



# Matrix Code Equivalence





# Radboud University



**Speeding-up  
general pairings**

2













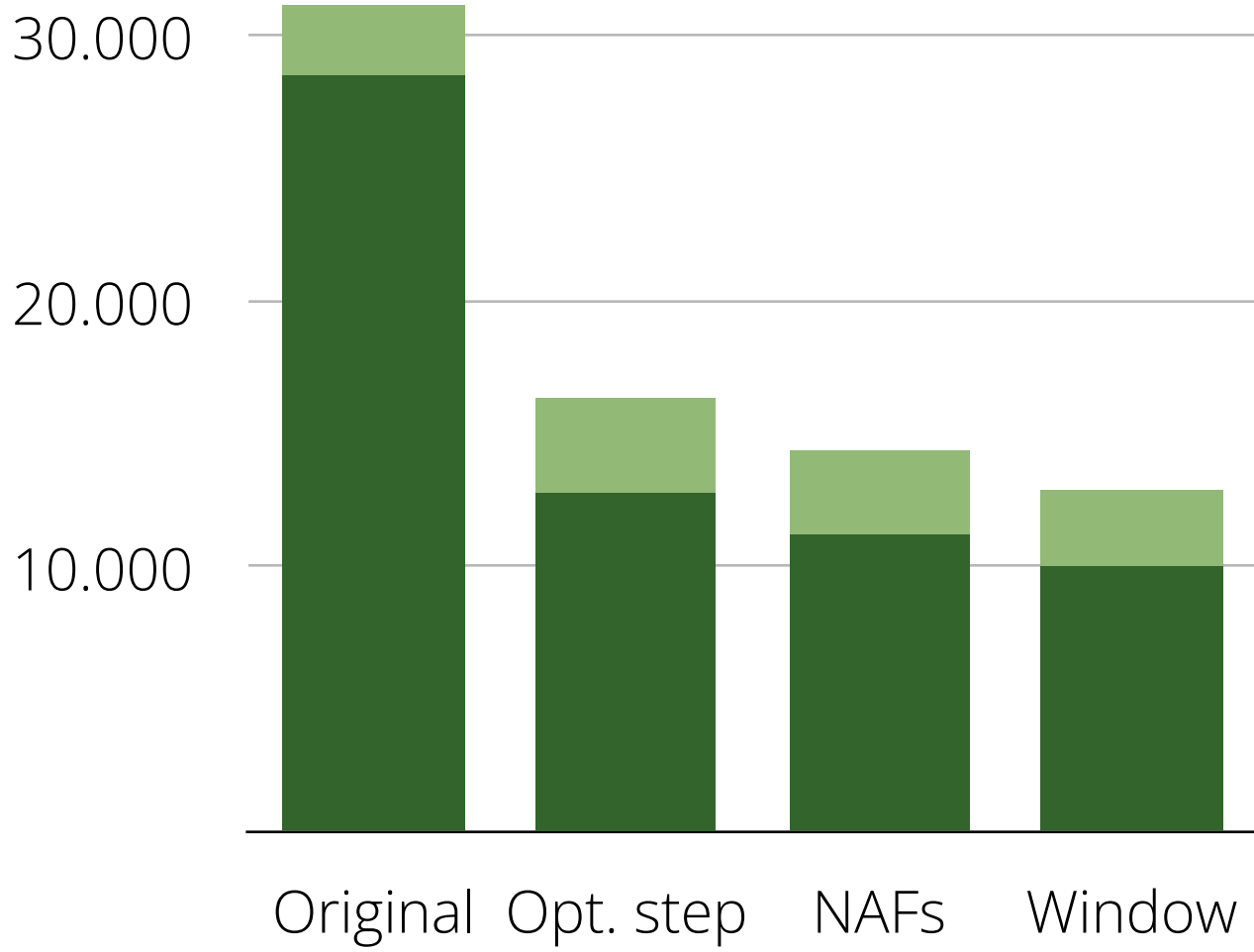












$Q$ ) with director  $(\vec{f}) = (\vec{P}) + 1_2(\vec{i}) + (-1)(\vec{P} + \vec{Q}) = 3(\vec{Q})$ , and let  $\vec{f} = p\lambda_2 + r_1$  be the tangent at  $\vec{f}$  with director  $(\vec{f}) = 21(\vec{Q}) + (-3)(\vec{Q}) = 3(\vec{Q})$ . The director of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathbb{P}_2$ .

the function  $\ell_{\text{pass}} = \vec{f}$  is  $(\ell_{\text{pass}}) = (\vec{f}) = (\vec{P}) + (\vec{Q}) + 21(\vec{Q}) + (-1)(\vec{P} + \vec{Q}) + (-3)(\vec{Q}) = 6(\vec{Q})$ . The director of  $\ell_{\text{pass}} = \vec{f}$  is  $(\ell_{\text{pass}}) = (\vec{f}) = (\vec{P}) + (\vec{Q}) + (-1)(\vec{P} + \vec{Q}) = 21(\vec{Q}) + (-3)(\vec{P} + \vec{Q}) = 21(\vec{Q}) + (-3)(\vec{Q})$ . Notice that  $\ell_{\text{pass}}$  does not intersect  $\vec{f}$  at  $\vec{Q}$ ; projecting  $\vec{f} = \frac{1}{2} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$  gives  $\frac{1}{2} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ , which does not give rise to any zero or pole at  $\mathbb{P} = 0$ . Suppose we wanted to depict the function  $\vec{f}$  on  $\mathbb{P}_2$ , and we multiplied out  $(y - \lambda_2 - r_1)(y - \lambda_2 - r_2)$ , substituted the  $y^2$  by  $x^2 + ax + b$  and wrote  $p = \frac{(\text{something})}{(\text{something})}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathbb{P}_2$ ) behave at points that are not on  $\mathbb{P}_2$ , where the substitution  $y^2 = x^2 + ax + b$  is not possible.

man in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{D}_\ell$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm, with postponed addition steps for even  $i$  and air-like pairings.

---

**Inputs:**  $Q' \in \mathcal{G}_2$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{d-1}, m_{d-2}, \dots, m_0)$ ,  $m_{d-1} = 1$

**Outputs:**  $f_{m,Q'}(P)$  representing a class in  $\mathcal{D}_\ell / \langle \mathcal{D}_\ell^p \rangle$

```

1  $R \leftarrow Q'$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2 for  $i$  from 0 to  $d - 1$  do
3   if  $(m_i = 1)$  then
4      $A_0[i] \leftarrow R$ ,  $A_1[i] \leftarrow f$ ,  $j \leftarrow j + 1$ 
5   end if
6    $f \leftarrow f^p \cdot \zeta_{m(Q',R)}(P)$ ,  $R \leftarrow [2]R$ 
7 end for
8  $R \leftarrow A_0[j]$ ,  $f \leftarrow A_1[j]$ 
9 for  $(i \leftarrow 1; j \leq d(m) - 1; i \leftarrow i + 1)$  do
10   $f \leftarrow f \cdot A_1[i] \cdot \zeta_{m(A_0[i],R)}(P)$ ,  $R \leftarrow R + A_0[i]$ 
11 end for
12 return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gauthier, Gaudsichet, and Pape [34, Algorithm 2] use a version of Algorithm 3

$$i(x) = \theta_{ij} - pr\alpha_{ij,p} \left( \frac{\theta_{ij}^2 + \delta}{\theta_{ij,p} + r\alpha_{ij,p} + \delta} (\theta_{ij} - \alpha_{ij,p}) + 1 \right).$$

We write this as  $\theta_{ij} = pr\alpha_{ij}$ . The vertical line condition is simply  $\alpha_{ij}^2 = \alpha_{ij} = \alpha_{ij,p}$ . Multiplying all three together gives  $\delta_{ij,p} = pr\alpha_{ij,p} + \theta_{ij,p}\delta_{ij,p}$  where

$$\alpha_{ij,p} = (\alpha_{ij,p}^2 - (\theta_{ij}^2 + \delta\theta_{ij} + \delta(\theta_{ij,p}^2)(\theta_{ij} - \alpha_{ij,p}))$$

and

$$\delta_{ij,p} = (\theta_{ij}^2 + \delta\theta_{ij} + \delta(\theta_{ij,p}^2) + \alpha_{ij,p}^2)(\theta_{ij} - \alpha_{ij,p}).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the values. In general case, from Lemma 2 we deduce that the line  $l$  is

$$y = pr \left( \frac{\alpha_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

and so

$$i(x) = \theta_{ij} - pr \left( \frac{\alpha_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

Writing this as  $\theta_{ij} + pr\alpha_{ij}$  we have  $\delta_{ij+1,p} = \alpha_{ij+1,p} + \theta_{ij,p}pr\delta_{ij+1,p}$  where

$$\alpha_{ij+1,p} = (\theta_{ij}^2 + \delta\theta_{ij} + \delta(\alpha_{ij,p}^2) - (\theta_{ij}^2 + \delta\theta_{ij} + \delta(\theta_{ij,p}^2)(\theta_{ij} - \alpha_{ij+1,p})).$$

and

$$\delta_{ij+1,p} = (\alpha_{ij,p} + \delta_{ij,p}\theta_{ij})(\theta_{ij} - \alpha_{ij+1,p}).$$



man in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{D}_\ell$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm with postponed addition steps for even  $k$  and ate-like pairings.

---

**Inputs:**  $\mathcal{Q}^* \in \mathcal{G}_2$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{2,1}, m_{2,2}, \dots, m_2)$ ,  $m_{2,1} = 1$

**Outputs:**  $f_{m, \mathcal{Q}^*}(P)$  representing a class in  $\mathcal{D}_\ell / \langle \mathcal{D}_\ell f \rangle$

```

1  $R \leftarrow \mathcal{Q}^*$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2 for  $i$  from 0 to  $d - 1$  do
3   if  $(m_i = 1)$  then
4      $A_R[1] \leftarrow R$ ,  $A_f[1] \leftarrow f$ ,  $j \leftarrow j + 1$ 
5   end if
6    $f \leftarrow f^2 \cdot \zeta_{\text{Frobenius}}(P)$ ,  $R \leftarrow [2]R$ 
7 end for
8  $R \leftarrow A_R[j]$ ,  $f \leftarrow A_f[j]$ 
9 for  $(i \leftarrow 1; j \leq \#(m) - 1; i \leftarrow i + 1)$  do
10   $f \leftarrow f \cdot A_f[i] \cdot \zeta_{\text{Frobenius}}(A_R[i])(P)$ ,  $R \leftarrow R + A_R[i]$ 
11 end for
12 return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gauthier, Gaudreault, and Papez [34, Algorithm 2] use a version of Algorithm 3

$$i(Q) = \theta_{ij} - pr \cdot \alpha_{i,p} \left( \frac{\theta_{ij}^2 + A}{B\theta_{i,p} + C\alpha_{i,p} + B} (\theta_{ij} - \alpha_{i,p}) + 1 \right).$$

