Généralisation de la transformée de Fourier tronquée pour des ordres quelconques

Robin Larrieu

Laboratoire d'informatique de l'École polytechnique (LIX)

JNCF 2017
16 janvier 2017

Exemple : multiplication de polynômes

Soient $A, B \in \mathbb{K}[X]$, $\deg AB < n$.

Méthode naïve

$$AB = \sum_{i<n} \sum_{j<i} A_j B_{i-j} X^i$$

Complexité quadratique

Exemple : multiplication de polynômes

Soient $A, B \in \mathbb{K}[X]$, $\deg AB < n$.

Évaluation-Interpolation

Exemple : multiplication de polynômes

Soient $A, B \in \mathbb{K}[X]$, $\deg AB < n$.

Évaluation-Interpolation

- Évaluation

$$\begin{aligned} A &\rightarrow (A(x_0), A(x_1), \ldots, A(x_{n-1})) \\ B &\rightarrow (B(x_0), B(x_1), \ldots, B(x_{n-1})) \end{aligned}$$

Exemple : multiplication de polynômes

Soient $A, B \in \mathbb{K}[X]$, $\deg AB < n$.

Évaluation-Interpolation

- Évaluation

$$\begin{aligned} A &\rightarrow (A(x_0), A(x_1), \ldots, A(x_{n-1})) \\ B &\rightarrow (B(x_0), B(x_1), \ldots, B(x_{n-1})) \end{aligned}$$

- Interpolation

$$(AB(x_0), AB(x_1), \ldots, AB(x_{n-1})) \rightarrow AB$$

FFT et TFT

FFT

- Soit ω une racine primitive n-ième de l'unité

$$A \quad \overset{FFT}{\longleftrightarrow} \quad (A(1), A(\omega), \ldots, A(\omega^{n-1}))$$

- Complexité $O(n \lg n)$ si n est une puissance de 2.

FFT et TFT

FFT

- Soit ω une racine primitive n-ième de l'unité

$$A \quad \overset{FFT}{\longleftrightarrow} \quad (A(1), A(\omega), \ldots, A(\omega^{n-1}))$$

- Complexité $O(n \lg n)$ si n est une puissance de 2.

Et si $n/2 \leqslant deg(AB) < n$?

FFT et TFT

FFT

- Soit ω une racine primitive n-ième de l'unité

$$A \quad \overset{FFT}{\longleftrightarrow} \quad (A(1), A(\omega), \ldots, A(\omega^{n-1}))$$

- Complexité $O(n \lg n)$ si n est une puissance de 2.

Et si $n/2 \leqslant deg(AB) < n$?

TFT

- Évaluation-Interpolation de taille l quelconque sur la base d'une FFT de taille n ($l \leqslant n$).
- Complexité $(l/n)\mathrm{F}(n) + O(n)$.

Généralisation de la TFT ; pourquoi ?

Choix des racines de l'unité : $n = 2^k$ pas toujours possible.

Généralisation de la TFT ; pourquoi ?

Choix des racines de l'unité : $n = 2^k$ pas toujours possible.

Exemple remarquable : $\mathbb{F}_{2^{60}}$

- représentation sur 60 bits $\Rightarrow$ 1 mot machine.

Généralisation de la TFT ; pourquoi ?

Choix des racines de l'unité : $n = 2^k$ pas toujours possible.

Exemple remarquable : $\mathbb{F}_{2^{60}}$

- représentation sur 60 bits $\Rightarrow$ 1 mot machine.
- $(X^{61} - 1)/(X - 1)$ irréductible sur $\mathbb{F}_2$ $\Rightarrow$ multiplication simple dans $\mathbb{F}_{2^{60}}$.

Généralisation de la TFT ; pourquoi ?

Choix des racines de l'unité : $n = 2^k$ pas toujours possible.

Exemple remarquable : $\mathbb{F}_{2^{60}}$

- représentation sur 60 bits $\Rightarrow$ 1 mot machine.
- $(X^{61} - 1)/(X - 1)$ irréductible sur $\mathbb{F}_2$ $\Rightarrow$ multiplication simple dans $\mathbb{F}_{2^{60}}$.
- Racines de l'unité d'ordre $2^{60} - 1$ avec

$$2^{60} - 1 = 3^2 \cdot 5^2 \cdot 7 \cdot 11 \cdot 13 \cdot 31 \cdot 41 \cdot 61 \cdot 151 \cdot 331 \cdot 1321$$

$\Rightarrow$ FFT très efficace.

Table des matières

Principe de la FFT (Cooley & Tukey) [CT65]

Définition de la transformée de Fourier discrète :

$$\hat{A}_k = A(\omega^k) = \sum_{i=0}^{n-1} A_i \omega^{ik}$$

Principe de la FFT (Cooley & