The elliptic-curve zoo

D. J. Bernstein

University of Illinois at Chicago
EC point counting


Input: prime power $q; \ a, b \in \mathbb{F}_q$ such that $6(4a^3 + 27b^2) \neq 0$.

Output: $\#\{(x, y) \in \mathbb{F}_q \times \mathbb{F}_q : y^2 = x^3 + ax + b\} + 1$; i.e., $\#E(\mathbb{F}_q)$ where $E$ is the elliptic curve $y^2 = x^3 + ax + b$.

Time: $(\log q)^O(1)$. 
Elliptic curves everywhere


1984 (published 1985) Miller, and independently

Bosma, Goldwasser–Kilian, Chudnovsky–Chudnovsky, Atkin: elliptic-curve primality proving.

These applications are different but share many optimizations.
Representing curve points

Crypto 1985, Miller, “Use of elliptic curves in cryptography”:

Given $n \in \mathbb{Z}$, $P \in E(\mathbb{F}_q)$, division-polynomial recurrence computes $nP \in E(\mathbb{F}_q)$ “in $26 \log_2 n$ multiplications”; but can do better!

“It appears to be best to represent the points on the curve in the following form:
Each point is represented by the triple $(x, y, z)$ which corresponds to the point $(x/z^2, y/z^3)$.”
1986 Chudnovsky–Chudnovsky, “Sequences of numbers generated by addition in formal groups and new primality and factorization tests”:

“The crucial problem becomes the choice of the model of an algebraic group variety, where computations mod $p$ are the least time consuming.”

Most important computations: ADD is $P, Q \mapsto P + Q$. DBL is $P \mapsto 2P$. 
“It is preferable to use models of elliptic curves lying in low-dimensional spaces, for otherwise the number of coordinates and operations is increasing. This limits us . . . to 4 basic models of elliptic curves.”

Short Weierstrass:

\[ y^2 = x^3 + ax + b. \]

Jacobi intersection:

\[ s^2 + c^2 = 1, \ as^2 + d^2 = 1. \]

Jacobi quartic: \( y^2 = x^4 + 2ax^2 + 1. \)

Hessian: \( x^3 + y^3 + 1 = 3dxy. \)
Some Newton polygons

- Short Weierstrass
- Montgomery
- Jacobi quartic
- Hessian
- Edwards
- Binary Edwards
Optimizing Jacobian coordinates

For “traditional” \((X/Z^2, Y/Z^3)\) on \(y^2 = x^3 + ax + b\):
1986 Chudnovsky–Chudnovsky state explicit formulas using
10\(M\) for DBL; 16\(M\) for ADD.

Consequence:
\[
\approx \left(10 \log n + 16 \frac{\log n}{\log \log n}\right) M
\]
to compute \(n, P \mapsto nP\) using “sliding windows” method of scalar multiplication.

Notation: \(\log = \log_2\);
\(M\) is cost of multiplying in \(F_q\).
Squaring is faster than \( M \).

Here are the DBL formulas:

\[
S = 4X_1 \cdot Y_1^2;
\]
\[
M = 3X_1^2 + aZ_1^4;
\]
\[
T = M^2 - 2S;
\]
\[
X_3 = T;
\]
\[
Y_3 = M \cdot (S - T) - 8Y_1^4;
\]
\[
Z_3 = 2Y_1 \cdot Z_1.
\]

Total cost \( 3M + 6S + 1D \) where

\( S \) is the cost of squaring in \( F_q \),

\( D \) is the cost of multiplying by \( a \).

The squarings produce

\( X_1^2, Y_1^2, Y_1^4, Z_1^2, Z_1^4, M^2 \).
Most ECC standards choose curves that make formulas faster.

Curve-choice advice from 1986 Chudnovsky–Chudnovsky:

Can eliminate the 1\(\text{D}\) by choosing curve with \(a = 1\).

But “it is even smarter” to choose curve with \(a = -3\).