We write this as  $\theta_{ij} = pr \cdot \alpha_{i,p}$ . The vertical line condition simply  $\alpha_{ij}^2 = \alpha_{ij} = \alpha_{i,p}$ . Multiplying all these together gives  $\beta_{i,p} = pr \cdot \alpha_{i,p} + \theta_{i,p} \alpha_{i,p}$  where

$$\alpha_{i,p} = (\alpha_{i,p}^2 - (\theta_{ij}^2 + A\theta_{ij} + B(\theta_{i,p}))(pr - \alpha_{i,p}))$$

and

$$\beta_{i,p} = (\alpha_{ij}^2 + A\alpha_{ij} + B\theta_{i,p} \alpha_{i,p} + \alpha_{i,p}^2)(pr - \alpha_{i,p}).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line  $l$  is

$$y = pr \left( \frac{\alpha_{ij} - 1}{\alpha_{i,p} - pr} (\alpha_{ij} - pr) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\alpha_{ij} - 1}{\alpha_{i,p} - pr} (\alpha_{ij} - pr) + 1 \right).$$

Writing this as  $\theta_{ij} = pr \cdot \alpha_{i,p}$  we have  $\beta_{i+1,p} = \alpha_{i+1,p} + \theta_{i,p} pr \cdot \alpha_{i+1,p}$  where

$$\alpha_{i+1,p} = (\alpha_{ij}^2 + A\alpha_{ij} + B\alpha_{i,p} \alpha_{ij} - (\alpha_{ij}^2 + A\alpha_{ij} + B\theta_{i,p} \alpha_{i,p}))(pr - \alpha_{i+1,p}),$$

and

$$\beta_{i+1,p} = (\alpha_{i,p} + \beta_{i,p} \alpha_{i,p})(pr - \alpha_{i+1,p}).$$

$Q)$  with director  $(\vec{f}) = (P^2) + (Q^2) + (-1)(P + Q) = 3(\vec{N})$ , and let  $\vec{f} = p = \lambda_1\vec{u} + \nu_1$  be the tangent at  $A$  with director  $(\vec{f}) = (1)(\vec{u}) + (-1)(\vec{N}) = 3(\vec{P})$ . The director of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathbb{E}$ .

the function  $\ell_{\text{pass}} = \vec{f}^2$  is  $(\ell_{\text{pass}}) = (\vec{f}) = (\vec{f}) = (P^2) + (Q^2) + 2(\vec{N}) + (-1)(P + Q) + (-1)(\vec{N}) = 6(\vec{P})$ . The director of  $\ell_{\text{pass}} = 6(\vec{P})$  is  $(\ell_{\text{pass}}) = (\vec{f}) = (\vec{f}) = (P^2) = 3Q(1) + (-1)(P + Q) = 2(\vec{N}) = (-1)(\vec{N})$ . Notice that  $\ell_{\text{pass}}$  does not intersect  $\vec{f}$  at  $O$ ; projecting  $6(\vec{P}) = \frac{1}{2} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$  gives  $\frac{1}{2} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ , which does not give rise to any area or point at  $\mathbb{E} = 6$ . Suppose we wanted to depict the function  $\vec{f}$  on  $\mathbb{E}$ , and we multiplied out  $(y - \lambda_1)^2 - \nu_1^2(y - \lambda_1 - \nu_1)$ , substituted the  $y^2$  by  $x^2 + ax + 1$  and wrote  $p = \frac{1}{2} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathbb{E}$ ) behave at points that are not on  $\mathbb{E}$ , where the substitution  $y^2 = x^2 + ax + b$  is not possible.



man in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{D}_\ell$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm, with postponed addition steps for even  $i$  and air-like pairings.

---

**Inputs:**  $\mathcal{Q}' \in \mathcal{G}_\ell$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{i-1}, m_{i-2}, \dots, m_0)$ ,  $m_{i-1} = 1$

**Outputs:**  $f_{m, \mathcal{Q}'}(P)$  representing a class in  $\mathcal{D}_\ell / \langle \mathcal{D}_\ell^2 \rangle$

```

1  $R \leftarrow \mathcal{Q}'$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2 for  $i$  from 0 to  $i-1$  do
3   if  $(m_i = 1)$  then
4      $A_0[i] \leftarrow R$ ,  $A_1[i] \leftarrow f$ ,  $j \leftarrow j+1$ 
5   end if
6    $f \leftarrow f^2 \cdot \text{pair}_{\text{air}}(P)$ ,  $R \leftarrow [2]R$ 
7 end for
8  $R \leftarrow A_0[0]$ ,  $f \leftarrow A_1[0]$ 
9 for  $(j \leftarrow 1; j \leq \theta(m) - 1; j \leftarrow j+1)$  do
10   $f \leftarrow f \cdot A_1[j] \cdot \text{pair}_{\text{air}}(A_0[j])(P)$ ,  $R \leftarrow R + A_0[j]$ 
11 end for
12 return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gathen, Gathenstätt, and Pape [34, Algorithm 2] use a version of Algorithm 3

$$i(Q) = \theta_{ij} - pr \cdot \alpha_{i,p} \left( \frac{\theta_{ij}^2 p + \delta}{\theta_{ij}^2 p + r \theta_{i,p} + \delta} (\theta_{ij} - \alpha_{i,p}) + 1 \right).$$

We write this as  $\theta_{ij} = pr \cdot \alpha_{i,p}$ . The vertical line condition simply  $\alpha_{ij}^2 = \alpha_{ij} = \alpha_{i,p}$ . Multiplying all these together gives  $\beta_{i,p} = pr \cdot \alpha_{i,p} + \theta_{ij} \alpha_{i,p}$  where

$$\alpha_{i,p} = (\alpha'_{i,p} p - (\alpha_{ij}^2 + \delta \alpha_{ij} + \delta \theta_{i,p})) / (\theta_{ij} - \alpha_{i,p})$$

and

$$\beta_{i,p} = (\alpha_{ij}^2 + \delta \alpha_{ij} + \delta \theta_{i,p} \alpha_{i,p} + \alpha'_{i,p} (\theta_{ij} - \alpha_{i,p})).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line  $l$  is

$$y = pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (x - \alpha_{ij}) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (\theta_{ij} - \alpha_{ij}) + 1 \right).$$

Writing this as  $\theta_{ij} = pr \cdot \alpha_{i,p}$  we have  $\beta_{i+1,p} = \alpha_{i+1,p} + \theta_{ij} pr \cdot \alpha_{i+1,p}$  where

$$\alpha_{i+1,p} = (\alpha_{ij}^2 + \delta \alpha_{ij} + \delta \alpha_{i,p} \alpha_{ij} - (\alpha_{ij}^2 + \delta \alpha_{ij} + \delta \theta_{i,p} (\theta_{ij} - \alpha_{i+1,p})))$$

and

$$\beta_{i+1,p} = (\alpha_{i,p} + \beta_{i,p} \alpha_{i,p}) / (\theta_{ij} - \alpha_{i+1,p}).$$

$Q)$  with director  $(\vec{f}) = (\vec{P}) + (\vec{Q}) + (-1\vec{P} + \vec{Q}) = 3(\vec{Q})$ , and let  $\vec{f} = p = \lambda_2\vec{x} + v_1$  be the tangent at  $N$  with director  $(\vec{f}) = 3(\vec{N}) + (-3\vec{N}) = 3(\vec{P})$ . The director of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathbb{R}$ .

the function  $f_{\text{sum}} = f^2$  is  $(f_{\text{sum}}) = (f) = (\vec{f}) = (\vec{P}) + (\vec{Q}) + 3(\vec{N}) + (-1\vec{P} + \vec{Q}) + (-3\vec{N}) = 3(\vec{Q})$ . The director of  $f_{\text{sum}} = f(f)$  is  $(f_{\text{sum}}) = (f) = (\vec{P}) + 3(\vec{Q}) + (-1\vec{P} + \vec{Q}) = 3(\vec{N}) + (-3\vec{N})$ . Notice that  $f_{\text{sum}}$  does not intersect  $\mathbb{R}$  at 0; projecting  $(f) = \frac{1}{3} \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix}$  gives  $\frac{1}{3} \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix}$ , which does not give rise to any area or point at  $\mathbb{R} = 0$ . Suppose we wanted to depict the function  $f^2$  on  $\mathbb{R}$ , and we multiplied out  $(y - \lambda_2x - v_1)(y - \lambda_2x - v_1)$ , substituted the  $y^2$  by  $x^2 + ax + b$  and wrote  $p = \frac{\text{something}}{\text{something}}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathbb{R}$ ) behave at points that are not on  $\mathbb{R}$ , where the substitution  $y^2 = x^2 + ax + b$  is not permitted.

man in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{P}_\ell$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These loop costs cannot be compensated for by using affine coordinates with the inversion-saving trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm with postponed addition steps for even  $k$  and one-like pairings.

---

**Inputs:**  $Q' \in \mathcal{E}_\ell$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{1,1}, m_{1,2}, \dots, m_{1,n})$ ,  $m_{1,1} = 1$

**Outputs:**  $\sum_{m_{1,j} \neq 0} (P)$  representing a class in  $\mathcal{P}_\ell / \langle \mathcal{P}_\ell^2 \rangle$

```

1:  $R \leftarrow Q'$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2: for  $i$  from 0 to  $d-1$  do
3:   if  $(m_{1,i} = 1)$  then
4:      $A_E[1] \leftarrow R$ ,  $A_E[i] \leftarrow f$ ,  $j \leftarrow j+1$ 
5:   end if
6:    $f \leftarrow f^2 \cdot \zeta_{\text{pairing}}(P)$ ,  $R \leftarrow 2R$ 
7: end for
8:  $R \leftarrow A_E[0]$ ,  $f \leftarrow A_E[0]$ 
9: for  $(j \leftarrow 1; j \leq \theta(m) - 1; j++)$  do
10:   $f \leftarrow f \cdot A_E[j] \cdot \zeta_{\text{pairing}}(m_{1,j} P)$ ,  $R \leftarrow R + A_E[j]$ 
11: end for
12: return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gathen, Gathenstätt, and Papp [34, Algorithm 2] use a version of Algorithm 3

$$i(Q) = \theta_{ij} - pr \left( \frac{\theta_{ij}^2 + \delta}{\theta_{ij}^2 + r\theta_{ij} + \delta} (\theta_{ij} - \theta_{i,j-1}) + 1 \right).$$

