The rest of the zoo

D. J. Bernstein
University of Illinois at Chicago

EC point counting

1983 (published 1985) Schoof: Algorithm to count points on elliptic curves over finite fields.

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

Output: $\#\{(x, y) \in \mathbf{F}_q \times \mathbf{F}_q :$ $y^2 = x^3 + ax + b\} + 1 = \#E(\mathbf{F}_q)$ where $E : y^2 = x^3 + ax + b$.

Time: $(\log q)^{O(1)}$.

Elliptic curves everywhere

1984 (published 1987) Lenstra: ECM, the elliptic-curve method of factoring integers.

1984 (published 1985) Miller, and independently
1984 (published 1987) Koblitz: ECC, elliptic-curve cryptography.

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 $n \in \mathbf{Z}$, $P \in E(\mathbf{F}_q)$, division-polynomial recurrence computes $nP \in E(\mathbf{F}_q)$ "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."

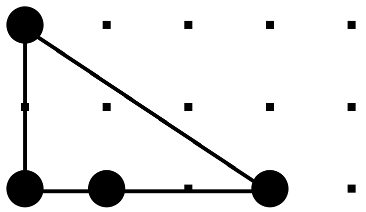
Short Weierstrass:
$y^2 = x^3 + ax + b$.

Jacobi intersection:
$s^2 + c^2 = 1$, $as^2 + d^2 = 1$.

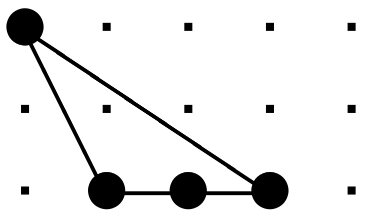
Jacobi quartic: $y^2 = x^4 + 2ax^2 + 1$.

Hessian: $x^3 + y^3 + 1 = 3dxy$.

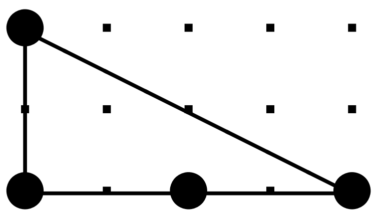
Some Newton polygons



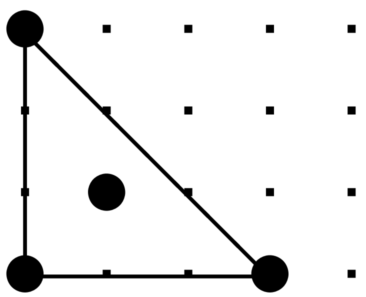
Short Weierstrass



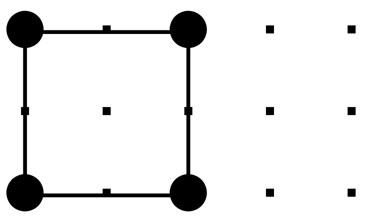
Montgomery



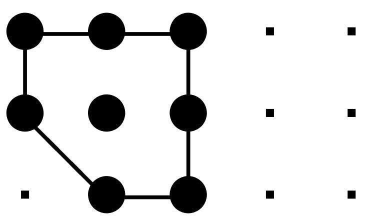
Jacobi quartic



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:

$$\approx \left(10 \lg n + 16 \frac{\lg n}{\lg \lg n}\right) \mathbf{M}$$

to compute $n, P \mapsto nP$ using "sliding windows" method of scalar multiplication.

Notation: $\lg = \log_2$;
M is cost of multiplying in $\mathbf{F}_q$.

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 $3\mathbf{M} + 6\mathbf{S} + 1\mathbf{D}$ where
S is the cost of squaring in $\mathbf{F}_q$,
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**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**S** with 1**M**.

Now DBL costs $4\mathbf{M} + 4\mathbf{S}$.

2001 Bernstein:
$3\mathbf{M} + 5\mathbf{S}$ for DBL.
$11\mathbf{M} + 5\mathbf{S}$ for ADD.

How? Easy $\mathbf{S} - \mathbf{M}$ 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 $1\mathbf{S} + 1\mathbf{M}$ in ADD and in DBL.

Advantages:
Save $2\mathbf{S} + 2\mathbf{M}$ at start of ADD.
Save $1\mathbf{S}$ 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 $1\mathbf{S} + 1\mathbf{M}$!

Best Jacobian speeds today,
including $\mathbf{S} - \mathbf{M}$ tradeoffs:
$3\mathbf{M} + 5\mathbf{S}$ for DBL if $a = -3$.
$11\mathbf{M} + 5\mathbf{S}$ for ADD.
$10\mathbf{M} + 4\mathbf{S}$ for reADD.
$7\mathbf{M} + 4\mathbf{S}$ for mADD (i.e. $Z_2 = 1$).

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 coordinates

1986 Chudnovsky–Chudnovsky:
Speed up ADD by switching from $(X/Z^2, Y/Z^3)$ to $(X/Z, Y/Z)$.
$7\mathbf{M} + 3\mathbf{S}$ for DBL if $a = -3$.
$12\mathbf{M} + 2\mathbf{S}$ for ADD.
$12\mathbf{M} + 2\mathbf{S}$ 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 $(a+2)/4$.

$2(x_2, y_2) = (x_4, y_4)$

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

$(x_3, y_3) - (x_2, y_2) = (x_1, y_1)$,
$(x_3, y_3) + (x_2, y_2) = (x_5, y_5)$

$$\Rightarrow x_5 = \frac{(x_2 x_3 - 1)^2}{x_1(x_2 - x_3)^2}.$$

Represent (x, y)
as $(X{:}Z)$ satisfying $x = X/Z$.

$B = (X_2 + Z_2)^2$,
$C = (X_2 - Z_2)^2$,
$D = B - C$, $X_4 = B \cdot C$,
$Z_4 = D \cdot (C + D(a + 2)/4) \Rightarrow$
$2(X_2{:}Z_2) = (X_4{:}Z_4)$.

$(X_3{:}Z_3) - (X_2{:}Z_2) = (X_1{:}Z_1)$,
$E = (X_3 - Z_3) \cdot (X_2 + Z_2)$,
$F = (X_3 + Z_3) \cdot (X_2 - Z_2)$,
$X_5 = Z_1 \cdot (E + F)^2$,
$Z_5 = X_1 \cdot (E - F)^2 \Rightarrow$
$(X_3{:}Z_3) + (X_2{:}Z_2) = (X_5{:}Y_5)$.

This representation
does not allow ADD but it allows
DADD, "differential addition":
$Q, R, Q - R \mapsto Q + R$.

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

$2\mathbf{M} + 2\mathbf{S} + 1\mathbf{D}$ for DBL.
$4\mathbf{M} + 2\mathbf{S}$ for DADD.
Save $1\mathbf{M}$ 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 $(X/Z, Y/Z^2)$.

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

2007 Bernstein–Lange:
$2\mathbf{M} + 5\mathbf{S}$ for DBL
on the same curves.

$12\mathbf{M} + 5\mathbf{S}$ 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 $x^3 + y^3 + 1 = 3dxy$.

12**M** for ADD:

$X_3 = Y_1X_2 \cdot Y_1Z_2 - Z_1Y_2 \cdot X_1Y_2$,
$Y_3 = X_1Z_2 \cdot X_1Y_2 - Y_1X_2 \cdot Z_1X_2$,
$Z_3 = Z_1Y_2 \cdot Z_1X_2 - X_1Z_2 \cdot Y_1Z_2$.

6**M** + 3**S** for DBL.

2001 Joye–Quisquater:

$2(X_1 : Y_1 : Z_1) =$
$(Z_1 : X_1 : Y_1) + (Y_1 : Z_1 : X_1)$
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:

$$(X : Y : Z : X^2 : Y^2 : Z^2 \\ : 2XY : 2XZ : 2YZ).$$

$6\mathbf{M} + 6\mathbf{S}$ for ADD.
$3\mathbf{M} + 6\mathbf{S}$ for DBL.

Jacobi intersections

1986 Chudnovsky–Chudnovsky:

$(S : C : D : Z)$ represent
$(S/Z, C/Z, D/Z)$ on
$s^2 + c^2 = 1$, $as^2 + d^2 = 1$.

$14\mathbf{M} + 2\mathbf{S} + 1\mathbf{D}$ for ADD.
"Tremendous advantage"
of being strongly unified.

$5\mathbf{M} + 3\mathbf{S}$ for DBL.
"Perhaps (?) . . . the most
efficient duplication formulas
which do not depend on the
coefficients of an elliptic curve."

2001 Liardet–Smart:

$13\mathbf{M} + 2\mathbf{S} + 1\mathbf{D}$ for ADD.

$4\mathbf{M} + 3\mathbf{S}$ for DBL.

2007 Bernstein–Lange:

$3\mathbf{M} + 4\mathbf{S}$ for DBL.

2008 Hisil–Wong–Carter–Dawson:

$13\mathbf{M} + 1\mathbf{S} + 2\mathbf{D}$ for ADD.

$2\mathbf{M} + 5\mathbf{S}$ for DBL.

Also $(S : C : D : Z : SC : DZ)$:

$11\mathbf{M} + 1\mathbf{S} + 2\mathbf{D}$ for ADD.

$2\mathbf{M} + 5\mathbf{S} + 1\mathbf{D}$ for DBL.

Jacobi quartics

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

1986 Chudnovsky–Chudnovsky:
$3\mathbf{M} + 6\mathbf{S} + 2\mathbf{D}$ for DBL.
Slow ADD.

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

2007 Bernstein–Lange:
$1\mathbf{M} + 9\mathbf{S} + 1\mathbf{D}$ for DBL.

2007 Hisil–Carter–Dawson:
$2\mathbf{M} + 6\mathbf{S} + 2\mathbf{D}$ for DBL.

2007 Feng–Wu:
$2\mathbf{M} + 6\mathbf{S} + 1\mathbf{D}$ for DBL.
$1\mathbf{M} + 7\mathbf{S} + 3\mathbf{D}$ 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 $(X : Y : Z : X^2 : Z^2)$
or $(X : Y : Z : X^2 : Z^2 : 2XZ)$.
Can combine with Feng–Wu.
Competitive with Edwards!

For more information

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

EFD has 296 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 $\in 3\mathbf{Z}$); complete addition algorithms (e.g., checking for ∞).