\mathcal{K}) \subseteq \mathsf{PreTypeOneTwoPrimes}(\mathcal{K}) \cup \mathsf{TypeOnePrimes}(\mathcal{K}) \\ \cup \mathsf{TypeTwoPrimes}(\mathcal{K}). \end{aligned}$

(*: With these assumptions, this is a finite set, as explained in next section)

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
0000000€0		000000	O	0000	000
Algorithm	for Quadratic	Isogeny Prin	nes		

Actually this is a corollary of the following.

Algorithm (B., 2021)

Let K be a quadratic field which is not imaginary quadratic of class number 1. Then there is an algorithm which computes a superset of $lsogPrimeDeg(K)^*$ as the union of three sets:

 $\begin{aligned} \mathsf{IsogPrimeDeg}(\mathcal{K}) \subseteq \mathsf{PreTypeOneTwoPrimes}(\mathcal{K}) \cup \mathsf{TypeOnePrimes}(\mathcal{K}) \\ \cup \mathsf{TypeTwoPrimes}(\mathcal{K}). \end{aligned}$

(*: With these assumptions, this is a finite set, as explained in next section)

Remark

If K is imaginary quadratic of class number one, then lsogPrimeDeg(K) is infinite because of complex multiplication.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000					
_					

Preview of the Main Calling Function

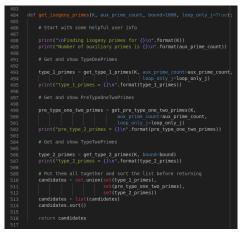
<pre>485 485 485 485tart with some helpful user info 487 487 487 print("\nFinding isogeny primes for (}\n".format(K)) print("Number of auxiliary primes is {}\n".format(aux_prime_count)) 491 491 491 491 491 491 491 492 493 493 493 493 493 493 493 493 493 493</pre>	
<pre>496 # Start with some helpful user info 497 print("neinding isogeny primes for ()\n".format(K)) 498 print("Number of auxiliary primes is ()\n".format(aux_prime_count)) 499 # Get and show TypeOnePrimes 493 type lprimes = get_type_lprimes(K, aux_prime_count, loog_only_lcog_only_l) 494 # Get and show PreTypeOneTwoPrimes 495 print("type_ingrimes = {)\n".format(type_lprime) 496 # Get and show PreTypeOneTwoPrimes 497 # Get and show PreTypeOneTwoPrimes 498 pre_type_one_two primes = get_pre_type_one_two_primes(K, 499 pre_type_one_two_primes = get_ont_aux_prime_count, loog_only_lcog_only_l) 496 print("type_lprimes = {)\n".format(type_type_one_two_primes)) 497 # Get and show TypeTwoPrimes 498 print("type_lprimes = ()\n".format(type_lprimes)) 498 # Get and show TypeTwoPrimes 499 print("type_lprimes = ()\n".format(type_lprimes)) 499 print("type_lprimes = ()\n".format(type_lprimes), set(pre_lprimes), set(pre_lprimes), set(pre_lprimes), set(pre_lprimes), set(pre_lprimes), set(pre_lprimes)) 498 # Gut the all together and sort the list before returning 499 candidates = ilsi(candidates) 490 type_lprimes = (long_lprimes), set(pre_lprimes), set(pre_lprimes)) 491 set(pre_lprimes) 492 candidates = lisi(candidates) 493 return candidates</pre>	
<pre>497 498 499 499 499 499 490 499 490 499 490 490</pre>	
<pre>488 print(`nfinding isogeny primes for ()n*.format(K)) 489 print(`Number of auxiliary primes is ()n*.format(aux_prime_count)) 489 489 fet and show TypeOnePrimes 482 493 type_lprimes = get_type_lprimes(K, aux_prime_count, icog_only_l=loog_only_l) 486 print(`type_lprimes = {}\n*.format(type_lprimes) 487 # Get and show PreTypeOneTwoPrimes 488 489 489 print(`type_one_two primes = get_pre_type_one_two_primes(K, aux_prime_count, icog_only_l=loog_only_l) 489 print(`type_lprimes = {}\n*.format(type_tpre_one_two_primes)) 489 # Get and show PreTypeOneTwoPrimes 489 pre_type_one_two primes = get_pre_type_one_two_primes)) 489 # Get and show TypeTwoPrimes 489 # Get and show TypeTwoPrimes 489 # Dut then all together and sort the list before returning 489 candidates = list(candidates) 480 set(pre_type_lprimes)) 480 set(pre_type_lprimes) 480 set(pre_type_lprimes)) 480 set(pre_type_lprimes) 480 set(pre_type_lprimes)) 480 set(pre_type_lprimes) 480 set(pre_type_lprimes)) 480 set(pre_type_lprim</pre>	# Start with some helpful user info
<pre>499 print("Number of auxiliary primes is ()\n".format(aux_prime_count)) 499 # @ Get and show TypeOnePrimes 492 493 type_lprimes = get_type_lprimes(K, aux_prime_count_aux_prime_count, 494 loop_only_lcoop_only_) 495 # @ det and show PreTypeOneTwoPrimes 496 # @ det and show PreTypeOneTwoPrimes 497 pre type one_two primes = @ t_pre_type_one_two primes(K, 498 aux_prime_count_aux_prime_count, 499 loop_only_lsoop_only_) 490 # @ det and show TypeTwoPrimes 499 pre_type_one_two_primes = @ t_pre_type_one_two_primes(K, 499 pre_type_one_two_primes = @ t_pre_type_one_two_primes(K, 499 pre_type_one_two_primes = @ t_pre_type_one_two_primes(K, 499 print("type_tprimes = @ t_pre_tpre_tpre_tpre_tpre_tpre_tpre_tpre</pre>	
<pre>499 49 # Get and show TypeOnePrimes 499 491 # Get and show TypeOnePrimes 499 494 495 495 495 496 496 496 497 496 497 496 497 496 498 498 498 498 498 498 498 498 498 498</pre>	
<pre>491 # 6 et and show TypeOnePrimes 492 493 494 495 496 497 498 498 499 499 499 499 499 499 499 499</pre>	<pre>print("Number of auxiliary primes is {}\n".format(aux_prime_count))</pre>
<pre>492 493 493 494 495 495 496 496 496 497 498 499 499 499 499 499 499 499 499 499</pre>	
493 type_lprimes = get_type_lprimes(k, aux_prime_count, aux_prime_count, locg only_i)=loop_only_j) 493 print("type_lprimes = ()\n".format(type_lprimes)) 494 idoo notice 495 # Get and show PreTypeOneTwoPrimes 496 pre_type_one_two primes = get_pre_type one two primes (k, prime count-aux prime count-aux prime count) 497 # Get and show PreTypeOneTwoPrimes 498 pre_type_one_two primes = get_pre_type_one_two_primes) 509 print("typeprimes = get_pre_type_one_two_primes)) 501 print("typeprimes = get_type_2primes(type_one_two_primes)) 503 # Get and show TypeTwoPrimes 504 # Get and show TypeTwoPrimes 505 type_2_primes = get_type_2_primes(k, bound-bound) 506 type_2_primes = get_type_2_primes(k, bound-bound) 507 geture and sort the list before returning 508 set(pre_type_one_two_primes), set(pre_type_one_two_primes), set(pre_type_one_two_primes), set(pre_type_one_two_primes), set(pre_type_one_two_primes)) 508 return candidates = list(candidates) 509 return candidates	
<pre>494 995 907 908 909 909 909 909 909 909 909 909 909</pre>	
<pre>495 print("type_l_primes = ()\n".format(type_l_primes)) 496 497 # Get and show PreTypeOneTwoPrimes 498 498 498 499 pre_type one two primes = get_pre_type one two primes(K, 490 autoprimes) 490 autoprimes 491 autoprimes 492 print("type_type_one_two_primes)) 593 # Get and show TypeTwoPrimes 595 595 type_primes = g(\\n".format(type_primes)) 596 # Put then all together and sort the list before returning 597 candidates = set.unoin(set(type_primes)) 598 set(pre_type_one_two_primes), 599 set(pre_type_one_two_primes), 591 set(pre_type_one_two_primes), 592 set(pre_type_one_two_primes), 593 candidates = list(candidates) 594 seturn candidates 595 trutra candidates 5 595 trutra 5 595</pre>	<pre>type_1_primes = get_type_1_primes(K, aux_prime_count=aux_prime_count,</pre>
<pre>496 497 # Get and show PreTypeOneTwoPrimes 499 497 # Get and show PreTypeOneTwoPrimes 499 499 498 # Get and show TypeTwoPrimes 504 # Get and show TypeTwoPrimes 505 505 506 # Put them all together and sort the list before returning 508 candidates = set.union(set(type_l primes)) 509 500 # Put them all together and sort the list before returning 500 candidates = list(candidates) 501 set(pre_type_one_two_primes)) 502 from the set of th</pre>	
497 # Get and show PreTypeOneTwoPrimes 498 pre_type_one_two primes = get_pre_type_two_primes(K, 609	<pre>print("type_1_primes = {}\n".format(type_1_primes))</pre>
<pre>499 499 499 499 499 499 499 499 499 499</pre>	
499 pre_type_one_two_primes = get_pre_type_one_two_primes(K, 500 aux_prime_count, 501 loop_only_i=loop_only_j) 502 print("pre_type_2_primes = (\\n".format(pre_type_one_two_primes))) 503 # Get and show TypeTvoPrimes 506 type_2_primes = d(\\n".format(type_2_primes)) 507 print("type_type_2_primes = (\\n".format(type_2_primes)) 508 # Put then all together and sort the list before returning 509 candidates = set.union(set(type_1_primes)) 511 set(pre_type_one_two_primes)) 512 set(pre_type_one_two_primes)) 513 candidates = list(candidates) 514 return candidates	
<pre>500 aux prime count, 500 brint('pre_type_2_primes = {})\n'.format(pre_type_only_) 502 print('pre_type_2_primes = {})\n'.format(pre_type_one_two_primes)) 503 # Get and show TypeTwoPrimes 504 type_2_primes = get_type_2_primes(K, bound-bound) 505 print('typeprimes = {})\n'.format(type_2_primes)) 508 # But them all together and sort the list before returning 509 candidates = set_union(set(type_1_primes), 510 candidates = list(candidates) 511 candidates.sort() 512 candidates.sort()</pre>	
501 top only_l=loop.only_l) 502 print("pre_type_2.primes = (\\n".format(pre_type_one_two_primes)) 503 # Get and show TypeTvoPrimes 504 # Get and show TypeTvoPrimes 505 type_2.primes = (\\n".format(type_2.primes)) 506 type_2.primes = (\\n".format(type_2.primes)) 507 # Put then all together and sort the list before returning 508 # Put then all together and sort the list before returning 509 candidates = set.union(set(type_1.primes)) 511 set(pre_type_one_two_primes)) 512 set(pre_type_one_two_primes)) 513 candidates = list(candidates) 514 return candidates	
<pre>502 print("pre_type_2_primes = ()\n".format(pre_type_one_two_primes)) 503 504 # Get and show TypeTwoPrimes 505 506 507 print("sype_2_primes = ()\n".format(type_2_primes)) 508 # Put them all together and sort the list before returning 509 candidates = set.union(set(type_1_primes)) 510 511</pre>	
503 # fet and show TypeTwoPrimes 505 type 2_primes = get_type 2_primes(k, bound-bound) 507 print("type 2_primes = ()\n".format(type 2_primes)) 508 # Put then all together and sort the list before returning 509 candidates = set_union(set(type 1_primes)) 511	
<pre>594 # Get and show TypeTwoPrimes 505 506 type 2 primes = get type 2 primes(K, bound-bound) 507 print("type 2 primes = ()\n".format(type 2 primes)) 508 # Put them all together and sort the list before returning 500 candidates = set union(set(type 1 primes), 511</pre>	<pre>print("pre_type_2_primes = {}\n".format(pre_type_one_two_primes))</pre>
505 type 2_primes = get_type 2_primes(K, bound-bound) 507 print("type_2_primes = ()\n".format(type_2_primes)) 508 # Put then all together and sort the list before returning 509 andidates = set_union(set(type_1_primes), 511 set(pre_type_one_two_primes), 512 set(pre_type_one_two_primes), 513 candidates = list(candidates) 514 candidates. 515 return candidates	
506 type_2_primes = get_type_2_primes(k, bound-bound) 507 print("type_2_primes = ()\n".format(type_2_primes)) 508 # Put them all together and sort the list before returning 510 candidates = set.union(set(type_1_primes), 511 set(type_2_primes)) 512 set(type_2_primes)) 513 candidates = list(candidates) 514 candidates.set(type_2_primes))	# Get and show TypeTwoPrimes
507 print("type_2_primes = ()\n".format(type_2_primes)) 508 # Put then all together and sort the list before returning 509 candidates = set.union(set(type_1 primes), 511 set(pre.type_one_two_primes), 512 set(pre.type_one_two_primes)) 513 candidates = list(candidates) 514 candidates 515 return candidates	
508 * Put them all together and sort the list before returning 510 candidates = set.union(set(type_l primes), 511 set(type_low_ene_two_primes), 512 set(type_low_ines)) 513 candidates = list(candidates) 514 candidates.sort() 515 return candidates	
509 # Put then all together and sort the list before returning 510 candidates = set.unin(set(type 1 primes), 511 set(type 2 primes)) 512 set(type 2 primes)) 513 candidates = list(candidates) 514 candidates.set(type 2 primes)) 515 return candidates	<pre>print("type_2_primes = {}\n".format(type_2_primes))</pre>
510 candidates = set.union(set(type l.primes), 511 set(type type one type on	
511 set(pre Type one two primes), 512 set(type_2 primes)) 513 candidates = lisi(candidates) 514 candidates.sort() 515 return candidates	
512 set(type_2_primes)) 513 candidates = list(candidates) 514 candidates.sort() 515 516	
513 candidates = list(candidates) 514 candidates.sort() 515 516 return candidates	
514 candidates.sort() 515 515 516 return candidates	
515 516 return candidates	
516 return candidates	candidates.sort()
	return candidates

lsogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000●		000000	O	0000	000
D. t.		n e ser			

