Elliptic vs. Hyper-elliptic

Part III
The Opponents

\[ g = 1 \quad \text{vs} \quad g = 2 \]

\(\ldots\) already after some transformations \(\ldots\)
Elliptic strikes back

D. J. Bernstein & T. Lange

http://cr.yp.to/newelliptic.html

− p. 3
Elliptic goes undercover


\[ x^2 + y^2 = a^2(1 + x^2y^2), \ a^5 \neq a \]

describes an elliptic curve.

- Edwards shows that generically every elliptic curve can be written in this form – over some extension field.

- Gauss (and this is basically the only mention of this form that Edwards and we could dig out) shows that

\[ x^2 + y^2 = 1 - x^2y^2 \]

is elliptic. To transform this curve we need \( \sqrt{-1} \) in the field.

D. J. Bernstein & T. Lange http://cr.yp.to/newelliptic.html – p. 4
Edwards coordinates

Introduce further parameter and relabel

\[ x^2 + y^2 = c^2(1 + dx^2y^2), \ c, d \neq 0, dc^4 \neq 1. \]

Neutral element is \((0, c)\), this is an affine point!

\[ -(x_P, y_P) = (-x_P, y_P). \]

\[ P + Q = \left( \frac{x_P y_Q + y_P x_Q}{c(1 + dx_P x_Q y_P y_Q)}, \frac{y_P y_Q - x_P x_Q}{c(1 - dx_P x_Q y_P y_Q)} \right). \]
Edwards coordinates

Introduce further parameter and relabel

\[ x^2 + y^2 = c^2(1 + dx^2y^2), \quad c, d \neq 0, dc^4 \neq 1. \]

Neutral element is \((0, c)\), this is an affine point!

\[-(x_P, y_P) = (-x_P, y_P).\]

\[ P + Q = \left( \frac{x_Py_Q + y_Px_Q}{c(1 + dx_Px_Qy_Py_Q)}, \frac{y_Py_Q - x_Px_Q}{c(1 - dx_Px_Qy_Py_Q)} \right). \]

\[ [2]P = \left( \frac{x_Py_P + y_Px_P}{c(1 + dx_Px_Py_Py_P)}, \frac{y_Py_P - x_Px_P}{c(1 - dx_Px_Py_Py_P)} \right). \]
Edwards coordinates

Introduce further parameter and relabel

\[ x^2 + y^2 = c^2(1 + dx^2y^2), \quad c, d \neq 0, dc^4 \neq 1. \]

- Neutral element is \((0, c)\), this is an affine point!
- \(- (x_P, y_P) = (-x_P, y_P)\).
- \(P + Q = \left( \frac{x_P y_Q + y_P x_Q}{c(1 + dx_P x_Q y_P y_Q)}, \frac{y_P y_Q - x_P x_Q}{c(1 - dx_P x_Q y_P y_Q)} \right)\).
- \([2]P = \left( \frac{x_P y_P + y_P x_P}{c(1 + dx_P x_P y_P y_P)}, \frac{y_P y_P - x_P x_P}{c(1 - dx_P x_P y_P y_P)} \right)\).
- Unified group operations!
Edwards coordinates

Introduce further parameter and relabel

\[ x^2 + y^2 = c^2(1 + dx^2y^2), \quad c, d \neq 0, dc^4 \neq 1. \]

- Neutral element is \((0, c)\), this is an affine point!
- \(- (x_P, y_P) = (-x_P, y_P)\).

\[ P + Q = \left( \frac{x_P y_Q + y_P x_Q}{c(1 + dx_P x_Q y_P y_Q)}, \frac{y_P y_Q - x_P x_Q}{c(1 - dx_P x_Q y_P y_Q)} \right). \]

\[
A = Z_P \cdot Z_Q; \quad B = A^2; \quad C = X_P \cdot X_Q; \quad D = Y_P \cdot Y_Q; \\
E = (X_P + Y_P) \cdot (X_Q + Y_Q) - C - D; \quad F = d \cdot C \cdot D; \\
X_{P \oplus Q} = A \cdot E \cdot (B - F); \quad Y_{P \oplus Q} = A \cdot (D - C) \cdot (B + F); \\
Z_{P \oplus Q} = c \cdot (B - F) \cdot (B + F).
\]
Edwards coordinates

