# Algorithme(s) de type Chudnovsky-Chudnovsky pour la multiplication dans les extensions finies de $\mathbb{F}_q$

**Julia Pieltant**
Inria Saclay - Île-de-France

Journées C2
25 mars 2014

# Plan

1. Introduction — Position du problème
   - Différentes notions de complexité
   - Lien avec le problème de la détermination du rang de tenseur
   - Complexité bilinéaire symétrique et asymétrique

2. Vers l'algorithme de Chudnovsky-Chudnovsky...
   - Quelques bornes de complexité bilinéaire
   - Algorithmes de type évalutation-interpolation

3. Algorithme de Chudnovsky-Chudnovsky
   - Principe général
   - Application
   - Résultat principal
   - Bornes théoriques

**Complexité de la multiplication dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$ :**

Nombre minimal d'opérations élémentaires dans $\mathbb{F}_q$ nécessaires pour calculer le produit de deux éléments quelconques $x, y \in \mathbb{F}_{q^n}$.

**Complexité de la multiplication dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$ :**

Nombre minimal d'opérations élémentaires dans $\mathbb{F}_q$ nécessaires pour calculer le produit de deux éléments quelconques $x, y \in \mathbb{F}_{q^n}$.

**Types d'opérations :**

- addition : $(\alpha, \beta) \mapsto \alpha + \beta$ où $\alpha, \beta \in \mathbb{F}_q$,
- multiplication scalaire : $x_i \mapsto \alpha \cdot x_i$ où $\alpha, x_i \in \mathbb{F}_q$, et $\alpha$ est une constante,
- multiplication non-scalaire ou bilinéaire : $(x_i, y_j) \mapsto x_i \cdot y_j$ où $x_i, y_j \in \mathbb{F}_q$ dépendent des éléments $x$ et $y$ de $\mathbb{F}_{q^n}$ dont on effectue le produit.

**Complexité de la multiplication dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$ :**

Nombre minimal d'opérations élémentaires dans $\mathbb{F}_q$ nécessaires pour calculer le produit de deux éléments quelconques $x, y \in \mathbb{F}_{q^n}$.

**Types d'opérations :**

- addition : $(\alpha, \beta) \mapsto \alpha + \beta$ où $\alpha, \beta \in \mathbb{F}_q$,
- multiplication scalaire : $x_i \mapsto \alpha \cdot x_i$ où $\alpha, x_i \in \mathbb{F}_q$, et $\alpha$ est une constante,
- multiplication non-scalaire ou bilinéaire : $(x_i, y_j) \mapsto x_i \cdot y_j$ où $x_i, y_j \in \mathbb{F}_q$ dépendent des éléments $x$ et $y$ de $\mathbb{F}_{q^n}$ dont on effectue le produit.

Le nombre minimal de multiplications bilinéaires nécessaires pour effectuer le produit de deux éléments quelconques de $\mathbb{F}_{q^n}$ est appelé **complexité bilinéaire de la multiplication dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$**, et notée $\mu_q(n)$.

# Analyse des opérations d'un produit

Soit $\mathcal{B} := (e_1, \ldots, e_n)$ une base de $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$.

On définit

$$e_i e_j := \sum_{k=1}^{n} \alpha_{i,j,k} e_k \text{ pour tous } i, j \in \{1, \ldots, n\}.$$

Soient $x = \sum_{i=1}^{n} x_i e_i$ et $y = \sum_{i=1}^{n} y_i e_i$ deux éléments de $\mathbb{F}_{q^n}$, on a

$$xy = \sum_{k=1}^{n} \left( \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{i,j,k} \cdot x_i \cdot y_j \right) e_k.$$

# Analyse des opérations d'un produit

Soit $\mathcal{B} := (e_1, \ldots, e_n)$ une base de $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$.

On définit

$$e_i e_j := \sum_{k=1}^{n} \alpha_{i,j,k} e_k \text{ pour tous } i, j \in \{1, \ldots, n\}.$$

Soient $x = \sum_{i=1}^{n} x_i e_i$ et $y = \sum_{i=1}^{n} y_i e_i$ deux éléments de $\mathbb{F}_{q^n}$, on a

$$xy = \sum_{k=1}^{n} \left( \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{i,j,k} \cdot x_i \cdot y_j \right) e_k.$$

**Nombre d'opérations :**

- $n^2$ multiplications bilinéaires,
- $n^3$ multiplications scalaires,
- $n(n-1)(n+1)$ additions d'éléments de $\mathbb{F}_q$.

# Analyse des opérations d'un produit

Soit $\mathcal{B} := (e_1, \ldots, e_n)$ une base de $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$.

On définit

$$e_i e_j := \sum_{k=1}^{n} \alpha_{i,j,k} e_k \text{ pour tous } i, j \in \{1, \ldots, n\}.$$

Soient $x = \sum_{i=1}^{n} x_i e_i$ et $y = \sum_{i=1}^{n} y_i e_i$ deux éléments de $\mathbb{F}_{q^n}$, on a

$$xy = \sum_{k=1}^{n} \left( \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{i,j,k} \cdot x_i \cdot y_j \right) e_k.$$

**Nombre d'opérations :**

- $n^2$ multiplications bilinéaires,
- $n^3$ multiplications scalaires,
- $n(n-1)(n+1)$ additions d'éléments de $\mathbb{F}_q$.

Soit $t_m$ le **tenseur de la multiplication** dans $\mathbb{F}_{q^n}$ : $\forall\, x, y \in \mathbb{F}_{q^n},\ t_m(x \otimes y) = xy$.

On considère une **décomposition de $t_m$ en $\lambda$ tenseurs élémentaires**, c-à-d on considère $a_i, b_i \in \mathbb{F}_{q^n}^{\star}$ et $c_i \in \mathbb{F}_{q^n}$ tels que tous $x, y \in \mathbb{F}_{q^n}$, on a

$$xy = t_m(x \otimes y) = \sum_{i=1}^{\lambda} a_i(x) b_i(y) c_i. \tag{1}$$

Toute expression de type (1) est appelée **algorithme de multiplication bilinéaire $\mathscr{U}$**.

Sa **complexité** $\lambda$ est notée $\mu(\mathscr{U})$.

Ainsi

$$\mu_q(n) = \min_{\mathscr{U}} \mu(\mathscr{U})$$

où $\mathscr{U}$ parcourt l'ensemble des algorithmes de multiplication bilinéaire dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$.

# Complexité bilinéaire symétrique

**Definition**

Un algorithme de multiplication bilinéaire est dit **symétrique** s'il admet une expression de la forme :

$$xy = \sum_{i=1}^{\lambda} a_i(x)a_i(y)c_i. \tag{2}$$

pour tous $x, y \in \mathbb{F}_{q^n}$, avec $a_i \in \mathbb{F}_{q^n}^{\star}$ et $c_i \in \mathbb{F}_{q^n}$

On définit alors la **complexité bilinéaire symétrique** de la multiplication dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$ en posant :

$$\mu_q^{\text{sym}}(n) := \min_{\mathscr{U}^{\text{sym}}} \mu(\mathscr{U}^{\text{sym}})$$

où $\mathscr{U}^{\text{sym}}$ parcourt l'ensemble des algorithmes symétriques de multiplication bilinéaire dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$.