We write this as  $\theta_{ij} = pr\theta_{ij}$ . The vertical line condition is simply  $\theta_{ij}^2 = \theta_{ij} = \theta_{i,j-1}$ . Multiplying all these together gives  $\theta_{i,j-1} = pr\theta_{i,j-1} + \theta_{i,j-1}\theta_{i,j-1}$  where

$$\theta_{i,j-1} = (\theta_{i,j-1}^2 + \delta) - (\theta_{ij}^2 + \delta) + \delta(\theta_{i,j-1} - \theta_{ij})$$

and

$$\theta_{i,j-1} = (\theta_{ij}^2 + \delta) + \delta(\theta_{i,j-1} - \theta_{ij}) + \delta(\theta_{i,j-1} - \theta_{ij})$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line is

$$y = pr \left( \frac{\theta_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\theta_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

Writing this as  $\theta_{ij} = pr\theta_{ij}$ , we have  $\theta_{i,j-1} = \theta_{i,j-1} + \theta_{i,j-1}\theta_{i,j-1}$  where

$$\theta_{i,j-1} = (\theta_{i,j-1}^2 + \delta) + \delta(\theta_{i,j-1} - \theta_{ij}) + \delta(\theta_{i,j-1} - \theta_{ij})$$

and

$$\theta_{i,j-1} = (\theta_{i,j-1} + \theta_{i,j-1}\theta_{i,j-1}) - (\theta_{ij} + \theta_{ij}\theta_{ij})$$

$Q)$  with director  $(\vec{f}) = (P^2) + (Q^2) + (-1(P^2 + Q^2) - 3)(\vec{N})$ , and let  $\vec{f} \cdot \vec{p} = \lambda_2 p + r_1$  be the tangent at  $\vec{f}$  with director  $(\vec{f}) = (1)(\vec{N}) + (-3)(\vec{N}) = 3(\vec{N})$ . The director of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathbb{R}$ .

the function  $\ell_{\text{sum}} = \vec{f}^2$  is  $(\ell_{\text{sum}}) = (\vec{f}) = (\vec{f}) = (P^2) + (Q^2) + 2(\vec{N}) + (-1(P^2 + Q^2) + (-3)(\vec{N}) - 6)(\vec{N})$ . The director of  $\ell_{\text{sum}} = (\vec{f})$  is  $(\ell_{\text{sum}}) = (\vec{f}) = (P^2) + (Q^2) + (-1(P^2 + Q^2) - 3)(\vec{N}) = 2(\vec{N}) + (-1(P^2 + Q^2) - 3)(\vec{N})$ . Notice that  $\ell_{\text{sum}}$  does not intersect  $\vec{f}$  at  $\vec{f}$ , projecting  $(\vec{f}) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$  gives  $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ , which does not give rise to any area or point at  $\vec{f} = \vec{f}$ . Suppose we wanted to depict the function  $\vec{f}$  on  $\vec{f}$ , and we multiplied out  $(\vec{f} - \lambda_2 p - r_1)(\vec{f} - \lambda_2 p - r_1)$ , substituted the  $y^2$  by  $x^2 + ax + b$  and wrote  $p = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\vec{f}$ ) behave at points that are not on  $\vec{f}$ , where the substitution  $y^2 = x^2 + ax + b$  is not possible.

man in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{P}_\ell$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm, with postponed addition steps for even  $i$  and one-like pairings.

---

**Inputs:**  $Q' \in G_2$ ,  $P \in G_1$ ,  $m = (m_1, m_2, \dots, m_t)$ ,  $m_{t+1} = 1$

**Outputs:**  $\sum_{m_i \neq 0} (P)$  representing a class in  $\mathcal{P}_\ell / \mathcal{P}_\ell^*$

```

1:  $R \leftarrow Q'$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2: for  $i$  from 0 to  $t - 1$  do
3:   if  $(m_i = 1)$  then
4:      $A_E[1] \leftarrow R$ ,  $A_E[i] \leftarrow f$ ,  $j \leftarrow j + 1$ 
5:   end if
6:    $f \leftarrow f^2 \cdot \text{pairing}(P)$ ,  $R \leftarrow 2R$ 
7: end for
8:  $R \leftarrow A_E[0]$ ,  $f \leftarrow A_E[0]$ 
9: for  $(j \leftarrow 1; j \leq \#(m) - 1; j++)$  do
10:   $f \leftarrow f \cdot A_E[j] \cdot \text{pairing}(P)$ ,  $R \leftarrow R + A_E[j]$ 
11: end for
12: return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gathen, Gathenstätt, and Pape [34, Algorithm 2] use a variant of Algorithm 3

$$i(Q) = \theta_{ij} - pr \left( \frac{\theta_{ij}^2 + \delta}{\theta_{ij}^2 + r\theta_{ij} + \delta} (\theta_{ij} - \theta_{i,j-1}) + 1 \right).$$

We write this as  $\theta_{ij} = pr\theta_{ij}$ . The vertical line satisfies simply  $\theta_{ij}^2 = \theta_{ij} = \theta_{i,j-1}$ . Multiplying all these together gives  $\theta_{i,j-1} = pr\theta_{i,j-1} + \theta_{i,j-1}\theta_{i,j-1}$  where

$$\theta_{i,j-1} = (\theta_{i,j-1}^2 + \delta) + \theta_{ij} + \delta(\theta_{i,j-1})(\theta_{ij} - \theta_{i,j-1})$$

and

$$\theta_{i,j-1} = (\theta_{ij}^2 + \delta) + \theta_{ij} + \delta(\theta_{i,j-1})(\theta_{ij} - \theta_{i,j-1}).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line is

$$y = pr \left( \frac{\theta_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\theta_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

Writing this as  $\theta_{ij} + pr\theta_{ij}$ , we have  $\theta_{i,j+1} = \theta_{i,j+1} + \theta_{ij}pr\theta_{i,j+1}$  where

$$\theta_{i,j+1} = (\theta_{ij}^2 + \delta) + \theta_{ij} + \delta(\theta_{i,j+1}) - (\theta_{ij}^2 + \delta) + \theta_{ij} + \delta(\theta_{i,j+1})(\theta_{ij} - \theta_{i,j+1})$$

and

$$\theta_{i,j+1} = (\theta_{i,j+1} + \theta_{i,j+1}\theta_{ij})(\theta_{ij} - \theta_{i,j+1}).$$



$Q)$  with director  $(\vec{f}) = (\vec{P}) + (\vec{Q}) + (-1\vec{P} + \vec{Q}) = 3(\vec{Q})$ , and let  $\vec{f} = p = \lambda_2\vec{x} + v_1$  be the tangent at  $N$  with director  $(\vec{f}) = 3(\vec{N}) + (-3\vec{N}) = 3(\vec{P})$ . The director of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathbb{R}$ .

the function  $f_{\text{sum}} = f^2$  is  $(f_{\text{sum}}) = (f) = (\vec{f}) = (\vec{P}) + (\vec{Q}) + 3(\vec{N}) + (-1\vec{P} + \vec{Q}) + (-3\vec{N}) = 6(\vec{Q})$ . The director of  $f_{\text{sum}} = f(f)$  is  $(f_{\text{sum}}) = (f) = (\vec{f}) = 3(\vec{Q}) + (-3\vec{N}) + (-1\vec{P} + \vec{Q}) = 3(\vec{N}) + (-3\vec{N})$ . Notice that  $f_{\text{sum}}$  does not intersect  $\mathbb{R}$  at 0; projecting  $(f^2) = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$  gives  $\frac{1}{2} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ , which does not give rise to any axes or point at  $\mathbb{R} = 0$ . Suppose we wanted to depict the function  $f^2$  on  $\mathbb{R}$ , and we multiplied out  $(y - \lambda_2x - v_1)(y - \lambda_2x - v_1)$ , substituted the  $y^2$  by  $x^2 + ax + b$  and wrote  $p = \frac{\text{something}}{\text{something}}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathbb{R}$ ) behave at points that are not on  $\mathbb{R}$ , where the substitution  $y^2 = x^2 + ax + b$  is not permitted.

man in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{P}_\ell$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These loop costs cannot be compensated for by using affine coordinates with the inversion-saving trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm with postponed addition steps for even  $k$  and one-like pairings.

---

**Inputs:**  $Q' \in \mathcal{E}_\ell$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{1,1}, m_{1,2}, \dots, m_{1,n})$ ,  $m_{1,1} = 1$

**Outputs:**  $\sum_{m_i \neq 0} (P)$  representing a class in  $\mathcal{P}_\ell / \langle \mathcal{P}_\ell^2 \rangle$

```

1:  $R \leftarrow Q'$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2: for  $i$  from 0 to  $d-1$  do
3:   if  $(m_i = 1)$  then
4:      $A_E[1] \leftarrow R$ ,  $A_E[j] \leftarrow f$ ,  $j \leftarrow j+1$ 
5:   end if
6:    $f \leftarrow f^2 \cdot \text{Line}_{R,2R}(P)$ ,  $R \leftarrow 2R$ 
7: end for
8:  $R \leftarrow A_E[0]$ ,  $f \leftarrow A_E[0]$ 
9: for  $(j \leftarrow 1; j \leq \theta(m) - 1; j \leftarrow j+1)$  do
10:   $f \leftarrow f \cdot A_E[j] \cdot \text{Line}_{R,A_E[j]}(P)$ ,  $R \leftarrow R + A_E[j]$ 
11: end for
12: return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gathen, Gathenstätt, and Papp [34, Algorithm 2] use a version of Algorithm 3

$$i(Q) = \theta_{ij} - pr \left( \frac{\theta_{ij}^2 + \delta}{\theta_{ij}^2 + r\theta_{ij} + \delta} (\theta_{ij} - \theta_{i,j-1}) + 1 \right).$$

We write this as  $\theta_{ij} = pr\theta_{ij}$ . The vertical line condition is simply  $\theta_{ij}^2 = \theta_{ij} = \theta_{i,j-1}$ . Multiplying all these together gives  $\theta_{i,j-1} = pr\theta_{i,j-1} + \theta_{i,j-1}\theta_{i,j-1}$  where

$$\theta_{i,j-1} = (\theta_{i,j-1}^2 + \delta) - (\theta_{ij}^2 + \delta) + \delta(\theta_{i,j-1}^2)(\theta_{ij} - \theta_{i,j-1})$$

and

$$\theta_{i,j-1} = (\theta_{ij}^2 + \delta) + \delta(\theta_{i,j-1}^2)(\theta_{ij} - \theta_{i,j-1}).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line is

$$y = pr \left( \frac{\theta_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\theta_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