Introduce further parameter and relabel

\[ x^2 + y^2 = c^2(1 + dx^2y^2), \quad c, d \neq 0, dc^4 \neq 1. \]

Neutral element is \((0, c)\), this is an affine point!

\[-(x_P, y_P) = (-x_P, y_P).\]

\[ P + Q = \left( \frac{x_Py_Q + y_Px_Q}{c(1 + dx_Px_Qy_Py_Q)}, \frac{y_Py_Q - x_Px_Q}{c(1 - dx_Px_Qy_Py_Q)} \right). \]

\[
\begin{align*}
A &= Z_P \cdot Z_Q; \quad B = A^2; \quad C = X_P \cdot X_Q; \quad D = Y_P \cdot Y_Q; \\
E &= (X_P + Y_P) \cdot (X_Q + Y_Q) - C - D; \quad F = d \cdot C \cdot D; \\
X_{P \oplus Q} &= A \cdot E \cdot (B - F); \quad Y_{P \oplus Q} = A \cdot (D - C) \cdot (B + F); \\
Z_{P \oplus Q} &= c \cdot (B - F) \cdot (B + F).
\end{align*}
\]

Needs 10M + 1S + 1C + 1D + 7A. At least one of \(c, d\) small.
# Fastest unified addition-or-doubling formulae

<table>
<thead>
<tr>
<th>System</th>
<th>Cost of unified addition-or-doubling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobian</td>
<td>$11M+6S+1D$; see Brier/Joye ’03</td>
</tr>
<tr>
<td>Jacobian if $a_4 = -1$</td>
<td>$13M+3S$; see Brier/Joye ’02</td>
</tr>
<tr>
<td>Jacobi intersection</td>
<td>$13M+2S+1D$; see Liardet/Smart ’01</td>
</tr>
<tr>
<td>Jacobi quartic</td>
<td>$10M+3S+3D$; see Billet/Joye ’01</td>
</tr>
<tr>
<td>Hessian</td>
<td>$12M$; see Joye/Quisquater ’01</td>
</tr>
<tr>
<td>Edwards $(c = 1)$</td>
<td>$10M+1S+1D$</td>
</tr>
</tbody>
</table>

- **Exactly the same formulae for doubling** (no re-arrangement like in Hessian; no if-else)

- **No exceptional cases** if $d$ is not a square. Formulae correct for all affine inputs (incl. $(0, c), -(P)$).

- **Caveat**: Edwards curves have a point of order 2, namely $(0, -c)$.

D. J. Bernstein & T. Lange  
[http://cr.yp.to/newelliptic.html](http://cr.yp.to/newelliptic.html)
But wait – there’s more!
How about non-unified doubling?

\[ [2]P = \left( \frac{x_P y_P + y_P x_P}{c(1 + d x_P x_P y_P y_P)}, \frac{y_P y_P - x_P x_P}{c(1 - d x_P x_P y_P y_P)} \right) \]

\[ = \left( \frac{2 x_P y_P}{c(1 + d (x_P y_P)^2)}, \frac{y_P^2 - x_P^2}{c(1 - d (x_P y_P)^2)} \right) \]
How about non-unified doubling?