**Rq.** $\mu_q(n) \leqslant \mu_q^{\text{sym}}(n)$

# Le cas des extensions de «petit» degré

**Théorème (S. Winograd et H.F. de Groote (1979, 1983))**

*La complexité bilinéaire de la multiplication dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$ vérifie*

$$\mu_q(n) \geqslant 2n-1.$$

*De plus,*

$$\mu_q^{\text{sym}}(n) = 2n-1 \iff n \leqslant \frac{q}{2}+1.$$

**Théorème (Shokrollahi (1992))**

*Si* $n \leqslant \frac{1}{2}(q+1+\epsilon(q))$, *alors*

$$\mu_q^{\text{sym}}(n) \leqslant 2n$$

*où* $\epsilon(q) = \begin{cases} 2\sqrt{q} \text{ si } q \text{ est un carré parfait,} \\ \text{le plus grand entier plus petit que } 2\sqrt{q} \text{ premier à } q \text{ sinon.} \end{cases}$

# Lien avec les codes linéaires

- À tout algorithme de multiplication bilinéaire symétrique de complexité $\lambda$ dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$, on peut associer un code linéaire $C\ [\lambda, n, \geqslant n]_q$ :

[Shparlinski, Tsfasman, Vlăduț, *Curves with many points and multiplication in finite fields*, AGCT-91]

# Lien avec les codes linéaires

- À tout algorithme de multiplication bilinéaire symétrique de complexité $\lambda$ dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$, on peut associer un code linéaire $C\ [\lambda, n, \geqslant n]_q$ :

  Soient $a_i \in \mathbb{F}_{q^n}^\star$, $c_i \in \mathbb{F}_{q^n}$ t.q. pour tous $x, y \in \mathbb{F}_{q^n}$,

$$xy = \sum_{i=1}^{\lambda} a_i(x) a_i(y) c_i$$

  On définit le code $C$ par

$$C = \operatorname{Im} \phi_a$$

  avec

$$\begin{array}{rccc} \phi_a : & \mathbb{F}_{q^n} & \longrightarrow & \mathbb{F}_q^\lambda \\ & x & \longmapsto & \big(a_1(x), \ldots, a_\lambda(x)\big) \end{array}$$

[Shparlinski, Tsfasman, Vlăduț, *Curves with many points and multiplication in finite fields*, AGCT-91]

# Lien avec les codes linéaires

- À tout algorithme de multiplication bilinéaire symétrique de complexité $\lambda$ dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$, on peut associer un code linéaire $C\ [\lambda, n, \geqslant n]_q$ :

  Soient $a_i \in \mathbb{F}_{q^n}^\star$, $c_i \in \mathbb{F}_{q^n}$ t.q. pour tous $x, y \in \mathbb{F}_{q^n}$,

  $$xy = \sum_{i=1}^{\lambda} a_i(x) a_i(y) c_i$$

  On définit le code $C$ par

  $$C = \operatorname{Im} \phi_a$$

  avec

  $$\begin{array}{rccc} \phi_a : & \mathbb{F}_{q^n} & \longrightarrow & \mathbb{F}_q^\lambda \\ & x & \longmapsto & \big(a_1(x), \ldots, a_\lambda(x)\big) \end{array}$$

- Inversement, les «supercodes exacts» sont des codes $[\lambda, n]_q$ en **bijection** avec les algorithmes de multiplication bilinéaire symétriques de complexité $\lambda$.

[Shparlinski, Tsfasman, Vlăduț, *Curves with many points and multiplication in finite fields*, AGCT-91]

# Borne inférieure

$$\mu_q(n) \geqslant 2n - 1$$

# Borne inférieure

$$\mu_q(n) \geqslant 2n - 1$$

**Preuve.**
Soit un algorithme de complexité bilinéaire $\lambda := \mu_q(n)$,
c-à-d soient $a_i, b_i \in \mathbb{F}_{q^n}^\star$, $c_i \in \mathbb{F}_{q^n}$ t.q.

$$\forall\ x, y \in \mathbb{F}_{q^n}, \qquad xy = \sum_{i=1}^{\lambda} a_i(x) b_i(y) c_i.$$

# Borne inférieure

$$\mu_q(n) \geqslant 2n - 1$$

**Preuve.**
Soit un algorithme de complexité bilinéaire $\lambda := \mu_q(n)$,
c-à-d soient $a_i, b_i \in \mathbb{F}_{q^n}^\star$, $c_i \in \mathbb{F}_{q^n}$ t.q.

$$\forall\ x, y \in \mathbb{F}_{q^n}, \qquad xy = \sum_{i=1}^{\lambda} a_i(x) b_i(y) c_i.$$

Soient $C_a := \operatorname{Im} \phi_a$ et $C_b := \operatorname{Im} \phi_b$, où :

$$\begin{array}{rccc} \phi_a : & \mathbb{F}_{q^n} & \longrightarrow & \mathbb{F}_q^\lambda \\ & x & \longmapsto & \big(a_1(x), \ldots, a_\lambda(x)\big) \end{array} \qquad \begin{array}{rccc} \phi_b : & \mathbb{F}_{q^n} & \longrightarrow & \mathbb{F}_q^\lambda \\ & y & \longmapsto & \big(b_1(y), \ldots, b_\lambda(y)\big) \end{array}$$

# Borne inférieure

$$\mu_q(n) \geqslant 2n - 1$$

**Preuve.**
Soit un algorithme de complexité bilinéaire $\lambda := \mu_q(n)$,
c-à-d soient $a_i, b_i \in \mathbb{F}_{q^n}^\star$, $c_i \in \mathbb{F}_{q^n}$ t.q.

$$\forall\ x, y \in \mathbb{F}_{q^n}, \qquad xy = \sum_{i=1}^{\lambda} a_i(x) b_i(y) c_i.$$

Soient $C_a := \operatorname{Im} \phi_a$ et $C_b := \operatorname{Im} \phi_b$, où :

$$\begin{array}{rccc} \phi_a : & \mathbb{F}_{q^n} & \longrightarrow & \mathbb{F}_q^\lambda \\ & x & \longmapsto & \big(a_1(x), \ldots, a_\lambda(x)\big) \end{array} \qquad \begin{array}{rccc} \phi_b : & \mathbb{F}_{q^n} & \longrightarrow & \mathbb{F}_q^\lambda \\ & y & \longmapsto & \big(b_1(y), \ldots, b_\lambda(y)\big) \end{array}$$

- $\phi_a$ et $\phi_b$ sont linéaires et injectives, donc $C_a$ et $C_b$ sont des codes linéaires $[\lambda, n]_q$,

# Borne inférieure

$$\mu_q(n) \geqslant 2n - 1$$

**Preuve.**
Soit un algorithme de complexité bilinéaire $\lambda := \mu_q(n)$,
c-à-d soient $a_i, b_i \in \mathbb{F}_{q^n}^\star$, $c_i \in \mathbb{F}_{q^n}$ t.q.

$$\forall\ x, y \in \mathbb{F}_{q^n}, \qquad xy = \sum_{i=1}^{\lambda} a_i(x) b_i(y) c_i.$$

Soient $C_a := \operatorname{Im} \phi_a$ et $C_b := \operatorname{Im} \phi_b$, où :

$$\begin{array}{rccc} \phi_a : & \mathbb{F}_{q^n} & \longrightarrow & \mathbb{F}_q^\lambda \\ & x & \longmapsto & \big(a_1(x), \ldots, a_\lambda(x)\big) \end{array} \qquad \begin{array}{rccc} \phi_b : & \mathbb{F}_{q^n} & \longrightarrow & \mathbb{F}_q^\lambda \\ & y & \longmapsto & \big(b_1(y), \ldots, b_\lambda(y)\big) \end{array}$$

- $\phi_a$ et $\phi_b$ sont linéaires et injectives, donc $C_a$ et $C_b$ sont des codes linéaires $[\lambda, n]_q$,
- $C_a$ et $C_b$ sont *mutuellement intersectants* : si $(c, \tilde{c}) \in C_a \times C_b$ sont à supports disjoints, alors $c = 0$ ou $\tilde{c} = 0$ ; sinon $\exists\, x, y \in \mathbb{F}_{q^n}$, t.q. $xy = 0$.