If \(a = -3\) then \(M = 3(X_1^2 - Z_1^4) = 3(X_1 - Z_1^2) \cdot (X_1 + Z_1^2)\).

Replace 2\(\text{S}\) with 1\(\text{M}\).

Now DBL costs 4\(\text{M}\) + 4\(\text{S}\).
2001 Bernstein:
3M + 5S for DBL.
11M + 5S for ADD.

How? Easy S – M tradeoff:
instead of computing $2Y_1 \cdot Z_1$, compute $(Y_1 + Z_1)^2 - Y_1^2 - Z_1^2$.
DBL formulas were already computing $Y_1^2$ and $Z_1^2$.

Same idea for the ADD formulas, but have to scale X, Y, Z
to eliminate divisions by 2.
ADD for $y^2 = x^3 + ax + b$:
$U_1 = X_1 Z_2^2$, $U_2 = X_2 Z_1^2$,
$S_1 = Y_1 Z_2^3$, $S_2 = Y_2 Z_1^3$,
many more computations.

1986 Chudnovsky–Chudnovsky:
“We suggest to write
addition formulas involving
$(X, Y, Z, Z^2, Z^3)$.”

Disadvantages:
Allocate space for $Z^2$, $Z^3$.
Pay $1S + 1M$ in ADD and in DBL.

Advantages:
Save $2S + 2M$ at start of ADD.
Save $1S$ at start of DBL.
1998 Cohen–Miyaji–Ono:
Store point as $(X : Y : Z)$.
If point is input to ADD, also cache $Z^2$ and $Z^3$.
No cost, aside from space.
If point is input to another ADD, reuse $Z^2$, $Z^3$. Save $1S + 1M$!

Best Jacobian speeds today, including $S - M$ tradeoffs:
$3M + 5S$ for DBL if $a = -3$.
$11M + 5S$ for ADD.
$10M + 4S$ for reADD.
$7M + 4S$ for mADD (i.e. $Z_2 = 1$).
Compare to speeds for Edwards curves $x^2 + y^2 = 1 + dx^2y^2$ in projective coordinates (2007 Bernstein–Lange):

$3M + 4S$ for DBL.

$10M + 1S + 1D$ for ADD.

$9M + 1S + 1D$ for mADD.

Inverted Edwards coordinates (2007 Bernstein–Lange):

$3M + 4S + 1D$ for DBL.

$9M + 1S + 1D$ for ADD.

$8M + 1S + 1D$ for mADD.
\[ y^2 = x^3 - 0.4x + 0.7 \]
The Weierstrass-turtle: old, trusted and slow. Warning: (picture) incomplete!

(Thanks to Tanja Lange for the pictures.)
\[ x^2 + y^2 = 1 - 300x^2y^2 \]
The Edwards starfish: new, fast and complete!
Start!
Weierstrass sets off, Edwards left behind sleeping
Weierstrass has made some progress - finally Edwards wakes up.
Exciting progress: Edwards about to overtake!!
And the winner is: Edwards!
Speed-oriented Jacobian standards

2000 IEEE “Std 1363” uses Weierstrass curves in Jacobian coordinates to “provide the fastest arithmetic on elliptic curves.” Also specifies a method of choosing curves $y^2 = x^3 - 3x + b$.

2000 NIST “FIPS 186–2” standardizes five such curves.

2005 NSA “Suite B” recommends two of the NIST curves as the only public-key cryptosystems for U.S. government use.
Projective for Weierstrass

1986 Chudnovsky–Chudnovsky: Speed up ADD by switching from $(X/Z^2, Y/Z^3)$ to $(X/Z, Y/Z)$. $7M + 3S$ for DBL if $a = -3$. $12M + 2S$ for ADD. $12M + 2S$ for reADD.

Option has been mostly ignored: DBL dominates in ECDH etc. But ADD dominates in some applications: e.g., batch signature verification.
Montgomery curves

1987 Montgomery:

Use \( by^2 = x^3 + ax^2 + x \).

Choose small \((a + 2)/4\).

\[
2(x_2, y_2) = (x_4, y_4)
\]

\[
\Rightarrow x_4 = \frac{(x_2^2 - 1)^2}{4x_2(x_2^2 + ax_2 + 1)}.
\]

\[
(x_3, y_3) - (x_2, y_2) = (x_1, y_1),
\]

\[
(x_3, y_3) + (x_2, y_2) = (x_5, y_5)
\]

\[
\Rightarrow x_5 = \frac{(x_2x_3 - 1)^2}{x_1(x_2 - x_3)^2}.
\]
Represent \((x, y)\) as \((X:Z)\) satisfying \(x = X/Z\).

\[
B = (X_2 + Z_2)^2,  \\
C = (X_2 - Z_2)^2,  \\
D = B - C, \quad X_4 = B \cdot C,  \\
Z_4 = D \cdot (C + D(a + 2)/4) \Rightarrow  \\
2(X_2:Z_2) = (X_4:Z_4).
\]

\[
(X_3:Z_3) - (X_2:Z_2) = (X_1:Z_1),  \\
E = (X_3 - Z_3) \cdot (X_2 + Z_2),  \\
F = (X_3 + Z_3) \cdot (X_2 - Z_2),  \\
X_5 = Z_1 \cdot (E + F)^2,  \\
Z_5 = X_1 \cdot (E - F)^2 \Rightarrow  \\
(X_3:Z_3) + (X_2:Z_2) = (X_5:Z_5).
\]
This representation does not allow ADD but it allows DADD, “differential addition”:
\( Q, R, Q - R \mapsto Q + R \).

e.g. 2\( P \), \( P \), \( P \) \( \mapsto \) 3\( P \).
e.g. 3\( P \), 2\( P \), \( P \) \( \mapsto \) 5\( P \).
e.g. 6\( P \), 5\( P \), \( P \) \( \mapsto \) 11\( P \).

2\( M \) + 2\( S \) + 1\( D \) for DBL.
4\( M \) + 2\( S \) for DADD.
Save 1\( M \) if \( Z_1 = 1 \).

Easily compute \( n(X_1 : Z_1) \) using \( \approx \lg n \) DBL, \( \approx \lg n \) DADD.
Almost as fast as Edwards \( nP \).
Relatively slow for \( mP + nQ \) etc.
Doubling-oriented curves

2006 Doche–Icart–Kohel:

Use $y^2 = x^3 + ax^2 + 16ax$. Choose small $a$.

Use $(X : Y : Z : Z^2)$ to represent $(X/Z, Y/Z^2)$.  

$3M + 4S + 2D$ for DBL. How? Factor DBL as $\hat{\varphi}(\varphi)$ where $\varphi$ is a 2-isogeny.

2007 Bernstein–Lange:

$2M + 5S + 2D$ for DBL on the same curves.
$12\text{M} + 5\text{S} + 1\text{D}$ for ADD.

Slower ADD than other systems, typically outweighing benefit of the very fast DBL.

But isogenies are useful.
Example, 2005 Gaudry: fast DBL+DADD on Jacobians of genus-2 hyperelliptic curves, using similar factorization.

Tricky but potentially helpful: tripling-oriented curves (see 2006 Doche–Icart–Kohel), double-base chains, . . .
Hessian curves

Credited to Sylvester by 1986 Chudnovsky–Chudnovsky:

\((X : Y : Z)\) represent \((X/Z, Y/Z)\) on \(x^3 + y^3 + 1 = 3dxy\).

**12M** for ADD:

\[
X_3 = Y_1 X_2 \cdot Y_1 Z_2 - Z_1 Y_2 \cdot X_1 Y_2,
Y_3 = X_1 Z_2 \cdot X_1 Y_2 - Y_1 X_2 \cdot Z_1 X_2,
Z_3 = Z_1 Y_2 \cdot Z_1 X_2 - X_1 Z_2 \cdot Y_1 Z_2.
\]

**6M + 3S** for DBL.
2001 Joye–Quisquater:
\[2(\frac{X_1}{Y_1} : \frac{Y_1}{Z_1}) = \frac{Z_1}{X_1} : \frac{X_1}{Y_1} + \frac{Y_1}{Z_1} : \frac{Z_1}{X_1}\]
so can use ADD to double.

“Unified addition formulas,” helpful against side channels. But not strongly unified: need to permute inputs.

2008 Hisil–Wong–Carter–Dawson:
\[(\frac{X}{Y} : \frac{Z}{X^2} : \frac{X^2}{Y} : \frac{Y^2}{Z} : \frac{Z^2}{2XY} : \frac{2XY}{2XZ} : \frac{2XZ}{2YZ}).\]

6M + 6S for ADD.
3M + 6S for DBL.
\[ x^3 - y^3 + 1 = 0.3xy \]
The Hessian-ray: uniform

but

not strongly so
Jacobi intersections

1986 Chudnovsky–Chudnovsky:

(S : C : D : Z) represent

(S/Z, C/Z, D/Z) on

\[ s^2 + c^2 = 1, \quad as^2 + d^2 = 1. \]

14M + 2S + 1D for ADD.

“Tremendous advantage”
of being strongly unified.

5M + 3S for DBL.

“Perhaps (?) . . . the most
efficient duplication formulas
which do not depend on the
coefficients of an elliptic curve.”
2001 Liardet–Smart:
$13M + 2S + 1D$ for ADD.
$4M + 3S$ for DBL.

2007 Bernstein–Lange:
$3M + 4S$ for DBL.

2008 Hisil–Wong–Carter–Dawson:
$13M + 1S + 2D$ for ADD.
$2M + 5S + 1D$ for DBL.
Also $(S : C : D : Z : SC : DZ)$:
$11M + 1S + 2D$ for ADD.
$2M + 5S + 1D$ for DBL.
Jacobi quartics

\((X:Y:Z)\) represent \((X/Z, Y/Z^2)\) on \(y^2 = x^4 + 2ax^2 + 1\).

1986 Chudnovsky–Chudnovsky: \(3M + 6S + 2D\) for DBL.
Slow ADD.

2002 Billet–Joye:
New choice of neutral element.
\(10M + 3S + 1D\) for ADD, strongly unified.

2007 Bernstein–Lange: \(1M + 9S + 1D\) for DBL.
2007 Hisil–Carter–Dawson:
$2M + 6S + 2D$ for DBL.

2007 Feng–Wu:
$2M + 6S + 1D$ for DBL.
$1M + 7S + 3D$ for DBL
on curves chosen with $a^2 + c^2 = 1$.

More speedups: 2007 Duquesne,
2007 Hisil–Carter–Dawson,
2008 Hisil–Wong–Carter–Dawson
use $(X : Y : Z : X^2 : Z^2)$
Competitive with Edwards!
\[ x^2 = y^4 - 1.9y^2 + 1 \]
The Jacobi-quartic squid: can be extended to XXYZZR giant squid.
For more information

Explicit-Formulas Database, joint work with Tanja Lange: hyperelliptic.org/EFD

EFD has 302 computer-verified formulas and operation counts for ADD, DBL, etc. in 20 representations on 8 shapes of elliptic curves.

Not yet handled by computer: generality of curve shapes (e.g., Hessian order ∈ 3Z); complete addition algorithms (e.g., checking for ∞).
Can do similar survey for elliptic curves over fields of characteristic 2.

News: EFD now includes characteristic-2 formulas!

Currently 102 computer-verified formulas and operation counts for ADD, DBL, etc. in 16 representations on 2 shapes (binary Edwards and short Weierstrass) of ordinary binary elliptic curves.