Writing this as  $\theta_{ij} + pr\theta_{ij}$ , we have  $\theta_{i,j+1} = \theta_{i,j+1} + \theta_{ij}pr\theta_{i,j+1}$  where

$$\theta_{i,j+1} = (\theta_{ij}^2 + \delta) + \delta(\theta_{i,j+1}^2) - (\theta_{ij}^2 + \delta) + \delta(\theta_{i,j+1}^2)(\theta_{ij} - \theta_{i,j+1})$$

and

$$\theta_{i,j+1} = (\theta_{i,j+1} + \theta_{i,j+1}\theta_{ij})(\theta_{ij} - \theta_{i,j+1}).$$

$Q$ ) with director  $(\vec{f}) = (\vec{P}) + \frac{1}{2}(\vec{f}) + \frac{1}{2}(-(\vec{P} + \vec{Q}) - 3(\vec{Z}))$ , and let  $\vec{f} = p\lambda_2 + r_1$  be the tangent at  $\vec{f}$  with director  $(\vec{f}) = \frac{1}{2}(\vec{f}) + \frac{1}{2}(\vec{f}) = 3(\vec{f})$ . The director of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathbb{P}^2$ .

the function  $\ell_{\text{sum}} = \vec{f}$  is  $(\ell_{\text{sum}}) = (\vec{f}) = (\vec{P}) + (\vec{Q}) + \frac{1}{2}(\vec{f}) + \frac{1}{2}(-(\vec{P} + \vec{Q}) - 3(\vec{Z})) = 3(\vec{f})$ . The director of  $\ell_{\text{sum}} = \vec{f}$  is  $(\ell_{\text{sum}}) = (\vec{f}) = (\vec{P}) + \frac{1}{2}(\vec{f}) + \frac{1}{2}(-(\vec{P} + \vec{Q}) - 3(\vec{Z})) = \frac{1}{2}(\vec{f}) + \frac{1}{2}(-(\vec{P} + \vec{Q}) - 3(\vec{Z}))$ . Notice that  $\ell_{\text{sum}}$  does not intersect  $\vec{f}$  at  $\vec{f}$ , projecting  $\vec{f}$  to  $\frac{1}{2}(\vec{f}) + \frac{1}{2}(-(\vec{P} + \vec{Q}) - 3(\vec{Z}))$ , which does not give rise to any zero or pole at  $\vec{f} = \vec{f}$ . Suppose we wanted to depict the function  $\vec{f}$  on  $\mathbb{P}^2$ , and we multiplied out  $(y - \lambda_2 - r_1)(y - \lambda_2 - r_1)$ , substituted the  $y^2$  by  $x^2 + ax + b$  and wrote  $p = \frac{(x^2 + ax + b)(x^2 + ax + b)}{(x^2 + ax + b)}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathbb{P}^2$ ) behave at points that are not on  $\mathbb{P}^2$ , where the substitution  $y^2 = x^2 + ax + b$  is not possible.

man in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{D}_\ell$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm, with postponed addition steps for even  $i$  and air-like pairings.

---

**Inputs:**  $Q' \in \mathcal{G}_2$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{d-1}, m_{d-2}, \dots, m_0)$ ,  $m_{d-1} = 1$

**Outputs:**  $f_{m,Q'}(P)$  representing a class in  $\mathcal{D}_\ell / \langle \mathcal{D}_\ell^2 \rangle$

```

1  $R \leftarrow Q'$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2 for  $i$  from 0 to  $d - 1$  do
3   if  $(m_i = 1)$  then
4      $A_0[i] \leftarrow R$ ,  $A_1[i] \leftarrow f$ ,  $j \leftarrow j + 1$ 
5   end if
6    $f \leftarrow f^2 \cdot \zeta_{\text{pairing}}(P)$ ,  $R \leftarrow [2]R$ 
7 end for
8  $R \leftarrow A_0[j]$ ,  $f \leftarrow A_1[j]$ 
9 for  $(i \leftarrow 1; j \leq d(m) - 1; i \leftarrow i + 1)$  do
10   $f \leftarrow f \cdot A_1[i] \cdot \zeta_{\text{pairing}}([2^i]P)$ ,  $R \leftarrow R + A_0[i]$ 
11 end for
12 return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gauthier, Gaudschöld, and Fuchs [34, Algorithm 2] use a version of Algorithm 3

$$i(Q) = \theta_{ij} - pr\alpha_{i,j} \left( \frac{\theta_{ij}^2 + \delta}{\theta_{i,j}^2 + r\alpha_{i,j} + \delta} (\theta_{ij} - \alpha_{i,j}) + 1 \right).$$

We write this as  $\theta_{ij} = pr\alpha_{i,j}$ . The vertical line condition is simply  $\alpha_{ij}^2 = \alpha_{ij} = \alpha_{i,j}$ . Multiplying all three together gives  $\beta_{i,j} = pr\alpha_{i,j} + \theta_{i,j}\alpha_{i,j}$  where

$$\alpha_{i,j} = (\alpha_{i,j}^2 + \delta - (\theta_{ij}^2 + \delta\theta_{ij} + \delta)(\theta_{ij} - \alpha_{i,j}))$$

and

$$\beta_{i,j} = (\theta_{ij}^2 + \delta\theta_{ij} + \delta\theta_{i,j}\alpha_{i,j} + \alpha_{i,j}^2)(\theta_{ij} - \alpha_{i,j}).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the values. In general case, from Lemma 2 we deduce that the line  $l$  is

$$y = pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (x - pr) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

Writing this as  $\theta_{ij} = pr\alpha_{i,j}$  we have  $\beta_{i+1,j} = \alpha_{i+1,j} + \theta_{ij}pr\alpha_{i+1,j}$  where

$$\alpha_{i+1,j} = (\theta_{ij}^2 + \delta\theta_{ij} + \delta\alpha_{i,j}\theta_{ij} - (\theta_{ij}^2 + \delta\theta_{ij} + \delta)(\theta_{ij} - \alpha_{i+1,j})).$$

and

$$\beta_{i+1,j} = (\alpha_{i,j} + \beta_{i,j}\alpha_{i,j})(\theta_{ij} - \alpha_{i+1,j}).$$

$Q)$  with director  $(f) = (f^P) + (f^Q) + (-)(f^R + Q) = 3(Z)$ , and let  $f : p = \lambda_2x + x_1$  be the tangent at  $R$  with director  $(f) = (f^P) + (-)(f^R) = 3(X)$ . The director of



Figure 3.5: Two functions  $f$  and  $g$  on  $\mathbb{E}$ .

the function  $f_{\text{sum}} = f^P$  is  $(f_{\text{sum}}) = (f) = (f^P) = (f^P) + (Q) + 2(f^R) + (-)(f^R + Q) + (-)(3, R) = 0(Z)$ . The director of  $f_{\text{sum}} = 0(Z)$  is  $(f_{\text{sum}}) = (f) = (f^P) = 3(Z) + (-)(f^R + Q) = 2(Z) + (-)(3(R))$ . Notice that  $f_{\text{sum}}$  does not intersect  $\mathbb{E}$  at  $O$ ; projecting  $0(Z) = \frac{1}{3} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$  gives  $\frac{1}{3} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ , which does not give rise to any area or point at  $\mathbb{E} = 0$ . Suppose we wanted to depict the function  $f^P$  on  $\mathbb{E}$ , and we multiplied out  $(y - \lambda_2x - x_1)(y - \lambda_2x - x_1)$ , substituted the  $y^2$  by  $x^2 + ax + b$  and wrote  $p = \frac{(x^2 + ax + b)(x^2 + ax + b)}{(x^2 + ax + b)}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathbb{E}$ ) behave at points that are not on  $\mathbb{E}$ , where the substitution  $y^2 = x^2 + ax + b$  is not possible.

max in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{P}_\ell$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm with postponed addition steps for even  $k$  and ate-like pairings.

---

**Inputs:**  $\mathcal{Q}^* \in \mathcal{G}_2$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{d-1}, m_{d-2}, \dots, m_0)$ ,  $m_{d-1} = 1$

**Outputs:**  $f_{m, \mathcal{Q}^*}(P)$  representing a class in  $\mathcal{P}_\ell / \langle \mathcal{P}_\ell^2 \rangle$

```

1  $R \leftarrow \mathcal{Q}^*$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2 for  $i$  from 0 to  $d - 1$  do
3   if  $(m_i = 1)$  then
4      $A_R[1] \leftarrow R$ ,  $A_f[1] \leftarrow f$ ,  $j \leftarrow j + 1$ 
5   end if
6    $f \leftarrow f^2 \cdot \zeta_{m, \mathcal{Q}^*}(P)$ ,  $R \leftarrow 2R$ 
7 end for
8  $R \leftarrow A_R[j]$ ,  $f \leftarrow A_f[j]$ 
9 for  $i \leftarrow 1$ ;  $j \leftarrow \phi(m) - 1$ ;  $i \leftarrow i + 1$  do
10   $f \leftarrow f \cdot A_f[i] \cdot \zeta_{m, \mathcal{Q}^*}(A_R[i])$ ,  $R \leftarrow R + A_R[i]$ 
11 end for
12 return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of multicore machines. Gauthier, Gaudsichet, and Pape [34, Algorithm 2] use a version of Algorithm 3



$$i(Q) = \theta_{ij} - pr \cdot \alpha_{i,p} \left( \frac{\theta_{ij}^2 + \delta}{\theta_{ij}^2 + r\theta_{i,p} + \delta} (\theta_{ij} - \alpha_{i,p}) + 1 \right).$$

We write this as  $\theta_{ij} = pr \cdot \alpha_{i,p}$ . The vertical line condition simply  $\alpha_{ij}^2 = \alpha_{ij} = \alpha_{i,p}$ . Multiplying all these together gives  $\beta_{i,p} = pr \cdot \alpha_{i,p} + \theta_{ij} \alpha_{i,p}$  where

$$\alpha_{i,p} = (\alpha_{i,p}^2 - (\theta_{ij}^2 + \delta \theta_{ij} + \delta (\theta_{ij}^2 + r\theta_{i,p} + \delta) (\theta_{ij} - \alpha_{i,p})))$$

and

$$\beta_{i,p} = (\alpha_{i,p}^2 + \delta \theta_{ij} + \delta (\theta_{ij}^2 + r\theta_{i,p} + \delta (\theta_{ij}^2 + r\theta_{i,p} + \delta) (\theta_{ij} - \alpha_{i,p}))).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line  $l$  is

$$y = pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (x - \alpha_{ij}) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (\theta_{ij} - \alpha_{ij}) + 1 \right).$$