\[ [2]P = \left( \frac{x_P y_P + y_P x_P}{c(1 + d x_P x_P y_P y_P)}, \frac{y_P y_P - x_P x_P}{c(1 - d x_P x_P y_P y_P)} \right) \]

\[ = \left( \frac{2x_P y_P}{c(1 + d(x_P y_P)^2)}, \frac{y_P^2 - x_P^2}{c(1 - d(x_P y_P)^2)} \right) \]

\[ = \left( \frac{2c x_P y_P}{c^2(1 + d(x_P y_P)^2)}, \frac{c(y_P^2 - x_P^2)}{c^2(2 - (1 + d(x_P y_P)^2))} \right) \]

Use curve equation \( x^2 + y^2 = c^2(1 + dx^2y^2) \).
How about non-unified doubling?

\[ [2]P = \left( \frac{x_P y_P + y_P x_P}{c(1 + d x_P x_P y_P y_P)}, \frac{y_P y_P - x_P x_P}{c(1 - d x_P x_P y_P y_P)} \right) \]

\[ = \left( \frac{2x_P y_P}{c(1 + d(x_P y_P)^2)}, \frac{y_P^2 - x_P^2}{c(1 - d(x_P y_P)^2)} \right) \]

\[ = \left( \frac{2c x_P y_P}{c^2(1 + d(x_P y_P)^2)}, \frac{c(y_P^2 - x_P^2)}{c^2(2 - (1 + d(x_P y_P)^2))} \right) \]

\[ = \left( \frac{2c x_P y_P}{x_P^2 + y_P^2}, \frac{c(y_P^2 - x_P^2)}{2c^2 - (x_P^2 + y_P^2)} \right) \]

D. J. Bernstein & T. Lange

http://cr.yp.to/newelliptic.html - p. 8
How about non-unified doubling?

\[
[2]P = \left( \frac{x_P y_P + y_P x_P}{c(1 + d x_P x_P y_P y_P)}, \frac{y_P y_P - x_P x_P}{c(1 - d x_P x_P y_P y_P)} \right)
\]

\[
= \left( \frac{2x_P y_P}{c(1 + d(x_P y_P)^2)}, \frac{y_P^2 - x_P^2}{c(1 - d(x_P y_P)^2)} \right)
\]

\[
= \left( \frac{2c x_P y_P}{c^2(1 + d(x_P y_P)^2)}, \frac{c(y_P^2 - x_P^2)}{c^2(2 - (1 + d(x_P y_P)^2))} \right)
\]

\[
= \left( \frac{2c x_P y_P}{x_P^2 + y_P^2}, \frac{c(y_P^2 - x_P^2)}{2c^2 - (x_P^2 + y_P^2)} \right)
\]

Inversion-free version needs 3M + 4S + 3C + 6A.

Can always choose \( c = 1 \)!
## Fastest doubling formulae

<table>
<thead>
<tr>
<th>System</th>
<th>Cost of doubling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projective</td>
<td>6M+5S+1D; HECC</td>
</tr>
<tr>
<td>Hessian</td>
<td>6M+6S; see Joye/Quisquater ’01</td>
</tr>
<tr>
<td>Jacobi quartic</td>
<td>1M+9S+3D; see Billet/Joye ’01</td>
</tr>
<tr>
<td>Jacobian</td>
<td>2M+7S+1D; HECC</td>
</tr>
<tr>
<td>Jacobian if $a_4 = -3$</td>
<td>3M+5S; see DJB ’01</td>
</tr>
<tr>
<td>Jacobi intersection</td>
<td>4M+3S+1D; see Liardet/Smart ’01</td>
</tr>
<tr>
<td>Edwards ($c = 1$)</td>
<td>3M+4S</td>
</tr>
</tbody>
</table>

- Edwards ADD takes 10M+1S+1D, mixed 9M+1S+1D.
- Edwards faster than Jacobian in DBL & ADD.
- Edwards coordinates allow to use windowing methods.
- Montgomery takes 5M+4S+1D per bit.

D. J. Bernstein & T. Lange

[http://cr.yp.to/newelliptic.html](http://cr.yp.to/newelliptic.html) – p. 9
But wait – there’s more!
http://cr.yp.to/
newelliptic.html