# Borne inférieure

$$\mu_q(n) \geqslant 2n-1$$

**Preuve.**
Soit un algorithme de complexité bilinéaire $\lambda := \mu_q(n)$,
c-à-d soient $a_i, b_i \in \mathbb{F}_{q^n}^\star$, $c_i \in \mathbb{F}_{q^n}$ t.q.

$$\forall\ x, y \in \mathbb{F}_{q^n}, \qquad xy = \sum_{i=1}^{\lambda} a_i(x) b_i(y) c_i.$$

Soient $C_a := \operatorname{Im} \phi_a$ et $C_b := \operatorname{Im} \phi_b$, où :

$$\begin{array}{rccc} \phi_a : & \mathbb{F}_{q^n} & \longrightarrow & \mathbb{F}_q^\lambda \\ & x & \longmapsto & (a_1(x), \ldots, a_\lambda(x)) \end{array} \qquad \begin{array}{rccc} \phi_b : & \mathbb{F}_{q^n} & \longrightarrow & \mathbb{F}_q^\lambda \\ & y & \longmapsto & (b_1(y), \ldots, b_\lambda(y)) \end{array}$$

- $\phi_a$ et $\phi_b$ sont linéaires et injectives, donc $C_a$ et $C_b$ sont des codes linéaires $[\lambda, n]_q$,
- $C_a$ et $C_b$ sont *mutuellement intersectants* : si $(c, \tilde{c}) \in C_a \times C_b$ sont à supports disjoints, alors $c = 0$ ou $\tilde{c} = 0$ ; sinon $\exists\, x, y \in \mathbb{F}_{q^n}$, t.q. $xy = 0$.

**Conséquence.**

$$d_{\min}(C_a) > \left\lceil \frac{\lambda}{2} \right\rceil \qquad \text{ou} \qquad d_{\min}(C_b) > \left\lceil \frac{\lambda}{2} \right\rceil$$

$$\text{donc} \qquad \left\lceil \frac{\lambda}{2} \right\rceil < \lambda - n + 1 \qquad \text{(borne de singleton)}$$

# Algorithme de Karatsuba

**Multiplication de deux polynômes de degré 1, à coefficients dans un corps $F$ :**

$$U(X) = aX + b \qquad V(X) = cX + d$$

**But.** Déterminer les coefficients $p_0, p_1, p_2 \in F$ du produit $U(X) \cdot V(X) = p_2X^2 + p_1X + p_0$.

# Algorithme de Karatsuba

**Multiplication de deux polynômes de degré 1, à coefficients dans un corps $F$ :**

$$U(X) = aX + b \qquad V(X) = cX + d$$

**But.** Déterminer les coefficients $p_0, p_1, p_2 \in F$ du produit $U(X) \cdot V(X) = p_2X^2 + p_1X + p_0$.

Évaluations en $0, 1, \infty$ :

$$\begin{array}{lll} U(0) = b & U(1) = a + b & U(\infty) = a \\ V(0) = d & V(1) = c + d & V(\infty) = c \end{array}$$

# Algorithme de Karatsuba

**Multiplication de deux polynômes de degré 1, à coefficients dans un corps $F$ :**

$$U(X) = aX + b \qquad V(X) = cX + d$$

**But.** Déterminer les coefficients $p_0, p_1, p_2 \in F$ du produit $U(X) \cdot V(X) = p_2X^2 + p_1X + p_0$.

Évaluations en $0, 1, \infty$ :

$$\begin{array}{lll} U(0) = b & U(1) = a + b & U(\infty) = a \\ V(0) = d & V(1) = c + d & V(\infty) = c \end{array}$$

On calcule alors

$$p_0 = U(0)V(0), \qquad p_2 = U(\infty)V(\infty), \qquad p_1 = U(1)V(1) - p_0 - p_2.$$

**Complexité :** 3 multiplications bilinéaires et 4 additions.

# Algorithme de Karatsuba

**Multiplication de deux polynômes de degré 1, à coefficients dans un corps $F$ :**

$$U(X) = aX + b \qquad V(X) = cX + d$$

**But.** Déterminer les coefficients $p_0, p_1, p_2 \in F$ du produit $U(X) \cdot V(X) = p_2X^2 + p_1X + p_0$.

Évaluations en $0, 1, \infty$ :

$$\begin{array}{lllllllll} U(0) & = & b & & U(1) & = & a+b & & U(\infty) = a \\ V(0) & = & d & & V(1) & = & c+d & & V(\infty) = c \end{array}$$

On calcule alors

$$p_0 = U(0)V(0), \qquad p_2 = U(\infty)V(\infty), \qquad p_1 = U(1)V(1) - p_0 - p_2.$$

**Complexité :** 3 multiplications bilinéaires et 4 additions.

**Généralisation au produit de deux polynômes de degré $n$ :**

**Complexité :** $O\left(n^{\log_2(3)}\right)$ multiplications bilinéaires et $O(n)$ additions.

# Un premier pas vers l'algorithme de Chudnovsky-Chudnovsky

Algorithme de Karatsuba pour les polynômes de degré 1 $\leftrightsquigarrow$ évaluations sur $\{0, 1, \infty\} \in \mathrm{P}^1(F)$

# Un premier pas vers l'algorithme de Chudnovsky-Chudnovsky

Algorithme de Karatsuba pour les polynômes de degré 1 $\leftrightsquigarrow$ évaluations sur $\{0, 1, \infty\} \in \mathrm{P}^1(F)$

**Rappel.**

- On note $\mathrm{P}^1(\overline{\mathbb{F}_q})$ la **droite projective** sur $\overline{\mathbb{F}_q}$ :

$$\mathrm{P}^1(\overline{\mathbb{F}_q}) := \left\{(x, y) \in \mathbb{A}^2(\overline{\mathbb{F}_q}) \backslash \{(0,0)\}\right\} / \sim$$

  où $\sim$ est la relation d'équivalence définie par la colinéarité.

- C'est une courbe algébrique de genre 0 avec $q + 1$ points rationnels :

$$\mathrm{P}^1(\mathbb{F}_q) = \left\{(x : 1) \;\middle|\; x \in \mathbb{F}_q\right\} \cup \{\infty\}.$$

# Un premier pas vers l'algorithme de Chudnovsky-Chudnovsky

Algorithme de Karatsuba pour les polynômes de degré 1 $\leftrightsquigarrow$ évaluations sur $\{0, 1, \infty\} \in \mathrm{P}^1(F)$

**Rappel.**

- On note $\mathrm{P}^1(\overline{\mathbb{F}_q})$ la **droite projective** sur $\overline{\mathbb{F}_q}$ :

$$\mathrm{P}^1(\overline{\mathbb{F}_q}) := \left\{(x, y) \in \mathbb{A}^2(\overline{\mathbb{F}_q}) \backslash \{(0,0)\}\right\} / \sim$$

  où $\sim$ est la relation d'équivalence définie par la colinéarité.

- C'est une courbe algébrique de genre 0 avec $q + 1$ points rationnels :

$$\mathrm{P}^1(\mathbb{F}_q) = \left\{(x : 1) \;\middle|\; x \in \mathbb{F}_q\right\} \cup \{\infty\}.$$

**Idée.** **Généraliser** la méthode de Karatsuba en faisant **plus d'évaluations** sur $\mathrm{P}^1(F)$ lorsque $|F| > 2$.