Preview of the Main Calling Function

Sage implementation available at

github.com/barinderbanwait/quadratic_isogeny_primes



lsogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000●		000000	O	0000	000
Droviow	of the Main Cal	ling Eurotion	_		

Preview of the Main Calling Function

Sage implementation available at

 $\verb|github.com/barinderbanwait/quadratic_isogeny_primes||$



We'll have a live-demo of the command-line tool after giving an overview of the algorithm.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000	0000000000				

PreTypeOneTwoPrimes

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	0●0000000000	000000	O	0000	000
The isog	eny character				

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	○●○○○○○○○○○	000000	O	0000	000
The isog	eny character				

Let E/K be an elliptic curve over a number field which admits a K-rational p-isogeny.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	0●0000000000	000000	O	0000	000
The isoge	eny character				

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	0●0000000000	000000	O	0000	000
The isog	eny character				

$$\lambda: G_K \longrightarrow \operatorname{Aut} V(\overline{K}) \cong \mathbb{F}_p^{\times},$$

lsogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	0●0000000000	000000	O	0000	000
The isoge	eny character				

$$\lambda: G_{\mathcal{K}} \longrightarrow \operatorname{Aut} V(\overline{\mathcal{K}}) \cong \mathbb{F}_{p}^{\times},$$

where V is the kernel of the isogeny

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	○●○○○○○○○○○	000000	O	0000	000
The isog	eny character				

$$\lambda: G_{\mathcal{K}} \longrightarrow \operatorname{Aut} V(\overline{\mathcal{K}}) \cong \mathbb{F}_{p}^{\times},$$

where V is the kernel of the isogeny, which can be thought of as a 1d G_K -representation.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	○●○○○○○○○○○	000000	O	0000	000
The isog	eny character				

$$\lambda: G_{\mathcal{K}} \longrightarrow \operatorname{Aut} V(\overline{\mathcal{K}}) \cong \mathbb{F}_{p}^{\times},$$

where V is the kernel of the isogeny, which can be thought of as a 1d G_K -representation.

Isogenies of prime degree over number fields

FUMIYUKI MOMOSE

Department of Mathematics, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112, Japan

Received 19 August 1993; accepted in final form 24 December 1993

In 1993, Momose classified isogenies into three types.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	00●000000000	000000	O	0000	000
Momose's	Classification	of Isogenies	into three	types	

Let K be a number field. Then there exists a constant $C_0 = C_0(K)$ such that for any prime $p > C_0$, and for any elliptic curve admitting a K-rational p-isogeny, the isogeny character λ falls into one of the following three types:

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
Momose's	Classification	of leagenies	into three		000

Let K be a number field. Then there exists a constant $C_0 = C_0(K)$ such that for any prime $p > C_0$, and for any elliptic curve admitting a K-rational p-isogeny, the isogeny character λ falls into one of the following three types:

Type 1. λ^{12} or $(\lambda \theta_p^{-1})^{12}$ is unramified $(\theta_p = mod-p \text{ cyclotomic character})$.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	00000000000	000000		0000	000
Momose	's Classification	of Isogenies	into three	types	

Let K be a number field. Then there exists a constant $C_0 = C_0(K)$ such that for any prime $p > C_0$, and for any elliptic curve admitting a K-rational p-isogeny, the isogeny character λ falls into one of the following three types:

Type 1. $\lambda^{12} \text{ or } (\lambda \theta_p^{-1})^{12}$ is unramified ($\theta_p = \text{mod-}p$ cyclotomic character). Type 2. $\lambda^{12} = \theta_p^6$ and $p \equiv 3 \pmod{4}$.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	00000000000	000000		0000	000
Momose	's Classification	of Isogenies	into three	types	

Let K be a number field. Then there exists a constant $C_0 = C_0(K)$ such that for any prime $p > C_0$, and for any elliptic curve admitting a K-rational p-isogeny, the isogeny character λ falls into one of the following three types:

Type 1. λ^{12} or $(\lambda \theta_p^{-1})^{12}$ is unramified $(\theta_p = mod-p \text{ cyclotomic character})$.

Type 2. $\lambda^{12} = \theta_p^6$ and $p \equiv 3 \pmod{4}$.

Type 3. K contains the Hilbert class field H_L of an imaginary quadratic field L.

Isogenies 000000000	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
	Classification		into three		000

Let K be a number field. Then there exists a constant $C_0 = C_0(K)$ such that for any prime $p > C_0$, and for any elliptic curve admitting a K-rational p-isogeny, the isogeny character λ falls into one of the following three types:

Type 1. λ^{12} or $(\lambda \theta_p^{-1})^{12}$ is unramified $(\theta_p = mod-p \text{ cyclotomic character})$.

Type 2. $\lambda^{12} = \theta_p^6$ and $p \equiv 3 \pmod{4}$.

Type 3. K contains the Hilbert class field H_L of an imaginary quadratic field L. The rational prime p splits in L:

$$p\mathcal{O}_L = \mathfrak{p}\overline{\mathfrak{p}}.$$

Isogenies 000000000	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
	Classification		into three		000

Let K be a number field. Then there exists a constant $C_0 = C_0(K)$ such that for any prime $p > C_0$, and for any elliptic curve admitting a K-rational p-isogeny, the isogeny character λ falls into one of the following three types:

Type 1. λ^{12} or $(\lambda \theta_p^{-1})^{12}$ is unramified $(\theta_p = mod-p \text{ cyclotomic character})$.

Type 2. $\lambda^{12} = \theta_p^6$ and $p \equiv 3 \pmod{4}$.

Type 3. K contains the Hilbert class field H_L of an imaginary quadratic field L. The rational prime p splits in L:

$$p\mathcal{O}_L = \mathfrak{p}\overline{\mathfrak{p}}.$$

For any prime q of K prime to \mathfrak{p} ,

$$\lambda^{12}(\mathsf{Frob}_{\mathfrak{q}}) = \alpha^{12} \pmod{\mathfrak{p}}$$

for any $\alpha \in K^{\times}$ with $\alpha \mathcal{O}_L = \operatorname{Nm}_{K/L}(\mathfrak{q})$.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	000●00000000	000000	O	0000	000
PreType(DneTwoPrimes				



Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	000●00000000	000000	O	0000	000
PreType	OneTwoPrimes				

If K is a quadratic field which is not imaginary quadratic of class number one, then there is a finite set of primes PreTypeOneTwoPrimes(K) outside of which the isogeny character is of Type 1 or 2.



Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	000€00000000	000000	O	0000	000
PreType	OneTwoPrimes				

If K is a quadratic field which is not imaginary quadratic of class number one, then there is a finite set of primes PreTypeOneTwoPrimes(K) outside of which the isogeny character is of Type 1 or 2.

From Momose's Theorem, we could take

 $\mathsf{PreTypeOneTwoPrimes}(\mathcal{K}) = \{p \text{ prime } : p < C_0\},\$



Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	000€0000000	000000	O	0000	000
PreType	OneTwoPrimes				

If K is a quadratic field which is not imaginary quadratic of class number one, then there is a finite set of primes PreTypeOneTwoPrimes(K) outside of which the isogeny character is of Type 1 or 2.

From Momose's Theorem, we could take

 $\mathsf{PreTypeOneTwoPrimes}(\mathcal{K}) = \{ p \text{ prime } : p < C_0 \},\$

but we show that it's possible to take the primes dividing a handful of explicitly computable integers.



Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	000€0000000	000000	O	0000	000
PreType	OneTwoPrimes				

If K is a quadratic field which is not imaginary quadratic of class number one, then there is a finite set of primes PreTypeOneTwoPrimes(K) outside of which the isogeny character is of Type 1 or 2.

From Momose's Theorem, we could take

 $\mathsf{PreTypeOneTwoPrimes}(\mathcal{K}) = \{p \text{ prime } : p < C_0\},\$

but we show that it's possible to take the primes dividing a handful of explicitly computable integers.

Theorem (Momose, Theorem B)

Let K be a quadratic field which is not an imaginary quadratic field of class number 1. Then lsogPrimeDeg(K) is finite.



Fumiyuki Momose

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	000€0000000	000000	O	0000	000
PreType	OneTwoPrimes				

If K is a quadratic field which is not imaginary quadratic of class number one, then there is a finite set of primes PreTypeOneTwoPrimes(K) outside of which the isogeny character is of Type 1 or 2.

From Momose's Theorem, we could take

 $\mathsf{PreTypeOneTwoPrimes}(\mathcal{K}) = \{ p \text{ prime } : p < C_0 \},\$

but we show that it's possible to take the primes dividing a handful of explicitly computable integers.

Theorem (Momose, Theorem B)

Let K be a quadratic field which is not an imaginary quadratic field of class number 1. Then lsogPrimeDeg(K) is finite.

Henceforth, when we say an *isogeny-finite* K, we will mean K as above.



Fumiyuki Momose

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	0000●0000000	000000	O	0000	000
Drilling i	nto Momose's p	roof I - Finit	ely many	ϵ s	



LEMMA 1. Assume that k is a Galois extension of \mathbf{Q} and that the rational prime p is unramified in k. Then for a fixed prime p of k lying over p, we have integers a_{σ} , $0 \leq a_{\sigma} \leq 12$, for $\sigma \in \text{Gal}(k/\mathbf{Q})$ such that

 $\lambda^{12}((\alpha)) \equiv \alpha^{\varepsilon} \pmod{\mathfrak{p}}$

for $\varepsilon = \Sigma_{\sigma} a_{\sigma} \sigma$ and $\alpha \in k^{\times}$ prime to p.



LEMMA 1. Assume that k is a Galois extension of \mathbf{Q} and that the rational prime p is unramified in k. Then for a fixed prime p of k lying over p, we have integers a_{σ} , $0 \leq a_{\sigma} \leq 12$, for $\sigma \in \text{Gal}(k/\mathbf{Q})$ such that

 $\lambda^{12}((\alpha)) \equiv \alpha^{\varepsilon} \pmod{\mathfrak{p}}$

for $\varepsilon = \Sigma_{\sigma} a_{\sigma} \sigma$ and $\alpha \in k^{\times}$ prime to p.

For quadratic K, we can identify ϵ as a pair $(a, b) := a + b\sigma$, for σ the non-trivial Galois element.



LEMMA 1. Assume that k is a Galois extension of \mathbf{Q} and that the rational prime p is unramified in k. Then for a fixed prime \mathfrak{p} of k lying over p, we have integers $a_{\sigma}, 0 \leq a_{\sigma} \leq 12$, for $\sigma \in \text{Gal}(k/\mathbf{Q})$ such that

 $\lambda^{12}((\alpha)) \equiv \alpha^{\varepsilon} \pmod{\mathfrak{p}}$

for $\varepsilon = \Sigma_{\sigma} a_{\sigma} \sigma$ and $\alpha \in k^{\times}$ prime to p.

For quadratic K, we can identify ϵ as a pair $(a, b) := a + b\sigma$, for σ the non-trivial Galois element.

REMARK 1. The integers $a_{\mathfrak{p}}$'s take the values 0, 12; 4, 8 (only if the modular invariant $j(E) \equiv 0 \pmod{\mathfrak{p}}$ and $p \equiv 2 \pmod{3}$; 6 (only if $j(E) \equiv 1728 \pmod{\mathfrak{p}}$ and $p \equiv 3 \pmod{4}$ (cf. [Ma1], Chap. 3; [Ma2]).

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	00000000000	000000	0	0000	000

<pre># The PreTypeOneTwo epsilons, EPSILONS_PRE_TYPE_1_2 = {</pre>	with	their	types
(0,12): 'quadratic',			
(12,0): 'quadratic',			
(0,4): 'quartic',			
(0,8): 'quartic',			
(4,0): 'quartic',			
(4,4): 'quartic',			
(4,8): 'quartic',			
(4,12): 'quartic',			
(8,0): 'quartic', (8,4): 'quartic',			
(8,8): 'quartic',			
(8,12): 'quartic',			
(12,4): 'quartic',			
(12,8): 'quartic',			
(0,6) : 'sextic',			
(6,0) : 'sextic',			
(6,12) : 'sextic',			
(12,6) : 'sextic'			
}			





Note that the three pairs (0,0), (12,12), (6,6) are not declared here, because ...

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
Momoco's	Classification	of leaganias	into throo	0000	000

Let K be a number field. Then there exists an effective constant $C_0 = C_0(K)$ such that for any prime $p > C_0$, and for any elliptic curve admitting a K-rational p-isogeny, the isogeny character λ falls into one of the following three types:

Type 1.
$$\lambda^{12}$$
 or $(\lambda \theta_p^{-1})^{12}$ is unramified.

Type 2.
$$\lambda^{12} = \theta_p^6$$
 and $p \equiv 3 \pmod{4}$.

Type 3. K contains the Hilbert class field H_L of an imaginary quadratic field. The rational prime p splits in L:

$$p\mathcal{O}_L = \mathfrak{p}\overline{\mathfrak{p}}.$$

For any prime q of K prime to \mathfrak{p} ,

$$\lambda^{12}(\mathsf{Frob}_{\mathfrak{q}}) = \alpha^{12} \pmod{\mathfrak{p}}$$

for any $\alpha \in K^{\times}$ with $\alpha \mathcal{O}_L = \operatorname{Nm}_{k/L}(\mathfrak{q})$.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
	00000000000				
Momose's	Classification	of Isogenies	into three	types	

Let K be a number field. Then there exists an effective constant $C_0 = C_0(K)$ such that for any prime $p > C_0$, and for any elliptic curve admitting a K-rational p-isogeny, the isogeny character λ falls into one of the following three types:

Type 1. λ^{12} or $(\lambda \theta_p^{-1})^{12}$ is unramified. $\leftarrow \epsilon = (0,0)$ or (12,12)

Type 2. $\lambda^{12} = \theta_p^6$ and $p \equiv 3 \pmod{4}$. $\leftarrow \epsilon = (6, 6)$

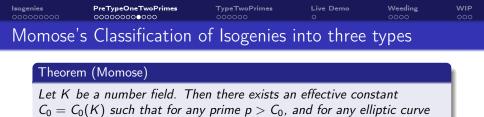
Type 3. K contains the Hilbert class field H_L of an imaginary quadratic field. The rational prime p splits in L:

$$p\mathcal{O}_L = \mathfrak{p}\overline{\mathfrak{p}}.$$

For any prime q of K prime to \mathfrak{p} ,

$$\lambda^{12}(\mathsf{Frob}_{\mathfrak{q}}) = \alpha^{12} \pmod{\mathfrak{p}}$$

for any $\alpha \in K^{\times}$ with $\alpha \mathcal{O}_L = \operatorname{Nm}_{k/L}(\mathfrak{q})$.



admitting a K-rational p-isogeny, the isogeny character λ falls into one of the following three types:

Type 1. λ^{12} or $(\lambda \theta_p^{-1})^{12}$ is unramified. $\leftarrow \epsilon = (0,0)$ or (12,12)

Type 2. $\lambda^{12} = \theta_p^6$ and $p \equiv 3 \pmod{4}$. $\leftarrow \epsilon = (6, 6)$

Type 3. K contains the Hilbert class field H_L of an imaginary quadratic field. The rational prime p splits in L:

 $p\mathcal{O}_L = \mathfrak{p}\overline{\mathfrak{p}}.$

For any prime q of K prime to p,

 $\lambda^{12}(\mathsf{Frob}_{\mathfrak{q}}) = \alpha^{12} \pmod{\mathfrak{p}}$

for any $\alpha \in K^{\times}$ with $\alpha \mathcal{O}_L = \operatorname{Nm}_{k/L}(\mathfrak{q})$.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
	000000000000				
Momose's	Classification of	of Isogenies	into three	types	

Let K be a number field. Then there exists an effective constant $C_0 = C_0(K)$ such that for any prime $p > C_0$, and for any elliptic curve admitting a K-rational p-isogeny, the isogeny character λ falls into one of the following three types:

Type 1.
$$\lambda^{12}$$
 or $(\lambda \theta_p^{-1})^{12}$ is unramified. $\leftarrow \epsilon = (0,0)$ or $(12,12)$

Type 2.
$$\lambda^{12} = \theta_p^6$$
 and $p \equiv 3 \pmod{4}$. $\leftarrow \epsilon = (6, 6)$

To make this explicit ...

For every other ϵ , find the possible isogeny primes which have an isogeny character acting via ϵ .

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	000000000●0	000000	O	0000	000
Propos	sition (B.)				

If *E* has a *K*-rational *p*-isogeny with character acting through ϵ , then for all good^{*} primes q of *K*, *p* must divide one of the following:

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
	00000000000				

If *E* has a *K*-rational *p*-isogeny with character acting through ϵ , then for all good^{*} primes q of *K*, *p* must divide one of the following:

$$\begin{split} & \mathcal{A}(\epsilon, \mathfrak{q}) := \mathsf{Nm}_{K/\mathbb{Q}}(\alpha^{\epsilon} - 1); \\ & \mathcal{B}(\epsilon, \mathfrak{q}) := \mathsf{Nm}_{K/\mathbb{Q}}(\alpha^{\epsilon} - q^{12h_{\kappa}}); \\ & \mathcal{C}(\epsilon, \mathfrak{q}) := \mathsf{lcm}(\{\mathsf{Nm}_{K(\beta)/\mathbb{Q}}(\alpha^{\epsilon} - \beta^{12h_{\kappa}}) \mid \beta \text{ is a Frobenius root over } \mathbb{F}_{\mathfrak{q}}\}). \end{split}$$

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
	00000000000				

If *E* has a *K*-rational *p*-isogeny with character acting through ϵ , then for all good^{*} primes q of *K*, *p* must divide one of the following:

$$\begin{split} & \mathcal{A}(\epsilon, \mathfrak{q}) := \mathsf{Nm}_{K/\mathbb{Q}}(\alpha^{\epsilon} - 1); \\ & \mathcal{B}(\epsilon, \mathfrak{q}) := \mathsf{Nm}_{K/\mathbb{Q}}(\alpha^{\epsilon} - q^{12h_{\kappa}}); \\ & \mathcal{C}(\epsilon, \mathfrak{q}) := \mathsf{lcm}(\{\mathsf{Nm}_{K(\beta)/\mathbb{Q}}(\alpha^{\epsilon} - \beta^{12h_{\kappa}}) \mid \beta \text{ is a Frobenius root over } \mathbb{F}_{\mathfrak{q}}\}). \end{split}$$

(Good means split in K, and non-principal if K is imaginary.)

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
	00000000000				

If *E* has a *K*-rational *p*-isogeny with character acting through ϵ , then for all good^{*} primes q of *K*, *p* must divide one of the following:

$$\begin{split} & \mathcal{A}(\epsilon, \mathfrak{q}) := \mathsf{Nm}_{K/\mathbb{Q}}(\alpha^{\epsilon} - 1); \\ & \mathcal{B}(\epsilon, \mathfrak{q}) := \mathsf{Nm}_{K/\mathbb{Q}}(\alpha^{\epsilon} - q^{12h_{\kappa}}); \\ & \mathcal{C}(\epsilon, \mathfrak{q}) := \mathsf{lcm}(\{\mathsf{Nm}_{K(\beta)/\mathbb{Q}}(\alpha^{\epsilon} - \beta^{12h_{\kappa}}) \mid \beta \text{ is a Frobenius root over } \mathbb{F}_{\mathfrak{q}}\}). \end{split}$$

(Good means split in K, and non-principal if K is imaginary.)

$$ABC(\epsilon, \mathfrak{q}) := \mathsf{Supp}(A(\epsilon, \mathfrak{q})) \cup \mathsf{Supp}(B(\epsilon, \mathfrak{q})) \cup \mathsf{Supp}(C(\epsilon, \mathfrak{q})).$$

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
	00000000000				

If *E* has a *K*-rational *p*-isogeny with character acting through ϵ , then for all good^{*} primes q of *K*, *p* must divide one of the following:

$$\begin{split} & A(\epsilon, \mathfrak{q}) := \operatorname{Nm}_{K/\mathbb{Q}}(\alpha^{\epsilon} - 1); \\ & B(\epsilon, \mathfrak{q}) := \operatorname{Nm}_{K/\mathbb{Q}}(\alpha^{\epsilon} - q^{12h_{K}}); \\ & C(\epsilon, \mathfrak{q}) := \operatorname{lcm}(\{\operatorname{Nm}_{K(\beta)/\mathbb{Q}}(\alpha^{\epsilon} - \beta^{12h_{K}}) \mid \beta \text{ is a Frobenius root over } \mathbb{F}_{\mathfrak{q}}\}). \end{split}$$

(Good means split in K, and non-principal if K is imaginary.)

$$ABC(\epsilon, \mathfrak{q}) := \operatorname{Supp}(A(\epsilon, \mathfrak{q})) \cup \operatorname{Supp}(B(\epsilon, \mathfrak{q})) \cup \operatorname{Supp}(C(\epsilon, \mathfrak{q})).$$

$$\mathsf{PreTypeOneTwoPrimes}(\mathcal{K}) := \bigcup_{\epsilon} \bigcap_{\mathfrak{q} \in \mathsf{Aux}} ABC(\epsilon, \mathfrak{q})$$

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
	00000000000				

If *E* has a *K*-rational *p*-isogeny with character acting through ϵ , then for all good^{*} primes q of *K*, *p* must divide one of the following:

$$\begin{split} & A(\epsilon, \mathfrak{q}) := \operatorname{Nm}_{K/\mathbb{Q}}(\alpha^{\epsilon} - 1); \\ & B(\epsilon, \mathfrak{q}) := \operatorname{Nm}_{K/\mathbb{Q}}(\alpha^{\epsilon} - q^{12h_{K}}); \\ & C(\epsilon, \mathfrak{q}) := \operatorname{lcm}(\{\operatorname{Nm}_{K(\beta)/\mathbb{Q}}(\alpha^{\epsilon} - \beta^{12h_{K}}) \mid \beta \text{ is a Frobenius root over } \mathbb{F}_{\mathfrak{q}}\}). \end{split}$$

(Good means split in K, and non-principal if K is imaginary.)

$$ABC(\epsilon, \mathfrak{q}) := \operatorname{Supp}(A(\epsilon, \mathfrak{q})) \cup \operatorname{Supp}(B(\epsilon, \mathfrak{q})) \cup \operatorname{Supp}(C(\epsilon, \mathfrak{q})).$$

$$\mathsf{PreTypeOneTwoPrimes}(\mathcal{K}) := \bigcup_{\epsilon} \bigcap_{\mathfrak{q} \in \mathsf{Aux}} ABC(\epsilon, \mathfrak{q})$$

for Aux a finite set of good auxiliary primes.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	00000000000	000000		0000	000
Λ • Ι	the second second	 Let a subscription 			

A quick glance at the implementation

```
def get_AB_primes(K, q, epsilons, q_class_group_order):
    output_dict_AB = {}
    alphas = (q ++ q_class_group_order).gens_reduced()
    assert len(alphas) == 1, "q^q_class_group_order not principal, which
    is very bad'
    alpha = alphas[0]
    rat q = ZZ(q,norm())
    assert rat,q.is prime(), "somehow the degree 1 prime is not prime"
    for eps in epsilons:
        alpha_to_eps = group_ring_exp(alpha,eps)
        A = (alpha_to_eps - (rat,q +* (12 * q_class_group_order))).norm()
        B = (alpha_to_eps - (rat,q +* (12 * q_class_group_order))).norm()
        output_dict_AB[eps] = lom(A,B)
```

```
for frob poly in frob polys to loop:
    if frob poly.is irreducible():
        frob poly root field = frob poly.root field('a')
       , K into KL, L into KL, = K.composite fields(frob poly root field, 'c', bo
        frob polv root field = IntegerRing()
   roots of frob = frob poly.roots(frob poly root field)
   betas = [r for r,e in roots of frob]
    for beta in betas:
        if beta in K:
            for eps in epsilons:
                N = (group ring exp(alpha, eps) - beta ** (12*g class group order)).al
                output dict C[eps] = lcm(output dict C[eps], N)
            for eps in epsilons:
                N = (K \text{ into } KL(\text{group ring } exp(alpha, eps)) - L \text{ into } KL(\text{beta} ** (12*g c))
                N = ZZ(N)
                output dict C[eps] = lcm(output dict C[eps], N)
return output dict C
```

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000	00000000000	00000			

TypeTwoPrimes

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
		00000			
C 11					
Condit	ion CC				

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	000000000000	○●○○○○	O	0000	000
Condition	n CC				

Let K be an isogeny-finite quadratic field, and E/K an elliptic curve admitting a K-rational p-isogeny, with p of Type 2.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	000000000000	○●○○○○	O	0000	000
Condition	CC				

Let K be an isogeny-finite quadratic field, and E/K an elliptic curve admitting a K-rational p-isogeny, with p of Type 2. Let q be a rational prime < p/4 such that $q^2 + q + 1 \not\equiv 0 \pmod{p}$. Then the following implication holds:

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	000000000000	0●0000	O	0000	000
Condition	CC				

Let K be an isogeny-finite quadratic field, and E/K an elliptic curve admitting a K-rational p-isogeny, with p of Type 2. Let q be a rational prime < p/4 such that $q^2 + q + 1 \not\equiv 0 \pmod{p}$. Then the following implication holds:

if q splits or ramifies in K, then q does not split in $\mathbb{Q}(\sqrt{-p})$.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000	000000000000	○●○○○○	O	0000	000
Condition	CC				

Let K be an isogeny-finite quadratic field, and E/K an elliptic curve admitting a K-rational p-isogeny, with p of Type 2. Let q be a rational prime < p/4 such that $q^2 + q + 1 \not\equiv 0 \pmod{p}$. Then the following implication holds:

if q splits or ramifies in K, then q does not split in $\mathbb{Q}(\sqrt{-p})$.



Dorian Goldfeld

Appendix. An Analogue of the Class Number One Problem

By D. Goldfeld

1. Let K be an algebraic number field of finite degree over \mathbf{Q} with discriminant k, and let S be a finite set of rational primes. Define $\mathcal{N}(K, S)$ to be the set of rational integers N satisfying the conditions:

-N is a discriminant of a quadratic field and for all primes $l \notin S$, l < |N|/4, if l splits completely in K, then l doesn't split in $\mathbb{Q}(\sqrt{-N})$.

In the case that K is equal to Q or a quadratic field, we shall show that $\mathcal{N}(K, S)$ is a finite set. The method of proof, however, is ineffective and all that can be deduced is

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000		00●000	O	0000	000
Gener	ralises Mazur's (

Claim. If the above case occurs then for all odd primes p < N/4 we have $\binom{p}{N} = -1$.



Barry C. Mazur receives National Medal of Science from US President Barack H. Obama

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
		00000			

... Generalises Mazur's Claim



Barry C. Mazur receives National Medal of Science from US President Barack H. Obama Claim. If the above case occurs then for all odd primes p < N/4 we have $\binom{p}{N} = -1$.

To conclude our theorem, we shall now prove that the above *claim* implies that $\mathbf{Q}(\sqrt{-N})$ has class number 1 and hence (by Baker-Stark-Heegner [3, 37, 38]) we have N = 11, 19, 43, 67, or 163 (ignoring the genus 0 cases).

Since $N \equiv -1 \mod 4$, quadratic reciprocity applied to (7.1) implies that for $2 , p remains prime in <math>\mathbb{Q}(\sqrt{-N})$.

Thus all ideals *I* of odd norm < N/4 are principal in the ring of integers of $\mathbf{Q}(\sqrt{-N})$. To be sure, if we had the stronger assertion that *all* ideals of norm < N/4 were principal, then $\mathbf{Q}(\sqrt{-N})$ would have class number 1 by Minkowski's theorem: the absolute value of the discriminant of $\mathbf{Q}(\sqrt{-N})$ is *N*; the Minkowski's constant is $2/\pi$; and $2/\pi \cdot \sqrt{N} < N/4$ for $N \ge 11$. We shall prove this stronger assertion. If 2 does not split in $\mathbf{Q}(\sqrt{-N})$, there is nothing to prove. Suppose, then, that 2 does split, in which case $N \equiv -1$ or 7 mod 16. We must show that one (and hence both) of the primes of norm 2 are principal. If $N \equiv -1 \mod 16$, consider the element $\alpha = (3 + \sqrt{-N})/2$. One sees that the norm of α is twice an odd number; hence $(\alpha) = p \cdot I$ where p is one of the primes of norm 2, and *I* is an "odd" ideal, with norm (9 + N)/8. Since $N \ge 11$, the norm of *I* is less than N/4, and therefore *I* is principal. Consequently so is p. If $N \equiv 7 \mod 16$, take the element $\alpha = (1 + \sqrt{-N})/2$, and repeat the above argument.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
		000000			

Determining the Type 2 primes is harder for general K.

Isogenies 000000000	PreTypeOneTwoPrimes	TypeTwoPrimes 000€00	Live Demo O	Weeding 0000	WIP 000
Determ	ining the Type 2 prim	nes is harder for g	general <i>K</i> .		
Larson	and Vaintrob obtained	d a <i>bound</i> on the	ese primes invo	olving	

"effectively computable absolute constants".

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
		000000			

Determining the Type 2 primes is harder for general *K*. Larson and Vaintrob obtained a *bound* on these primes involving "effectively computable absolute constants".



Eric Larson



Dmitry Vaintrob

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
		000000			

Determining the Type 2 primes is harder for general *K*. Larson and Vaintrob obtained a *bound* on these primes involving "effectively computable absolute constants".



Eric Larson

Dmitry Vaintrob

Theorem 7.9. Under GRH, there are effectively computable absolute constants c_2 , c_3 , and c_4 such that we can take in Theorems 6.4 and 5.16

$$\prod_{\ell \in S_K} \ell \leq \exp\left(c_2^{n_K} \cdot \left(R_K \cdot n_K^{r_K} + h_K^2 \cdot \left(\log \Delta_K\right)^2\right)\right)$$

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
		000000			

Determining the Type 2 primes is harder for general *K*. Larson and Vaintrob obtained a *bound* on these primes involving "effectively computable absolute constants".



Eric Larson

Dmitry Vaintrob

Theorem 7.9. Under GRH, there are effectively computable absolute constants c_2 , c_3 , and c_4 such that we can take in Theorems 6.4 and 5.16

$$\prod_{\ell \in S_K} \ell \leq \exp\left(c_2^{n_K} \cdot \left(R_K \cdot n_K^{r_K} + h_K^2 \cdot \left(\log \Delta_K\right)^2\right)\right)$$

Question

Can we remove the "effectively computable absolute constants"?

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000	00000000000	000000	0	0000	000

Proposition (B.)

Assume GRH. Let K be an isogeny-finite quadratic field, and E/K an elliptic curve possessing a K-rational p-isogeny, for p a Type 2 prime. Then p satisfies

$$p \leq (16 \log p + 16 \log(12\Delta_{\mathcal{K}}) + 26)^4.$$

In particular, there are only finitely many primes p as above.

Isogenies PreType	eOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000 00000	000000	000000	0	0000	000

Proposition (B.)

Assume GRH. Let K be an isogeny-finite quadratic field, and E/K an elliptic curve possessing a K-rational p-isogeny, for p a Type 2 prime. Then p satisfies

$$p\leq (16\log p+16\log(12\Delta_{\mathcal{K}})+26)^4.$$

In particular, there are only finitely many primes p as above.

Strategy

Check all primes up to this bound for whether they satisfy condition CC or not.

000 0000 0000000 000000 000000 00000000	Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
	000000000	00000000000	000000	0	0000	000



def get_type_2_primes(K, bound=None): """Compute a list containing the type 2 primes"""

First get the bound if bound is None: bound = get_type_2_bound(K) print("type_2_bound = {}".format(bound))

We need to include all primes up to 25 # see Larson/Vaintrob's proof of Theorem 6.4 output = set(prime_range(25))



blockSize=100000; export(blockSize)

checktypetwo(pBeg) =

. export(checktypetwo)

howMany=floor(typetwobound/blockSize); parapply(checktypetwo,[0..howMany]);

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000	00000000000	000000	•	0000	000

Live Demo

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000	00000000000	000000	0	0000	000

$IsogPrimeDeg(\mathbb{Q}(\sqrt{5}))$

	sogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
¢	00000000	0000000000	000000	0	0000	000

 $\{23, 29, 31, 41, 47, 53, 59, 61, 71, 73, 79\}$.

Isogenies 000000000	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo O	Weeding 0●00	WIP 000

$\{23, 29, 31, 41, 47, 53, 59, 61, 71, 73, 79\}$.

i.e. for each p in this set, determine whether the modular curve $X_0(p)$ admits any non-cuspidal $\mathbb{Q}(\sqrt{5})$ -rational points.

Isogenies 000000000	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo O	Weeding ○●○○	WIP 000

 $\{23, 29, 31, 41, 47, 53, 59, 61, 71, 73, 79\}$.

i.e. for each p in this set, determine whether the modular curve $X_0(p)$ admits any non-cuspidal $\mathbb{Q}(\sqrt{5})$ -rational points.

 $X_0(23)$ does admit such points:

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
				0000	

N:=23:

```
\{23, 29, 31, 41, 47, 53, 59, 61, 71, 73, 79\}.
```

i.e. for each p in this set, determine whether the modular curve $X_0(p)$ admits any non-cuspidal $\mathbb{Q}(\sqrt{5})$ -rational points.

 $X_0(23)$ does admit such points:

X := SmallModularCurve(N,K); > RationalPoints(X : Bound:=10);



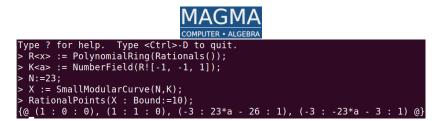
{@ (1 : 0 : 0), (1 : 1 : 0), (-3 : 23*a - 26 : 1), (-3 : -23*a - 3 :

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
				0000	

```
\{23, 29, 31, 41, 47, 53, 59, 61, 71, 73, 79\}.
```

i.e. for each p in this set, determine whether the modular curve $X_0(p)$ admits any non-cuspidal $\mathbb{Q}(\sqrt{5})$ -rational points.

 $X_0(23)$ does admit such points:



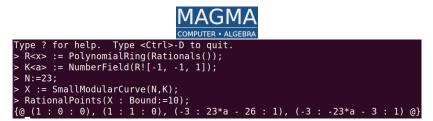
This method also works for 47.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
				0000	

```
\{23, 29, 31, 41, 47, 53, 59, 61, 71, 73, 79\}.
```

i.e. for each p in this set, determine whether the modular curve $X_0(p)$ admits any non-cuspidal $\mathbb{Q}(\sqrt{5})$ -rational points.

 $X_0(23)$ does admit such points:



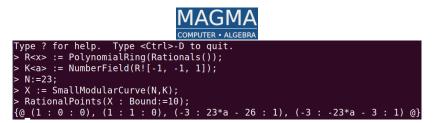
This method also works for 47. But it doesn't work for the other cases.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
				0000	

```
\{23, 29, 31, 41, 47, 53, 59, 61, 71, 73, 79\}.
```

i.e. for each p in this set, determine whether the modular curve $X_0(p)$ admits any non-cuspidal $\mathbb{Q}(\sqrt{5})$ -rational points.

 $X_0(23)$ does admit such points:



This method also works for 47. But it doesn't work for the other cases.

All of these primes are such that $genus(X_0(p)) \leq 5$.

 Isogenies
 PreTypeOneTwoPrimes
 TypeTwoPrimes
 Live Demo
 Weeding
 WIP

 000000000
 000000000
 000000
 00000
 00000
 00000

Quadratic Points of Low-genus modular curves



Peter J. Bruin

Hyperelliptic modular curves $X_0(N)$ and isogenies of elliptic curves over quadratic fields, 2015



Filip Najman



Ekin Özman



Quadratic points on modular curves with infinite Mordell-Weil group, 2021

Quadratic points on modular curves, 2019



Samir Siksek

Josha Box

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000		000000	O	000●	000
Summary					

Using their results, we can rule out the other values to conclude that

 $\mathsf{IsogPrimeDeg}(\mathbb{Q}(\sqrt{5})) = \mathsf{IsogPrimeDeg}(\mathbb{Q}) \cup \{23, 47\}.$

lsogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000		000000	O	000●	000
Summary					

Using their results, we can rule out the other values to conclude that

 $\mathsf{IsogPrimeDeg}(\mathbb{Q}(\sqrt{5})) = \mathsf{IsogPrimeDeg}(\mathbb{Q}) \cup \{23, 47\}.$

One similarly shows

$$\begin{split} & \mathsf{IsogPrimeDeg}(\mathbb{Q}(\sqrt{7})) = \mathsf{IsogPrimeDeg}(\mathbb{Q}).\\ & \mathsf{IsogPrimeDeg}(\mathbb{Q}(\sqrt{-10})) = \mathsf{IsogPrimeDeg}(\mathbb{Q}). \end{split}$$

Isogenies 000000000	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo O	Weeding 000●	WIP 000
Summary					

Using their results, we can rule out the other values to conclude that

 $\mathsf{IsogPrimeDeg}(\mathbb{Q}(\sqrt{5})) = \mathsf{IsogPrimeDeg}(\mathbb{Q}) \cup \{23, 47\}.$

One similarly shows

$$\begin{split} & \mathsf{IsogPrimeDeg}(\mathbb{Q}(\sqrt{7})) = \mathsf{IsogPrimeDeg}(\mathbb{Q}).\\ & \mathsf{IsogPrimeDeg}(\mathbb{Q}(\sqrt{-10})) = \mathsf{IsogPrimeDeg}(\mathbb{Q}). \end{split}$$

See the final section of the paper for the details.

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000	00000000000				000

Further Avenues

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
					000

Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
					000

Working on determining IsogCyclicDeg(K) for certain Ks with Oana Adascalitei in Boston, USA, and Filip Najman in Zagreb, Croatia.



Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
					000

Working on determining IsogCyclicDeg(K) for certain Ks with Oana Adascalitei in Boston, USA, and Filip Najman in Zagreb, Croatia.

Working on extending the methods to cubic and higher degree number fields with Maarten Derickx in Den Haag, The Netherlands.





Isogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
00000000	00000000000	000000	0	0000	000

Working on determining IsogCyclicDeg(K) for certain Ks with Oana Adascalitei in Boston, USA, and Filip Najman in Zagreb, Croatia.

Working on extending the methods to cubic and higher degree number fields with Maarten Derickx in Den Haag, The Netherlands.

I'll be giving a live demo of our latest algorithm on a cubic field at my VaNTAGe seminar talk on **June 29th**:

https://sites.google.com/
view/vantageseminar







lsogenies	PreTypeOneTwoPrimes	TypeTwoPrimes	Live Demo	Weeding	WIP
000000000		000000	O	0000	00●
Thanks f	or listening!				

Image	Copyright Holder	License	Image	Copyright Holder	License
	George M. Bergman, via MFO	CC BY-SA 2.0		Math. Dept., Chuo Univer- sity, Tokyo, Japan	Fair use
2	George M. Bergman, via MFO	CC BY-SA 2.0		Maarten Derickx	Fair use
	Ada Goldfeld	Written permission obtained		Getty images/Jewel Samad	Fair use (embed)
0	Univ. of Washington	Fair use	S	Dmitry Vaintrob	Fair use
				Matematički kolokvij u Osi- jeku	Fair use
<u>I</u>	Peter J. Bruin	Fair use	()	Ekin Özman	Fair use
3	Univ. of Warwick	Fair use		Univ. of Warwick	Fair use
		M		gsinstitut Oberwolfach	ran use
	MFO =	= wathematische	s rorschung	gsinstitut Operwolfach	