Writing this as  $\theta_{ij} = pr \cdot \alpha_{i,p}$  we have  $\beta_{i+1,p} = \alpha_{i+1,p} + \theta_{ij} pr \cdot \alpha_{i+1,p}$  where

$$\alpha_{i+1,p} = (\alpha_{i,p}^2 + \delta \theta_{ij} + \delta (\alpha_{i,p}^2 - (\theta_{ij}^2 + \delta \theta_{ij} + \delta (\theta_{ij}^2 + r\theta_{i,p} + \delta) (\theta_{ij} - \alpha_{i,p}))))$$

and

$$\beta_{i+1,p} = (\alpha_{i,p} + \beta_{i,p} \theta_{ij}) (\theta_{ij} - \alpha_{i+1,p}).$$

$Q)$  with director  $(\vec{f}) = (P^2) + (Q^2) + (-1)(P^2 + Q^2) = 3(\vec{N})$ , and let  $\vec{f} = p = \lambda_{22} + r_1$  be the tangent at  $A$  with director  $(\vec{f}) = (1)(N) + (-1)(N) = 3(\vec{P})$ . The director of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathbb{R}$ .

the function  $\ell_{\text{pass}} = \vec{f}$  is  $(\ell_{\text{pass}}) = (\vec{f}) = (P^2) + (Q^2) + (1)(N) + (-1)(P^2 + Q^2) + (-1)(N) = 0(\vec{P})$ . The director of  $\ell_{\text{pass}} = 0(\vec{P})$  is  $(\ell_{\text{pass}}) = (\vec{f}) = (P^2) = (Q^2) + (-1)(P^2 + Q^2) = (1)(N) + (-1)(N)$ . Notice that  $\ell_{\text{pass}}$  does not intersect  $\vec{f}$  at  $A$ , perfectly lying  $0(\vec{P}) = \frac{1}{2} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$  gives  $\frac{1}{2} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ , which does not give rise to any area or point at  $\mathbb{Z} = 0$ . Suppose we wanted to depict the function  $\vec{f}$  on  $\mathbb{Z}$ , and we multiplied out  $(y - \lambda_{22} - r_1)(y - \lambda_{22} - r_1)$ , substituted the  $y^2$  by  $x^2 + ax + b$  and wrote  $p = \frac{1}{2} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathbb{Z}$ ) behave at points that are not on  $\mathbb{Z}$ , where the substitution  $y^2 = x^2 + ax + b$  is not possible.

man in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{D}_\ell$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm, with postponed addition steps for even  $i$  and air-like pairings.

---

**Inputs:**  $\mathcal{Q}' \in \mathcal{G}_\ell$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{i-1}, m_{i-2}, \dots, m_0)$ ,  $m_{i-1} = 1$

**Outputs:**  $\sum_{m_i \neq 0} f(P)$  representing a class in  $\mathcal{D}_\ell / \langle \mathcal{D}_\ell^2 \rangle$

```

1  $R \leftarrow \mathcal{Q}'$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2 for  $i$  from 0 to  $i-1$  do
3   if  $(m_i = 1)$  then
4      $A_0[1] \leftarrow R$ ,  $A_1[1] \leftarrow f$ ,  $j \leftarrow j+1$ 
5   end if
6    $f \leftarrow f^2 \cdot \text{pair}_{\text{air}}(P)$ ,  $R \leftarrow [2]R$ 
7 end for
8  $R \leftarrow A_0[0]$ ,  $f \leftarrow A_1[0]$ 
9 for  $(j \leftarrow 1; j \leq \theta(m) - 1; j \leftarrow j+1)$  do
10   $f \leftarrow f \cdot A_1[j] \cdot \text{pair}_{\text{air}}(R, A_0[j])$ ,  $R \leftarrow R + A_0[j]$ 
11 end for
12 return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gathen, Gathenstätt, and Pape [34, Algorithm 2] use a version of Algorithm 3

$$i(Q) = \theta_{ij} - pr \cdot \alpha_{i,p} \left( \frac{\theta_{ij}^2 + \delta}{\theta_{ij}^2 + r\theta_{i,p} + \delta} (\theta_{ij} - \alpha_{i,p}) + 1 \right).$$

We write this as  $\theta_{ij} = pr \cdot \alpha_{i,p}$ . The vertical line condition simply  $\alpha_{ij}^2 = \alpha_{ij} = \alpha_{i,p}$ . Multiplying all these together gives  $\beta_{i,p} = pr \cdot \alpha_{i,p} + \theta_{ij} \alpha_{i,p}$  where

$$\alpha_{i,p} = (\alpha_{i,p}^2 - (\theta_{ij}^2 + \delta \theta_{ij} + \delta)(\theta_{ij} - \alpha_{i,p}))$$

and

$$\beta_{i,p} = (\alpha_{ij}^2 + \delta \theta_{ij} + \delta \theta_{ij} \alpha_{i,p} + \alpha_{i,p}^2)(\theta_{ij} - \alpha_{i,p}).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line  $l$  is

$$y = pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (x - \alpha_{ij}) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (\theta_{ij} - \alpha_{ij}) + 1 \right).$$

Writing this as  $\theta_{ij} = pr \cdot \alpha_{i,p}$  we have  $\beta_{i+1,p} = \alpha_{i+1,p} + \theta_{ij} pr \cdot \alpha_{i+1,p}$  where

$$\alpha_{i+1,p} = (\alpha_{ij}^2 + \delta \theta_{ij} + \delta \theta_{ij} \alpha_{i,p} - (\theta_{ij}^2 + \delta \theta_{ij} + \delta)(\theta_{ij} - \alpha_{i+1,p})).$$

and

$$\beta_{i+1,p} = (\alpha_{i,p} + \beta_{i,p} \alpha_{i,p})(\theta_{ij} - \alpha_{i+1,p}).$$

$Q$ ) with director  $(\vec{f}) = (\vec{P}) + 1\vec{Q} + (-1\vec{P} + \vec{Q}) = 2(\vec{Q})$ , and let  $\vec{f} = p + \lambda_2\vec{u} + \nu_1\vec{v}$  be the tangent at  $\vec{P}$  with director  $(\vec{f}) = 2(\vec{P}) + (-3\vec{Q}) = 3(\vec{P})$ . The director of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathbb{R}$ .

the function  $\ell_{\text{pass}} = \vec{f}$  is  $(\ell_{\text{pass}}) = (\vec{f}) = (\vec{P}) + (\vec{Q}) + 2(\vec{P}) + (-1\vec{P} + \vec{Q}) + (-3\vec{Q}) = 6(\vec{P})$ . The director of  $\ell_{\text{pass}} = \vec{f}$  is  $(\ell_{\text{pass}}) = (\vec{f}) = (\vec{P}) + (\vec{Q}) + (-1\vec{P} + \vec{Q}) = 2(\vec{P}) + (-3\vec{Q}) = 2(\vec{P}) + (-3\vec{Q})$ . Notice that  $\ell_{\text{pass}}$  does not intersect  $\vec{f}$  at  $\vec{Q}$ ; projecting  $\vec{f}$  =  $\frac{1}{2} \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$  gives  $\frac{1}{2} \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$ , which does not give rise to any area or point at  $\vec{Q} = 0$ . Suppose we wanted to depict the function  $\vec{f}$  on  $\mathbb{R}$ , and we multiplied out  $(y - \lambda_2 - \nu_1)(y - \lambda_2 - \nu_1)$ , substituted the  $y^2$  by  $x^2 + ax + b$  and wrote  $p = \frac{(x^2 + ax + b)(x^2 + ax + b)}{(x^2 + ax + b)}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathbb{R}$ ) behave at points that are not on  $\mathbb{R}$ , where the substitution  $y^2 = x^2 + ax + b$  is not possible.

max in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{P}_d$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm with postponed addition steps for even  $k$  and ate-like pairings.

---

**Inputs:**  $\mathcal{Q}^* \in \mathcal{G}_2$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{d-1}, m_{d-2}, \dots, m_0)$ ,  $m_{d-1} = 1$

**Outputs:**  $f_{m, \mathcal{Q}^*}(P)$  representing a class in  $\mathcal{P}_d / \langle \mathcal{P}_d^* f \rangle$

```

1  $R \leftarrow \mathcal{Q}^*$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2 for  $i$  from 0 to  $d - 1$  do
3   if  $(m_i = 1)$  then
4      $A_R[1] \leftarrow R$ ,  $A_f[1] \leftarrow f$ ,  $j \leftarrow j + 1$ 
5   end if
6    $f \leftarrow f^2 \cdot \zeta_{m, \mathcal{Q}^*}(P)$ ,  $R \leftarrow 2R$ 
7 end for
8  $R \leftarrow A_R[j]$ ,  $f \leftarrow A_f[j]$ 
9 for  $i \leftarrow 1$ ;  $j \leftarrow \phi(m) - 1$ ;  $i \leftarrow +$  do
10   $f \leftarrow f \cdot A_f[i] \cdot \zeta_{m, \mathcal{Q}^*}(A_R[i])$ ,  $R \leftarrow R + A_R[i]$ 
11 end for
12 return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gauthier, Gaudschöld, and Pape [34, Algorithm 2] use a version of Algorithm 3

$$i(Q) = \theta_{ij} - pr \left( \frac{\theta_{ij}^2 + \delta}{\theta_{ij}^2 + r\theta_{ij} + \delta} (\theta_{ij} - \theta_{i,p}) + 1 \right).$$

We write this as  $\theta_{ij} = pr\theta_{i,p}$ . The vertical line condition simply  $\theta_{ij}^2 = \theta_{ij} = \theta_{i,p}$ . Multiplying all these together gives  $\delta_{i,p} = pr\alpha_{i,p} + \theta_{i,p}\delta_{i,p}$  where

$$\alpha_{i,p} = (\theta_{i,p}^2 - (\theta_{ij}^2 + \delta r_{ij} + \delta(\theta_{i,p}^2)))(\theta_{ij} - \theta_{i,p})$$

and

$$\delta_{i,p} = (\theta_{ij}^2 + \delta r_{ij} + \delta(\theta_{i,p}^2) + \theta_{i,p}^2)(\theta_{ij} - \theta_{i,p}).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line  $l$  is

$$y = pr \left( \frac{\theta_{ij} - 1}{\theta_{ij} - pr} (x - pr) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\theta_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