# Multiplication de polynômes et multiplication dans $\mathbb{F}_{q^n}$ (I)

Soit $P(X) \in \mathbb{F}_q[X]$ un polynôme unitaire de degré $n$, irréductible sur $\mathbb{F}_q$.

Alors

$$\mathbb{F}_{q^n} \simeq \mathbb{F}_q[X]/(P(X))$$

et si $\alpha$ est une racine de $P(X)$, alors $\mathcal{B} = (1, \alpha, \alpha^2, \ldots, \alpha^{n-1})$ est une $\mathbb{F}_q$-base de $\mathbb{F}_{q^n}$.

# Multiplication de polynômes et multiplication dans $\mathbb{F}_{q^n}$ (I)

Soit $P(X) \in \mathbb{F}_q[X]$ un polynôme unitaire de degré $n$, irréductible sur $\mathbb{F}_q$.

Alors

$$\mathbb{F}_{q^n} \simeq \mathbb{F}_q[X]/(P(X))$$

et si $\alpha$ est une racine de $P(X)$, alors $\mathcal{B} = (1, \alpha, \alpha^2, \ldots, \alpha^{n-1})$ est une $\mathbb{F}_q$-base de $\mathbb{F}_{q^n}$.

Pour connaître le produit deux éléments $x, y \in \mathbb{F}_{q^n}$ t.q.

$$x = \sum_{i=0}^{n-1} a_i \alpha^i \qquad \text{et} \qquad y = \sum_{i=0}^{n-1} b_i \alpha^i$$

il suffit de déterminer les coefficients du produit des deux polynômes

$$A(X) = \sum_{i=0}^{n-1} a_i X^i \qquad \text{et} \qquad B(X) = \sum_{i=0}^{n-1} b_i X^i$$

car

$$xy = A(\alpha) \cdot B(\alpha) = (AB)(\alpha).$$

# Multiplication de polynômes et multiplication dans $\mathbb{F}_{q^n}$ (II)

Supposons que $\#\mathrm{P}^1(\mathbb{F}_q) \geqslant \deg A(X) \cdot B(X) + 1$, i.e. $q + 1 \geqslant 2n - 1$.

On choisit $S \subseteq \mathrm{P}^1(\mathbb{F}_q)$ de cardinal $2n - 1$.

# Multiplication de polynômes et multiplication dans $\mathbb{F}_{q^n}$ (II)

Supposons que $\#\mathrm{P}^1(\mathbb{F}_q) \geqslant \deg A(X) \cdot B(X) + 1$, i.e. $q + 1 \geqslant 2n - 1$.

On choisit $S \subseteq \mathrm{P}^1(\mathbb{F}_q)$ de cardinal $2n - 1$.

1. On détermine les évaluations $A(w)$ et $B(w)$ en tous les points $w$ de $S$.
2. On calcule $(AB)(w) = A(w) \cdot B(w)$, pour tout $w \in S$.
3. Par interpolation, on retrouve les coefficients de $(AB)(X)$, et donc le produit $xy = (AB)(\alpha)$ dans la base $\mathcal{B}$.

# Multiplication de polynômes et multiplication dans $\mathbb{F}_{q^n}$ (II)

Supposons que $\#\mathrm{P}^1(\mathbb{F}_q) \geqslant \deg A(X) \cdot B(X) + 1$, i.e. $q + 1 \geqslant 2n - 1$.

On choisit $S \subseteq \mathrm{P}^1(\mathbb{F}_q)$ de cardinal $2n - 1$.

1. On détermine les évaluations $A(w)$ et $B(w)$ en tous les points $w$ de $S$.
2. On calcule $(AB)(w) = A(w) \cdot B(w)$, pour tout $w \in S$.
3. Par interpolation, on retrouve les coefficients de $(AB)(X)$, et donc le produit $xy = (AB)(\alpha)$ dans la base $\mathcal{B}$.

**Complexité bilinéaire :** $\#S = 2n - 1$ multiplications.

**Conséquence.** Si $n \leqslant \dfrac{q}{2} + 1$, alors $\mu_q(n) \leqslant 2n - 1$.

# Rappel

**Théorème (S. Winograd et H.F. de Groote (1979, 1983))**

*La complexité bilinéaire de la multiplication dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$ vérifie*

$$\mu_q(n) \geqslant 2n - 1.$$

*De plus,*

$$\mu_q^{\text{sym}}(n) = 2n - 1 \iff n \leqslant \frac{q}{2} + 1.$$

**Théorème (Shokrollahi (1992))**

*Si $n \leqslant \frac{1}{2}(q + 1 + \epsilon(q))$, alors*

$$\mu_q^{\text{sym}}(n) \leqslant 2n$$

*où* $\epsilon(q) = \begin{cases} 2\sqrt{q} \text{ si } q \text{ est un carré parfait,} \\ \text{le plus grand entier plus petit que } 2\sqrt{q} \text{ premier à } q \text{ sinon.} \end{cases}$

# Rappel

**Théorème (S. Winograd et H.F. de Groote (1979, 1983))**

*La complexité bilinéaire de la multiplication dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$ vérifie*

$$\mu_q(n) \geqslant 2n-1.$$

*De plus,*

$$\mu_q^{\text{sym}}(n) = 2n-1 \iff n \leqslant \frac{q}{2}+1.$$

**Théorème (Shokrollahi (1992))**

*Si* $n \leqslant \frac{1}{2}(q+1+\epsilon(q))$, *alors*

$$\mu_q^{\text{sym}}(n) \leqslant 2n$$

*où* $\epsilon(q) = \begin{cases} 2\sqrt{q} \text{ si } q \text{ est un carré parfait,} \\ \text{le plus grand entier plus petit que } 2\sqrt{q} \text{ premier à } q \text{ sinon.} \end{cases}$

# Correspondance entre courbes et corps de fonctions

| **Courbe** | **Corps de fonctions** |
|---|---|
| $\mathrm{P}^1(\mathbb{F}_q)$<br>droite projective | $\mathbb{F}_q(x)$, $x$ transcendant sur $\mathbb{F}_q$<br>corps des fonctions rationnelles |
| $C : f(x, y) = 0$<br>courbe algébrique définie sur $\mathbb{F}_q$ | $\mathbf{F} = \mathbb{F}_q(x)(y)$ où $f(x, y) = 0$<br>avec $f(x, Y) \in \mathbb{F}_q(x)[Y]$ irréductible |

**Exemple.**
Courbe elliptique sur $\mathbb{F}_3$ :

$$E \ : \quad y^2 = x^3 + x^2 + 1$$

Corps de fonctions elliptiques associé :

$$\mathbf{F} = \mathbb{F}_3(x, y) = \mathbb{F}_3(x)(y)$$

où y est racine de $Y^2 - x^3 - x^2 - 1$.

# Évaluation sur de bonnes courbes elliptiques

**Théorème (Waterhouse (1969))**

*Pour tout $q$ puissance d'un premier, il existe un corps de fonctions elliptiques $\mathbf{F}/\mathbb{F}_q$ avec $q + 1 + \epsilon(q)$ places rationnelles.*

# Évaluation sur de bonnes courbes elliptiques

**Théorème (Waterhouse (1969))**

*Pour tout $q$ puissance d'un premier, il existe un corps de fonctions elliptiques $\mathbf{F}/\mathbb{F}_q$ avec $q+1+\epsilon(q)$ places rationnelles.*

**Évaluation-interpolation sur des courbes elliptiques :**

- droite projective sur $\mathbb{F}_q$ $\rightsquigarrow$ courbe elliptique,
- évaluations sur les $q+1+\epsilon(q)$ points rationnels,
- $\mathbb{F}_{q^n}$ est identifié à $\mathcal{L}$, un espace de fonctions qui est un $\mathbb{F}_q$-ev de dimension $n$,
- si $u, v \in \mathcal{L}$ alors $u \cdot v \in \mathcal{L}^2$ où $\dim \mathcal{L}^2 = 2 \dim \mathcal{L}$.

Complexité bilinéaire : $2n$.

**Conclusion.** Si $2n \leqslant q+1+\epsilon(q)$, alors $\mu_q(n) \leqslant 2n$.

# Évaluation sur de bonnes courbes elliptiques

**Théorème (Waterhouse (1969))**

*Pour tout $q$ puissance d'un premier, il existe un corps de fonctions elliptiques $\mathbf{F}/\mathbb{F}_q$ avec $q+1+\epsilon(q)$ places rationnelles.*

**Évaluation-interpolation sur des courbes elliptiques :**

- droite projective sur $\mathbb{F}_q$ $\rightsquigarrow$ courbe elliptique,
- évaluations sur les $q+1+\epsilon(q)$ points rationnels,
- $\mathbb{F}_{q^n}$ est identifié à $\mathcal{L}$, un espace de fonctions qui est un $\mathbb{F}_q$-ev de dimension $n$,
- si $u, v \in \mathcal{L}$ alors $u \cdot v \in \mathcal{L}^2$ où $\dim \mathcal{L}^2 = 2 \dim \mathcal{L}$.

Complexité bilinéaire : $2n$.

**Conclusion.** Si $2n \leqslant q+1+\epsilon(q)$, alors $\mu_q(n) \leqslant 2n$.

**N.B.** C'est une application de l'algorithme de Chudnovsky-Chudnovsky.

# Algorithme de multiplication dans $\mathbb{F}_{q^n}$

Soit $\mathbf{F}/\mathbb{F}_q$ un corps de fonctions algébriques défini sur $\mathbb{F}_q$ de genre $g$ pour lequel on a

- $Q$ une place de degré $n$,
- $\mathscr{P} := \{P_1, \ldots, P_N\}$ un ensemble de N places rationnelles,
- $\mathcal{D}$ un diviseur effectif tel que $\mathrm{supp}(\mathcal{D}) \cap \{Q, P_1, \ldots, P_N\} = \varnothing$.

# Algorithme de multiplication dans $\mathbb{F}_{q^n}$

Soit $\mathbf{F}/\mathbb{F}_q$ un corps de fonctions algébriques défini sur $\mathbb{F}_q$ de genre $g$ pour lequel on a

- $Q$ une place de degré $n$,
- $\mathscr{P} := \{P_1, \ldots, P_N\}$ un ensemble de N places rationnelles,
- $\mathcal{D}$ un diviseur effectif tel que $\mathrm{supp}(\mathcal{D}) \cap \{Q, P_1, \ldots, P_N\} = \varnothing$.

Si on a

(i) un morphisme de $\mathbb{F}_q$-espaces vectoriels **surjectif**

$$\begin{array}{rccc} \mathrm{Ev}_Q : & \mathscr{L}(\mathcal{D}) & \longrightarrow & F_Q \simeq \mathbb{F}_{q^n} \\ & f & \longmapsto & f(Q) \end{array}$$

(ii) un morphisme de $\mathbb{F}_q$-espaces vectoriels **injectif**

$$\begin{array}{rccc} \mathrm{Ev}_{\mathscr{P}} : & \mathscr{L}(2\mathcal{D}) & \longrightarrow & F_{P_1} \times \cdots \times F_{P_N} \simeq \mathbb{F}_q^N \\ & f & \longmapsto & \big(f(P_1), \ldots, f(P_N)\big) \end{array}$$

# Algorithme de multiplication dans $\mathbb{F}_{q^n}$

Soit $\mathbf{F}/\mathbb{F}_q$ un corps de fonctions algébriques défini sur $\mathbb{F}_q$ de genre $g$ pour lequel on a

- $Q$ une place de degré $n$,
- $\mathscr{P} := \{P_1, \ldots, P_N\}$ un ensemble de N places rationnelles,
- $\mathcal{D}$ un diviseur effectif tel que $\mathrm{supp}(\mathcal{D}) \cap \{Q, P_1, \ldots, P_N\} = \varnothing$.

Si on a

(i) un morphisme de $\mathbb{F}_q$-espaces vectoriels **surjectif**

$$\begin{array}{rccc} \mathrm{Ev}_Q : & \mathscr{L}(\mathcal{D}) & \longrightarrow & F_Q \simeq \mathbb{F}_{q^n} \\ & f & \longmapsto & f(Q) \end{array}$$

(ii) un morphisme de $\mathbb{F}_q$-espaces vectoriels **injectif**

$$\begin{array}{rccc} \mathrm{Ev}_{\mathscr{P}} : & \mathscr{L}(2\mathcal{D}) & \longrightarrow & F_{P_1} \times \cdots \times F_{P_N} \simeq \mathbb{F}_q^N \\ & f & \longmapsto & \big(f(P_1), \ldots, f(P_N)\big) \end{array}$$

alors

$$\mu_q^{\mathrm{sym}}(n) \leqslant N.$$

# Algorithme de multiplication dans $\mathbb{F}_{q^n}$

Soit $\mathbf{F}/\mathbb{F}_q$ un corps de fonctions algébriques défini sur $\mathbb{F}_q$ de genre $g$ pour lequel on a

- $Q$ une place de degré $n$,
- $\mathscr{P} := \{P_1, \ldots, P_N\}$ un ensemble de N places rationnelles,
- $\mathcal{D}$ un diviseur effectif tel que $\mathrm{supp}(\mathcal{D}) \cap \{Q, P_1, \ldots, P_N\} = \varnothing$.

Si on a

(i) un morphisme de $\mathbb{F}_q$-espaces vectoriels **surjectif**

$$\begin{array}{rccc} \mathrm{Ev}_Q : & \mathscr{L}(\mathcal{D}) & \longrightarrow & F_Q \simeq \mathbb{F}_{q^n} \\ & f & \longmapsto & f(Q) \end{array}$$

(ii) un morphisme de $\mathbb{F}_q$-espaces vectoriels **injectif**

$$\begin{array}{rccc} \mathrm{Ev}_{\mathscr{P}} : & \mathscr{L}(2\mathcal{D}) & \longrightarrow & F_{P_1} \times \cdots \times F_{P_N} \simeq \mathbb{F}_q^N \\ & f & \longmapsto & \big(f(P_1), \ldots, f(P_N)\big) \end{array}$$

alors

$$\mu_q^{\mathrm{sym}}(n) \leqslant N.$$

**Rq.** En pratique, il suffit d'avoir
(i) $\deg(\mathcal{D}) = n + g - 1$ et $\dim \mathcal{D} = n$,
(ii) $N \geqslant 2n + 2g - 1$.
Dans ce cas, la complexité de l'algorithme est
$\mu^{\mathrm{sym}}{}_q(n) \leqslant 2n + g - 1$.

# Algorithme de multiplication dans $\mathbb{F}_{q^n}$

Soit $\mathbf{F}/\mathbb{F}_q$ un corps de fonctions algébriques défini sur $\mathbb{F}_q$ de genre $g$ pour lequel on a

- $Q$ une place de degré $n$,
- $\mathscr{P} := \{P_1, \ldots, P_N\}$ un ensemble de $N$ places rationnelles,
- $\mathcal{D}$ un diviseur effectif tel que $\mathrm{supp}(\mathcal{D}) \cap \{Q, P_1, \ldots, P_N\} = \varnothing$.