Writing this as  $\theta_{ij} + pr\theta_{i+1,p}$  we have  $\delta_{i+1,p} = \theta_{i+1,p} + \theta_{ij}pr\delta_{i+1,p}$  where

$$\alpha_{i+1,p} = (\theta_{ij}^2 + \delta r_{ij} + \delta(\alpha_{i,p}^2) - (\theta_{ij}^2 + \delta r_{ij} + \delta(\theta_{i,p}^2)))(\theta_{ij} - \theta_{i+1,p}),$$

and

$$\delta_{i+1,p} = (\theta_{i,p} + \delta_{i,p}r_{ij})(\theta_{ij} - \theta_{i+1,p}).$$

$Q)$  with direction  $(f) = (f^x) + (f^y) + (-)(f^z) = (f^x) + (f^y) + 2(f^z)$ , and let  $\mathcal{F} : p = \lambda_2 q + r_1$  be the tangent at  $\mathcal{F}$  with direction  $(f) = (f^x) + (-)(f^y) = 3(f^z)$ . The direction of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathcal{Z}$ .

the function  $\ell_{\text{max}} = \ell^x$  is  $(\ell_{\text{max}}) = (f) - (f^x) = (f^x) + (f^y) + 2(f^z) + (-)(f^x + Qf) + (-)(1, 0) = 0(f^z)$ . The direction of  $\ell_{\text{max}} = 0(\mathcal{F})$  is  $(\ell_{\text{max}}) = (f) - (f^x) = (f^x) + (f^y) + (-)(f^z) = 2(f^z)$ . Notice that  $\ell_{\text{max}}$  does not intersect  $\mathcal{Z}$  at  $\mathcal{O}$ ; projecting  $0(\mathcal{F}) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$  gives  $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ , which does not give rise to any area or point at  $\mathcal{Z} = 0$ . Suppose we wanted to depict the function  $\ell^x$  on  $\mathcal{Z}$ , and we multiplied out  $1_y - \lambda_2 q - r_1^x(1_y - \lambda_2 q - r_1^x)$ , substituted the  $y^2$  by  $x^2 + ax + 1$  and wrote  $p = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathcal{Z}$ ) behave at points that are not on  $\mathcal{Z}$ , where the substitution  $y^2 = x^2 + ax + 1$  is not permitted.



man in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{P}_\ell$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm with postponed addition steps for even  $k$  and air-like pairings.

---

**Inputs:**  $Q' \in \mathcal{G}_\ell$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{-1}, m_{-2}, \dots, m_0)$ ,  $m_{-1} = 1$

**Outputs:**  $f_{m,Q'}(P)$  representing a class in  $\mathcal{P}_\ell / \langle \mathcal{P}_\ell^2 \rangle$

```

1  $R \leftarrow Q'$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2 for  $i$  from 0 to  $d - 1$  do
3   if  $(m_i = 1)$  then
4      $A_{\ell}[i] \leftarrow R$ ,  $A_0[i] \leftarrow f$ ,  $j \leftarrow j + 1$ 
5   end if
6    $f \leftarrow f^2 \cdot \zeta_{\text{air},\text{even}}(P)$ ,  $R \leftarrow [2]R$ 
7 end for
8  $R \leftarrow A_0[0]$ ,  $f \leftarrow A_0[0]$ 
9 for  $(j \leftarrow 1; j \leq \theta(m) - 1; j \leftarrow j + 1)$  do
10   $f \leftarrow f \cdot A_0[j] \cdot \zeta_{\text{air},\text{even},\text{air}}([j]P)$ ,  $R \leftarrow R + A_{\ell}[j]$ 
11 end for
12 return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gauthier, Gauthier, and Pape [34, Algorithm 2] use a version of Algorithm 3

$$i(Q) = \theta_{ij} - pr \left( \frac{\theta_{ij}^2 + \delta}{\theta_{ij}^2 + r\theta_{ij} + \delta} (\theta_{ij} - \alpha_{ij}) + 1 \right).$$

We write this as  $\theta_{ij} = pr\alpha_{ij}$ . The vertical line condition simply  $\alpha_{ij}^2 = \alpha_{ij} = \alpha_{ij,p}$ . Multiplying all these together gives  $\beta_{ij,p} = pr\alpha_{ij,p} + \theta_{ij,p}\alpha_{ij,p}$  where

$$\alpha_{ij,p} = (\alpha_{ij,p}^2 - (\theta_{ij}^2 + \delta r_{ij} + \delta)(\theta_{ij} - \alpha_{ij,p}))$$

and

$$\beta_{ij,p} = (\alpha_{ij}^2 + \delta r_{ij} + \delta)(\theta_{ij,p} + \alpha_{ij,p}^2)(\theta_{ij} - \alpha_{ij,p}).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line is

$$y = pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (\alpha_{ij} - \alpha_{ij}) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (\alpha_{ij} - \alpha_{ij}) + 1 \right).$$

Writing this as  $\theta_{ij} = pr\alpha_{ij}$ , we have  $\beta_{i+1,p} = \alpha_{i+1,p} + \theta_{i+1,p}\alpha_{i+1,p}$  where

$$\alpha_{i+1,p} = (\alpha_{ij}^2 + \delta r_{ij} + \delta)(\alpha_{ij,p} + \alpha_{ij,p}^2) - (\alpha_{ij}^2 + \delta r_{ij} + \delta)(\theta_{ij} - \alpha_{i+1,p}).$$

and

$$\beta_{i+1,p} = (\alpha_{ij,p} + \beta_{ij,p}\alpha_{ij,p})(\theta_{ij} - \alpha_{i+1,p}).$$

$Q)$  with director  $(\vec{f}) = (P^2) + (Q^2) + (-)(P^2 + Q^2) = 3(\vec{N})$ , and let  $\vec{f} = p = \lambda_{22} + r_1$  be the tangent at  $A$  with director  $(\vec{f}) = (1)(N) + (-)(N) = 3(\vec{P})$ . The director of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathbb{R}$ .

the function  $f_{\text{para}} = f^2$  is  $(f_{\text{para}}) = (f) = (P^2) = (P^2) + (Q^2) + (1)(N) + (-)(P^2 + Q^2) + (-)(1, 0) = 0(\vec{P})$ . The director of  $f_{\text{para}} = (f)^2$  is  $(f_{\text{para}}) = (f) = (P^2) = (Q^2) + (-)(P^2 + Q^2) = (1)(N) + (-)(N)$ . Notice that  $f_{\text{para}}$  does not intersect  $\mathbb{R}$  at  $0$ , projecting  $(f)^2 = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$  gives  $\frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ , which does not give rise to any area or point at  $\mathbb{R} = 0$ . Suppose we wanted to depict the function  $f^2$  on  $\mathbb{R}$ , and we multiplied out  $(y - \lambda_{22} - r_1)(y - \lambda_{22} - r_1)$ , substituted the  $y^2$  by  $x^2 + ax + b$  and wrote  $p = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathbb{R}$ ) behave at points that are not on  $\mathbb{R}$ , where the substitution  $y^2 = x^2 + ax + b$  is not possible.

man in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{P}_d$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm with postponed addition steps for even  $k$  and ate-like pairings.

---

**Inputs:**  $\mathcal{Q}^* \in \mathcal{G}_2$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{d-1}, m_{d-2}, \dots, m_0)$ ,  $m_{d-1} = 1$

**Outputs:**  $f_{m, \mathcal{Q}^*}(P)$  representing a class in  $\mathcal{P}_d / \langle \mathcal{P}_d^* f \rangle$

```

1  $R \leftarrow \mathcal{Q}^*$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2 for  $i$  from 0 to  $d-1$  do
3   if  $(m_i = 1)$  then
4      $A_R[1] \leftarrow R$ ,  $A_f[1] \leftarrow f$ ,  $j \leftarrow j+1$ 
5   end if
6    $f \leftarrow f^2 \cdot \zeta_{\text{sat}(\mathcal{Q}^*)}(P)$ ,  $R \leftarrow [2]R$ 
7 end for
8  $R \leftarrow A_R[j]$ ,  $f \leftarrow A_f[j]$ 
9 for  $(i \leftarrow 1; j \leq \phi(m)-1; i \leftarrow i+1)$  do
10   $f \leftarrow f \cdot A_f[i] \cdot \zeta_{\text{sat}(\mathcal{Q}^*)}(P)$ ,  $R \leftarrow R + A_R[i]$ 
11 end for
12 return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gauthier, Gaudreault, and Pape [34, Algorithm 2] use a version of Algorithm 3

$Q)$  with director  $(\vec{f}) = (P^2) + (Q^2) + (-1)(P^2 + Q^2) = 3(\vec{N})$ , and let  $\vec{f} = p = \lambda_{22} + r_1$  be the tangent at  $A$  with director  $(\vec{f}) = (1)(N) + (-1)(N) = 3(\vec{P})$ . The director of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathbb{R}$ .

the function  $\ell_{\text{geom}} = \ell^2$  is  $\langle \ell_{\text{geom}} \rangle = \langle \ell \rangle = \langle f \rangle = \langle f \rangle + \langle g \rangle + 2\langle N \rangle + (-1)(P^2 + Q^2) + (-1)(N) = 6\langle P \rangle$ . The director of  $\ell_{\text{geom}} = \langle \ell \rangle$  is  $(\ell_{\text{geom}}) = \langle \ell \rangle = \langle f \rangle = \langle P^2 \rangle = 3\langle N \rangle + (-1)(P^2 + Q^2) = 2\langle N \rangle + (-1)(N)$ . Notice that  $\ell_{\text{geom}}$  does not intersect  $\mathbb{R}$  at 0, projecting  $\langle \ell \rangle = \frac{1}{2} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$  gives  $\frac{1}{2} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ , which does not give rise to any area or point at  $\mathbb{R} = 0$ . Suppose we wanted to depict the function  $\ell^2$  on  $\mathbb{R}$ , and we multiplied out  $(y - \lambda_{22} - r_1)(y - \lambda_{22} - r_2)$ , substituted the  $y^2$  by  $x^2 + ax + b$  and wrote  $p = \frac{1}{2} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathbb{R}$ ) behave at points that are not on  $\mathbb{R}$ , where the substitution  $y^2 = x^2 + ax + b$  is not possible.

man in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{D}_\ell$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm, with postponed addition steps for even  $i$  and air-like pairings.

---

**Inputs:**  $\mathcal{Q}' \in \mathcal{G}_\ell$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{i-1}, m_{i-2}, \dots, m_1)$ ,  $m_{i-1} = 1$

**Outputs:**  $f_{m, \mathcal{Q}'}(P)$  representing a class in  $\mathcal{D}_\ell / \langle \mathcal{D}_\ell^2 \rangle$

```

1  $R \leftarrow \mathcal{Q}'$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2 for  $i$  from 0 to  $i-1$  do
3   if  $(m_i = 1)$  then
4      $A_0[i] \leftarrow R$ ,  $A_1[i] \leftarrow f$ ,  $j \leftarrow j+1$ 
5   end if
6    $f \leftarrow f^2 \cdot \text{pair}_{\text{air}}(P)$ ,  $R \leftarrow [2]R$ 
7 end for
8  $R \leftarrow A_0[0]$ ,  $f \leftarrow A_1[0]$ 
9 for  $(j \leftarrow 1; j \leq \theta(m) - 1; j \leftarrow +)$  do
10   $f \leftarrow f \cdot A_1[j] \cdot \text{pair}_{\text{air}}(R, A_0[j])$ ,  $R \leftarrow R + A_0[j]$ 
11 end for
12 return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gathen, Gathenstätt, and Pape [34, Algorithm 2] use a version of Algorithm 3

$$i(Q) = \theta_{ij} - pr \cdot \alpha_{i,p} \left( \frac{\theta_{ij}^2 + \delta}{\theta_{ij}^2 + r\theta_{i,p} + \delta} (\theta_{ij} - \alpha_{i,p}) + 1 \right).$$

We write this as  $\theta_{ij} = pr \cdot \alpha_{i,p}$ . The vertical line condition simply  $\alpha_{ij}^2 = \alpha_{ij} = \alpha_{i,p}$ . Multiplying all these together gives  $\beta_{i,p} = pr \cdot \alpha_{i,p} + \theta_{ij} \alpha_{i,p}$  where

$$\alpha_{i,p} = (\alpha_{i,p}^2 - (\theta_{ij}^2 + \delta \theta_{ij} + \delta)(\theta_{ij} - \alpha_{i,p}))$$

and

$$\beta_{i,p} = (\alpha_{ij}^2 + \delta \theta_{ij} + \delta \theta_{ij} \alpha_{i,p} + \alpha_{i,p}^2)(\theta_{ij} - \alpha_{i,p}).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line  $l$  is

$$y = pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (x - \alpha_{ij}) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (\theta_{ij} - \alpha_{ij}) + 1 \right).$$

Writing this as  $\theta_{ij} = pr \cdot \alpha_{i,p}$  we have  $\beta_{i+1,p} = \alpha_{i+1,p} + \theta_{ij} pr \cdot \alpha_{i+1,p}$  where

$$\alpha_{i+1,p} = (\alpha_{ij}^2 + \delta \theta_{ij} + \delta \theta_{ij} \alpha_{i,p} - (\theta_{ij}^2 + \delta \theta_{ij} + \delta)(\theta_{ij} - \alpha_{i+1,p})).$$

and

$$\beta_{i+1,p} = (\alpha_{i,p} + \beta_{i,p} \alpha_{i,p})(\theta_{ij} - \alpha_{i+1,p}).$$

$Q)$  with director  $(\vec{f}) = (P^2) + (Q^2) + (-)(P^2 + Q^2) = 3(\vec{N})$ , and let  $\vec{f} = p = \lambda_{22} + r_1$  be the tangent at  $A$  with director  $(\vec{f}) = (1)(N) + (-)(3)(N) = 3(\vec{P})$ . The director of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathbb{E}$ .

the function  $\ell_{\text{pass}} = \vec{f}^2$  is  $(\ell_{\text{pass}}) = (f) = (P^2) = (P^2) + (Q^2) + (1)(N) + (-)(P^2 + Q^2) + (-)(3)(N) = 0(\vec{P})$ . The director of  $\ell_{\text{pass}} = 0(\vec{P})$  is  $(\ell_{\text{pass}}) = (f) = (P^2) = (Q^2) + (-)(P^2 + Q^2) = (1)(N) + (-)(3)(N)$ . Notice that  $\ell_{\text{pass}}$  does not intersect  $\mathbb{E}$  at  $O$ , projecting  $0(\vec{P}) = \frac{1}{3} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$  gives  $\frac{1}{3} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ , which does not give rise to any area or point at  $\mathbb{E} = 0$ . Suppose we wanted to depict the function  $\vec{f}$  on  $\mathbb{E}$ , and we multiplied out  $(y - \lambda_{22} - r_1)(y - \lambda_{22} - r_1)$ , substituted the  $y^2$  by  $x^2 + ax + b$  and wrote  $p = \frac{1}{3} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathbb{E}$ ) behave at points that are not on  $\mathbb{E}$ , where the substitution  $y^2 = x^2 + ax + b$  is not permitted.



max in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{P}_d$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm with postponed addition steps for even  $k$  and ate-like pairings.

---

**Inputs:**  $\mathcal{Q}^* \in \mathcal{G}_2$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{d-1}, m_{d-2}, \dots, m_0)$ ,  $m_{d-1} = 1$

**Outputs:**  $f_{m, \mathcal{Q}^*}(P)$  representing a class in  $\mathcal{P}_d / \langle \mathcal{P}_d^* f \rangle$

```

1  $R \leftarrow \mathcal{Q}^*$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2 for  $i$  from 0 to  $d-1$  do
3   if  $(m_i = 1)$  then
4      $A_R[1] \leftarrow R$ ,  $A_f[1] \leftarrow f$ ,  $j \leftarrow j+1$ 
5   end if
6    $f \leftarrow f^2 \cdot \text{pair}_{\text{ate}}(P)$ ,  $R \leftarrow [2]R$ 
7 end for
8  $R \leftarrow A_R[j]$ ,  $f \leftarrow A_f[j]$ 
9 for  $(i \leftarrow 1; j \leq \phi(m)-1; i \leftarrow i+1)$  do
10   $f \leftarrow f \cdot A_f[i] \cdot \text{pair}_{\text{ate}}(A_R[i])(P)$ ,  $R \leftarrow R + A_R[i]$ 
11 end for
12 return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gauthier, Gaudreault, and Pape [34, Algorithm 2] use a version of Algorithm 3

$$i(Q) = \theta_{ij} - pr \left( \frac{\theta_{ij}^2 p + \delta}{\theta_{ij}^2 p + r\theta_{ij} + \delta} (\theta_{ij} - \theta_{ij} p) + 1 \right).$$

We write this as  $\theta_{ij} = pr\theta_{ij}$ . The vertical line condition simply  $\theta_{ij}^2 = \theta_{ij} = \theta_{ij} p$ . Multiplying all three together gives  $\theta_{ij} p = pr\theta_{ij} + \theta_{ij} p\theta_{ij}$  where

$$\theta_{ij} p = (\theta_{ij}^2 p + \delta) - (\theta_{ij}^2 + \delta \theta_{ij} + \delta \theta_{ij} p) (\theta_{ij} - \theta_{ij} p)$$

and

$$\theta_{ij} p = (\theta_{ij}^2 + \delta \theta_{ij} + \delta \theta_{ij} p) (\theta_{ij} - \theta_{ij} p) + (\theta_{ij}^2 p + \delta) (\theta_{ij} - \theta_{ij} p).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line  $l$  is

$$y = pr \left( \frac{\theta_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\theta_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

Writing this as  $\theta_{ij} + pr\theta_{ij}$  we have  $\theta_{ij+1} p = \theta_{ij+1} p + \theta_{ij} p\theta_{ij+1} p$  where

$$\theta_{ij+1} p = (\theta_{ij}^2 + \delta \theta_{ij} + \delta \theta_{ij} p) - (\theta_{ij}^2 + \delta \theta_{ij} + \delta \theta_{ij} p) (\theta_{ij} - \theta_{ij+1} p)$$

and

$$\theta_{ij+1} p = (\theta_{ij} p + \theta_{ij} p\theta_{ij}) (\theta_{ij} - \theta_{ij+1} p).$$

$Q)$  with director  $(\vec{f}) = (P^2) + (Q^2) + (-)(P^2 + Q^2) = 3(\vec{N})$ , and let  $\vec{f} = p = \lambda_{22} + r_1$  be the tangent at  $A$  with director  $(\vec{f}) = (1)(N) + (-)(3)(N) = 3(\vec{P})$ . The director of



Figure 3.5: Two functions  $f$  and  $F$  on  $\mathbb{E}$ .

the function  $\ell_{\text{pass}} = \vec{f}^2$  is  $(\ell_{\text{pass}}) = (\vec{f}) \cdot (\vec{f}) = (P^2) + (Q^2) + (1)(P^2 + Q^2) + (-)(P^2 + Q^2) = 3(\vec{N}) + 3(\vec{N}) = 6(\vec{N})$ . The director of  $\ell_{\text{pass}} = (\vec{f})^2$  is  $(\ell_{\text{pass}}) = (\vec{f}) \cdot (\vec{f}) = (P^2) + (Q^2) + (-)(P^2 + Q^2) = 3(\vec{N}) + (-)(3)(N) = 0$ . Notice that  $\ell_{\text{pass}}$  does not intersect  $\mathbb{E}$  at  $O$ ; projecting  $(\vec{f}) = \frac{1}{3} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$  gives  $\frac{1}{3} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ , which does not give rise to any area or point at  $\mathbb{E} = 0$ . Suppose we wanted to depict the function  $\vec{f}$  on  $\mathbb{E}$ , and we multiplied out  $(y - \lambda_{22} - r_1)(y - \lambda_{22} - r_1)$ , substituted the  $y^2$  by  $x^2 + ax + b$  and wrote  $p = \frac{1}{3} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ . It does not make sense to try and depict this function since all the pictures we have used for illustrative purposes also show how the functions (on  $\mathbb{E}$ ) behave at points that are not on  $\mathbb{E}$ , where the substitution  $y^2 = x^2 + ax + b$  is not possible.

man in the conventional left-to-right algorithm as it is given in Algorithm 2 on page 7. In the right-to-left version, each addition step in line 10 needs a general  $\mathcal{D}_q$ -multiplication and a multiplication with a line function value. The conventional algorithm only needs a multiplication with a line. These large costs cannot be compensated for by using affine coordinates with the inversion-sharing trick.

---

**Algorithm 3** Right-to-left version of Miller’s algorithm, with postponed addition steps for even  $i$  and air-like pairings.

---

**Inputs:**  $\mathcal{Q}' \in \mathcal{G}_2$ ,  $P \in \mathcal{G}_1$ ,  $m = (m_{i-1}, m_{i-2}, \dots, m_0)$ ,  $m_{i-1} = 1$

**Outputs:**  $f_{m, \mathcal{Q}'}(P)$  representing a class in  $\mathcal{D}_q / \langle \mathcal{D}_q^2 \rangle$

```

1  $R \leftarrow \mathcal{Q}'$ ,  $f \leftarrow 1$ ,  $j \leftarrow 0$ 
2 for  $i$  from 0 to  $i-1$  do
3   if  $(m_i = 1)$  then
4      $A_0[P] \leftarrow R$ ,  $A_j[P] \leftarrow f$ ,  $j \leftarrow j+1$ 
5   end if
6    $f \leftarrow f^2 \cdot \zeta_{m, \mathcal{Q}'}(P)$ ,  $R \leftarrow [2]R$ 
7 end for
8  $R \leftarrow A_0[3P]$ ,  $f \leftarrow A_j[3P]$ 
9 for  $(j \leftarrow 1; j \leq \theta(m) - 1; j \leftarrow j + 1)$  do
10   $f \leftarrow f \cdot A_j[P] \cdot \zeta_{m, \mathcal{Q}'}([2j]P)$ ,  $R \leftarrow R + A_0[jP]$ 
11 end for
12 return  $f$ 
```

---

### Parallelizing a single pairing

However, the right-to-left algorithm can be parallelized, and this could lead to more efficient implementations by taking advantage of many-core machines. Gathen, Gathenstätt, and Pape [34, Algorithm 2] use a version of Algorithm 3

$$i(Q) = \theta_{ij} - pr \cdot \alpha_{i,j} \left( \frac{\theta_{ij}^2 + \delta}{\theta_{ij}^2 + r\theta_{i,j} + \delta} (\theta_{ij} - \alpha_{i,j}) + 1 \right).$$

We write this as  $\theta_{ij} = pr \cdot \alpha_{i,j}$ . The vertical line condition simply  $\alpha_{ij}^2 = \alpha_{ij} = \alpha_{i,j}$ . Multiplying all these together gives  $\beta_{i,j} = pr \cdot \alpha_{i,j} + \theta_{i,j} \alpha_{i,j}$  where

$$\alpha_{i,j} = (\alpha_{i,j}^2 - (\theta_{ij}^2 + \delta \theta_{ij} + \delta \theta_{i,j})) / (\theta_{ij} - \alpha_{i,j})$$

and

$$\beta_{i,j} = (\alpha_{ij}^2 + \delta \theta_{ij} + \delta \theta_{i,j} \alpha_{i,j} + \alpha_{i,j}^2) / (\theta_{ij} - \alpha_{i,j}).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line  $l$  is

$$y = pr \left( \frac{\alpha_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

and so

$$i(Q) = \theta_{ij} - pr \left( \frac{\alpha_{ij} - 1}{\theta_{ij} - pr} (\theta_{ij} - pr) + 1 \right).$$

Writing this as  $\theta_{ij} = pr \cdot \alpha_{i,j}$  we have  $\beta_{i+1,j} = \alpha_{i+1,j} + \theta_{ij} pr \cdot \alpha_{i+1,j}$  where

$$\alpha_{i+1,j} = (\alpha_{ij}^2 + \delta \theta_{ij} + \delta \theta_{i,j} \alpha_{i,j} - (\theta_{ij}^2 + \delta \theta_{ij} + \delta \theta_{i,j} \alpha_{i,j})) / (\theta_{ij} - \alpha_{i+1,j})$$

and

$$\beta_{i+1,j} = (\alpha_{i,j} + \beta_{i,j} \alpha_{i,j}) / (\theta_{ij} - \alpha_{i+1,j}).$$



## core idea

For  $P \in E(\mathbb{F}_p)$  and  $Q \in E^t(\mathbb{F}_p)$ ,  
don't use curve arithmetic  
but pairing  $e(P, Q)$  to get  
overlap in orders!







Better suited for papers than slides

Computing pairs fast is quite technical.



# generations





Instead I describe the general approach,

and available details out

general approach







1

implemental tricks

that apply



2

benchmark speed



and fine tune



3

fast pairings



0

take someone's literature

$$i(Q) = \theta_{ij} - pr \cdot \alpha_{i,p} \left( \frac{\theta_{ij}^2 + \delta}{\theta_{ij}^2 + r\theta_{i,p} + \delta} (\theta_{ij} - \alpha_{i,p}) + 1 \right).$$

We write this as  $\theta_{ij} = pr \cdot \alpha_{i,p}$ . The vertical line condition simply  $\alpha_{ij}^2 = \alpha_{ij} = \alpha_{i,p}$ . Multiplying all these together gives  $\beta_{i,p} = pr \cdot \alpha_{i,p} + \theta_{ij} \alpha_{i,p}$  where

$$\alpha_{i,p} = (\alpha_{i,p}^2 - (\theta_{ij}^2 + \delta \theta_{ij} + \delta \theta_{i,p})) / (\theta_{ij} - \alpha_{i,p})$$

and

$$\beta_{i,p} = (\alpha_{ij}^2 + \delta \theta_{ij} + \delta \theta_{i,p} \theta_{i,p} + \alpha_{i,p}^2) / (\theta_{ij} - \alpha_{i,p}).$$

This completes proof of first part of the first claim.

Now suppose a further addition is performed in Miller's algorithm. It is known that the final addition does not affect the form of the value. In general case, from Lemma 2 we deduce that the line  $l$  is

$$y = pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (x - \alpha_{ij}) + 1 \right).$$

and so

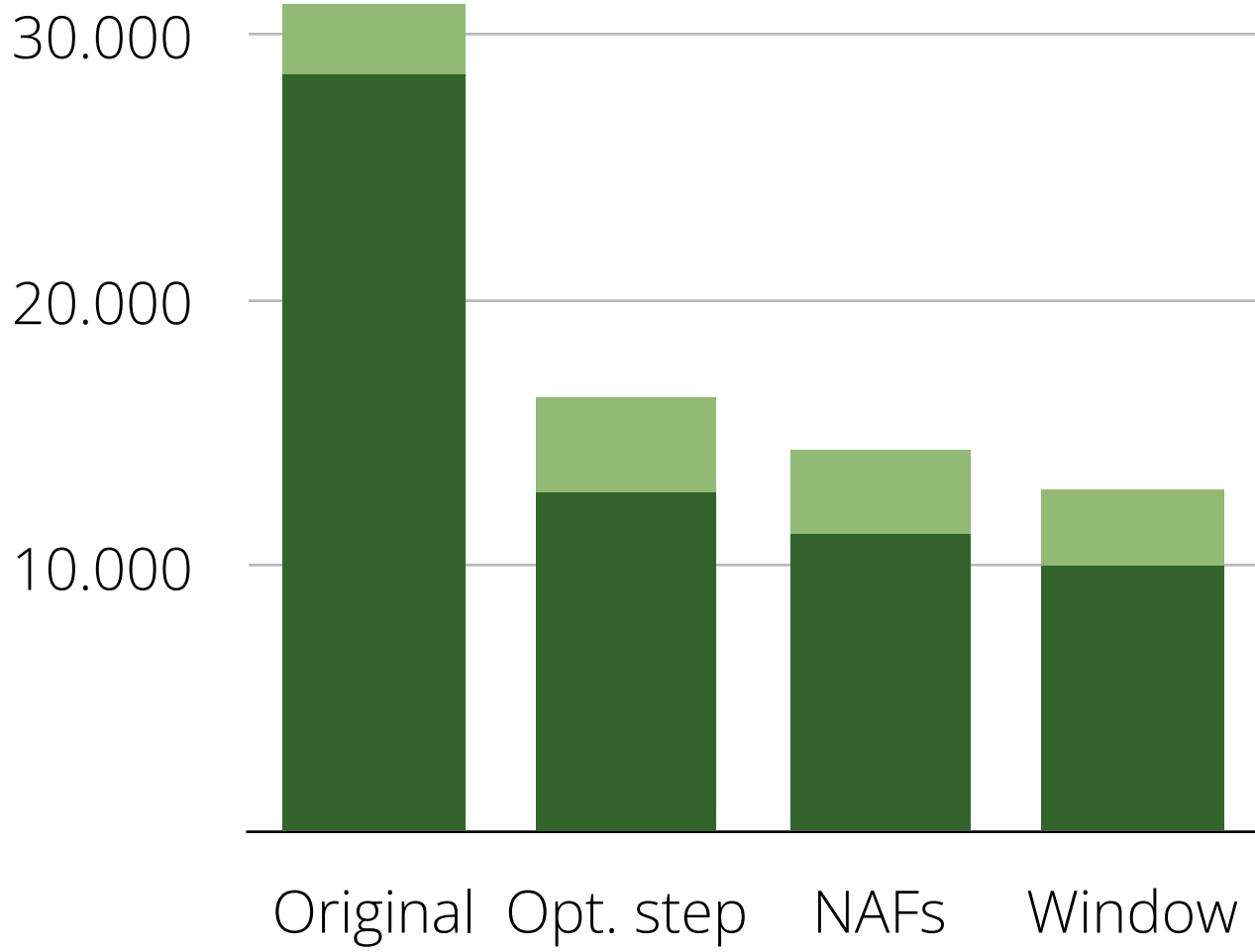
$$i(Q) = \theta_{ij} - pr \left( \frac{\alpha_{ij} - 1}{\alpha_{ij} - pr} (\theta_{ij} - \alpha_{ij}) + 1 \right).$$

Writing this as  $\theta_{ij} = pr \cdot \alpha_{i,p}$  we have  $\beta_{i+1,p} = \alpha_{i+1,p} + \theta_{ij} pr \cdot \alpha_{i+1,p}$  where

$$\alpha_{i+1,p} = (\alpha_{ij}^2 + \delta \theta_{ij} + \delta \theta_{i,p} \theta_{i,p} - (\theta_{ij}^2 + \delta \theta_{ij} + \delta \theta_{i,p})) / (\theta_{ij} - \alpha_{i+1,p})$$

and

$$\beta_{i+1,p} = (\alpha_{i,p} + \beta_{i,p} \theta_{ij}) / (\theta_{ij} - \alpha_{i+1,p}).$$





3

fast pairings