Si on a

(i) un morphisme de $\mathbb{F}_q$-espaces vectoriels **surjectif**

$$\begin{array}{rccc} \mathrm{Ev}_Q : & \mathscr{L}(\mathcal{D}) & \longrightarrow & F_Q \simeq \mathbb{F}_{q^n} \\ & f & \longmapsto & f(Q) \end{array}$$

(ii) un morphisme de $\mathbb{F}_q$-espaces vectoriels **injectif**

$$\begin{array}{rccc} \mathrm{Ev}_{\mathscr{P}} : & \mathscr{L}(2\mathcal{D}) & \longrightarrow & F_{P_1} \times \cdots \times F_{P_N} \simeq \mathbb{F}_q^N \\ & f & \longmapsto & \big(f(P_1), \ldots, f(P_N)\big) \end{array}$$

alors

$$\mu_q^{\mathrm{sym}}(n) \leqslant N.$$

**Rq.** En pratique, il suffit d'avoir
(i) $\deg(\mathcal{D}) = n + g - 1$ et $\dim \mathcal{D} = n$,
(ii) $N \geqslant 2n + 2g - 1$.
Dans ce cas, la complexité de l'algorithme est
$\mu^{\mathrm{sym}}{}_q(n) \leqslant 2n + g - 1$.

# Algorithme de multiplication dans $\mathbb{F}_{q^n}$

Soit $\mathbf{F}/\mathbb{F}_q$ un corps de fonctions algébriques défini sur $\mathbb{F}_q$ de genre $g$ pour lequel on a

- $Q$ une place de degré $n$,
- $\mathscr{P} := \{P_1, \ldots, P_N\}$ un ensemble de $N$ places rationnelles,
- $\mathcal{D}$ un diviseur effectif tel que $\mathrm{supp}(\mathcal{D}) \cap \{Q, P_1, \ldots, P_N\} = \varnothing$.

Si on a

(i) un morphisme de $\mathbb{F}_q$-espaces vectoriels **surjectif**

$$\begin{array}{rccc} \mathrm{Ev}_Q : & \mathscr{L}(\mathcal{D}) & \longrightarrow & F_Q \simeq \mathbb{F}_{q^n} \\ & f & \longmapsto & f(Q) \end{array}$$

(ii) un morphisme de $\mathbb{F}_q$-espaces vectoriels **injectif**

$$\begin{array}{rccc} \mathrm{Ev}_{\mathscr{P}} : & \mathscr{L}(2\mathcal{D}) & \longrightarrow & F_{P_1} \times \cdots \times F_{P_N} \simeq \mathbb{F}_q^N \\ & f & \longmapsto & \big(f(P_1), \ldots, f(P_N)\big) \end{array}$$

alors

$$\mu_q^{\mathrm{sym}}(n) \leqslant N.$$

**Rq.** En pratique, il suffit d'avoir
(i) $\deg(\mathcal{D}) = n + g - 1$ et $\dim \mathcal{D} = n$,
(ii) $N \geqslant 2n + 2g - 1$.
Dans ce cas, la complexité de l'algorithme est
$\mu^{\mathrm{sym}}{}_q(n) \leqslant 2n + g - 1$.

**Borne de Serre**
$|N - (q+1)| \leqslant g\left[2\sqrt{q}\right]$

# Algorithme de Chudnovsky-Chudnovsky sur le corps de fonctions Hermitien

Soit $F/\mathbb{F}_{q^2}$ le corps défini par : $F := \mathbb{F}_{q^2}(x, y)$ où $y^q + y = x^{q+1}$.

Alors $g(F) = \frac{q(q-1)}{2}$ et $N(F) = q^2 + 1 + 2gq = q^3 + 1$.

# Algorithme de Chudnovsky-Chudnovsky sur le corps de fonctions Hermitien

Soit $F/\mathbb{F}_{q^2}$ le corps défini par : $F := \mathbb{F}_{q^2}(x, y)$ où $y^q + y = x^{q+1}$.

Alors $g(F) = \frac{q(q-1)}{2}$ et $N(F) = q^2 + 1 + 2gq = q^3 + 1$.

Si $N(F) \geqslant 2n + 2g(F) - 1$ i.e. $n \leqslant \frac{1}{2}(q^3 - q^2 + q + 2)$

alors on peut appliquer l'algorithme de Chudnovsky-Chudnovsky sur le corps de fonctions $F/\mathbb{F}_{q^2}$ pour multiplier dans $\mathbb{F}_{q^{2n}}$.

# Algorithme de Chudnovsky-Chudnovsky sur le corps de fonctions Hermitien

Soit $F/\mathbb{F}_{q^2}$ le corps défini par : $F := \mathbb{F}_{q^2}(x, y)$ où $y^q + y = x^{q+1}$.

Alors $g(F) = \frac{q(q-1)}{2}$ et $N(F) = q^2 + 1 + 2gq = q^3 + 1$.

Si $N(F) \geqslant 2n + 2g(F) - 1$ i.e. $n \leqslant \frac{1}{2}(q^3 - q^2 + q + 2)$

alors on peut appliquer l'algorithme de Chudnovsky-Chudnovsky sur le corps de fonctions $F/\mathbb{F}_{q^2}$ pour multiplier dans $\mathbb{F}_{q^{2n}}$.

**Exemple.** Pour $q := 4$ : multiplication dans des extensions $\mathbb{F}_{16^n}/\mathbb{F}_{16}$.

On a $g(F/\mathbb{F}_{16}) = 6$ et $N(F/\mathbb{F}_{16}) = 65$ ; on peut donc déterminer un algorithme de multiplication dans $\mathbb{F}_{16^n}$ à partir de $F/\mathbb{F}_{16}$ tant que $n \leqslant 27$.

Pour $n \leqslant 27$, la complexité bilinéaire de l'algorithme obtenu est :

$$\mu_{q^2}^{\text{sym}}(n) \leqslant 2n + g - 1.$$

# Algorithme de Chudnovsky-Chudnovsky sur le corps de fonctions Hermitien

Soit $F/\mathbb{F}_{q^2}$ le corps défini par : $F := \mathbb{F}_{q^2}(x, y)$ où $y^q + y = x^{q+1}$.

Alors $g(F) = \frac{q(q-1)}{2}$ et $N(F) = q^2 + 1 + 2gq = q^3 + 1$.

Si $N(F) \geqslant 2n + 2g(F) - 1$ i.e. $n \leqslant \frac{1}{2}(q^3 - q^2 + q + 2)$

alors on peut appliquer l'algorithme de Chudnovsky-Chudnovsky sur le corps de fonctions $F/\mathbb{F}_{q^2}$ pour multiplier dans $\mathbb{F}_{q^{2n}}$.

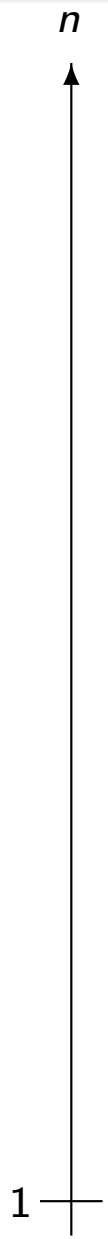
**Exemple.** Pour $q := 4$ : multiplication dans des extensions $\mathbb{F}_{16^n}/\mathbb{F}_{16}$.

On a $g(F/\mathbb{F}_{16}) = 6$ et $N(F/\mathbb{F}_{16}) = 65$ ; on peut donc déterminer un algorithme de multiplication dans $\mathbb{F}_{16^n}$ à partir de $F/\mathbb{F}_{16}$ tant que $n \leqslant 27$.

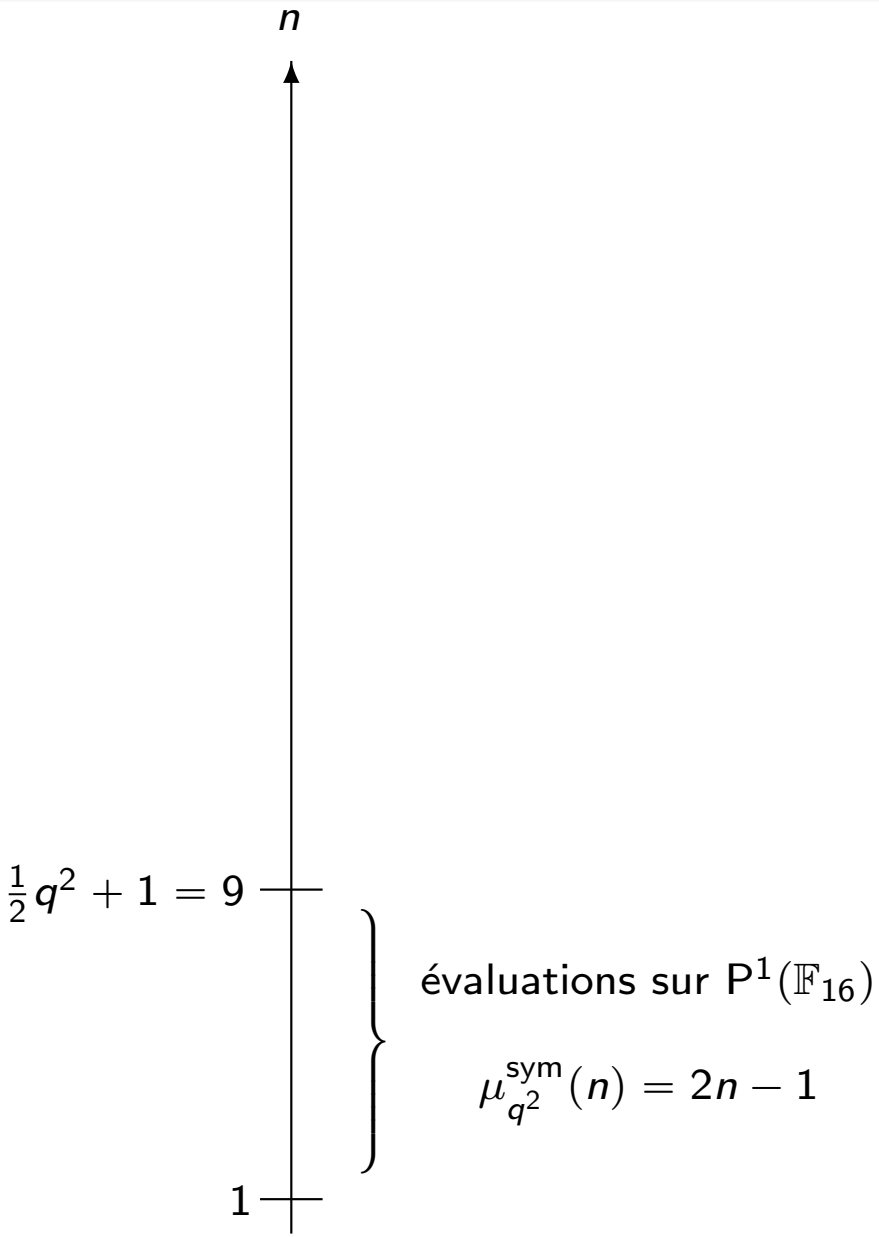
Pour $n \leqslant 27$, la complexité bilinéaire de l'algorithme obtenu est :

$$\mu_{q^2}^{\text{sym}}(n) \leqslant 2n + 5.$$

# Cas des *petites* extensions $\mathbb{F}_{16^n}$ de $\mathbb{F}_{16}$

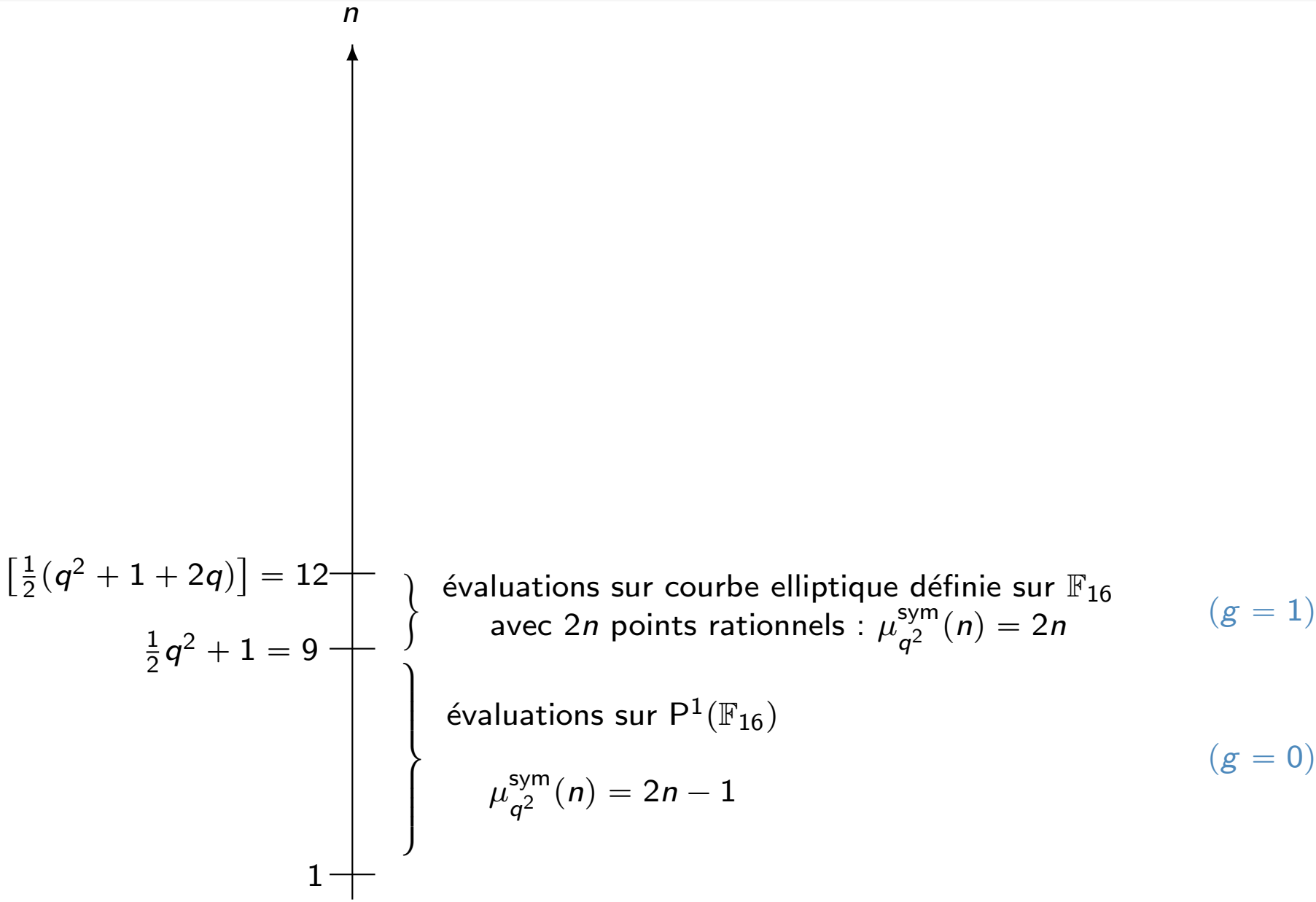


# Cas des *petites* extensions $\mathbb{F}_{16^n}$ de $\mathbb{F}_{16}$

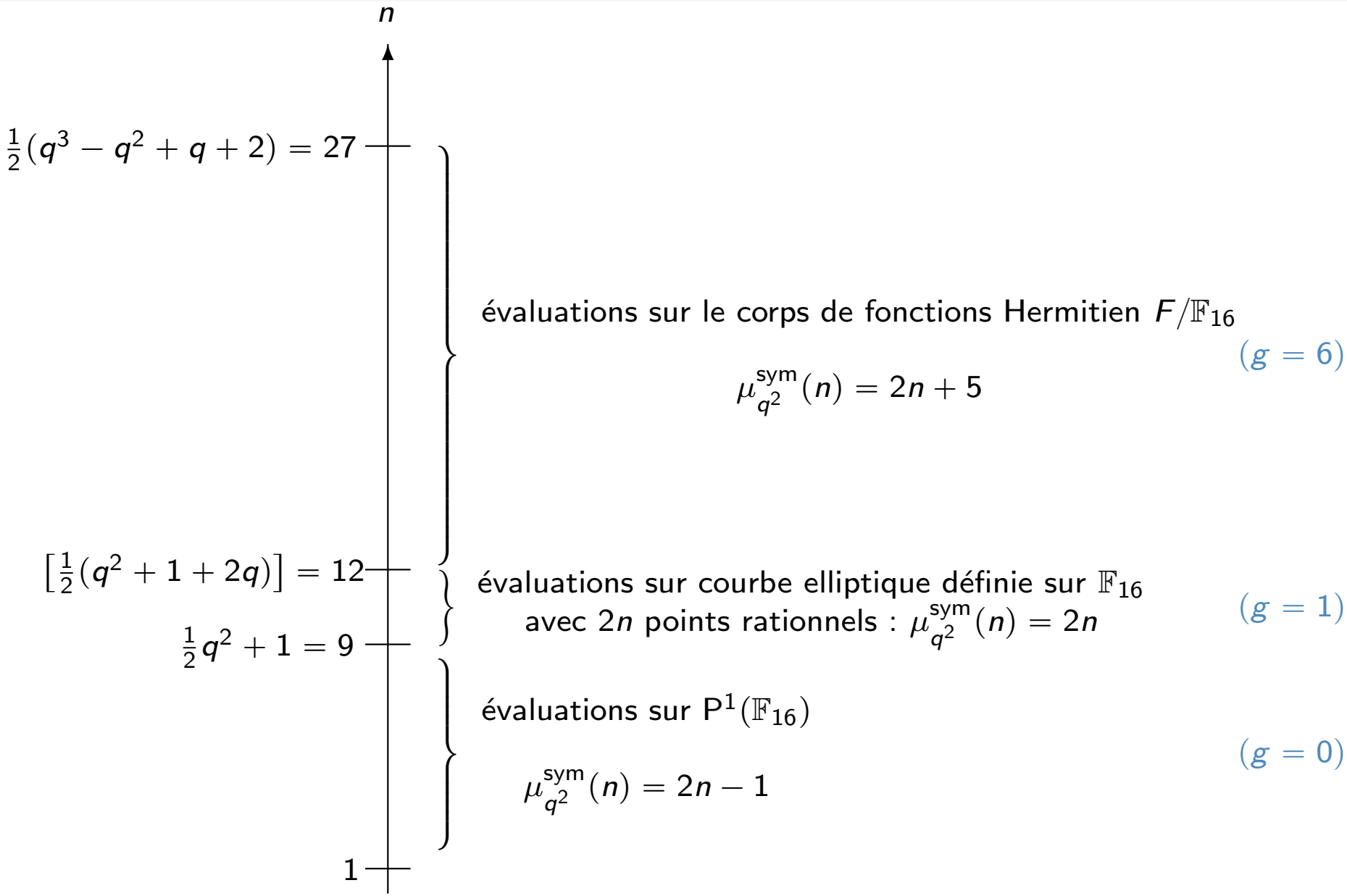



$(g = 0)$

# Cas des *petites* extensions $\mathbb{F}_{16^n}$ de $\mathbb{F}_{16}$



# Cas des *petites* extensions $\mathbb{F}_{16^n}$ de $\mathbb{F}_{16}$



L'application de l'algorithme sur une **suite asymptotiquement bonne de corps de fonctions** prouve que la complexité bilinéaire de la multiplication dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$ est **linéaire en** $n$ :

**Théorème (Chudnovsky et Chudnovsky (1987))**

*Soit $q = p^r$ avec $p$ premier, il existe une constante $C_q$ telle que pour tout $n$,*

$$\mu_q^{\mathrm{sym}}(n) \leqslant C_q n.$$

L'application de l'algorithme sur une **suite asymptotiquement bonne de corps de fonctions** prouve que la complexité bilinéaire de la multiplication dans $\mathbb{F}_{q^n}$ sur $\mathbb{F}_q$ est **linéaire en** $n$ :

**Théorème (Chudnovsky et Chudnovsky (1987))**

*Soit $q = p^r$ avec $p$ premier, il existe une constante $C_q$ telle que pour tout $n$,*

$$\mu_q^{\mathrm{sym}}(n) \leqslant C_q n.$$

**Objectifs.**

- Établir des bornes théoriques :
  - améliorer l'algorithme (évaluations sur des places de degré supérieur, dissymétrisation...)
  - démontrer l'existence de corps de fonctions avec de meilleures propriétés
- Construire des algorithmes explicites pour la multiplication dans $\mathbb{F}_{q^n}$, pour $n$ choisi.

# Bornes uniformes

$$C_q = \begin{cases} 19.6 & \text{si } q = 2 & \left[\begin{array}{l}\text{Ballet, P. (2011)}\\ \text{Cenk, Özbudak (2010)}\end{array}\right] \\ 27 & \text{si } q = 3 & [\text{Ballet (1999)}] \\ 2\left(1+\frac{2}{p-3}\right) & \text{si } q = p^2 \geqslant 25 & [\text{Ballet, Chaumine (2004)}] \\ 2\left(1+\frac{p}{\sqrt{q}-3}\right) & \text{si } q = p^{2m} \geqslant 16 & [\text{Ballet (2003)}] \\ 3\left(1+\frac{4}{p-3}\right) & \text{si } q = p \geqslant 5 & [\text{Ballet, Chaumine (2004)}] \\ 3\left(1+\frac{2p}{q-3}\right) & \text{si } q = p^m \geqslant 16 & \left[\begin{array}{l}\text{Ballet, Rolland (2004)}\\ \text{Ballet, Le Brigand, Rolland (2009)}\\ \text{Ballet, Le Brigand (2006)}\end{array}\right] \\ 6\left(1+\frac{p}{q-3}\right) & \text{si } q = p^m > 3 & [\text{Ballet (2003)}] \end{cases}$$

**Stratégie :**

- utiliser des tours asymptotiquement optimales sur $\mathbb{F}_{q^2}$, et leur descente sur $\mathbb{F}_q$ et $\mathbb{F}_p$,
- «densifier» les tours.

# Merci pour votre attention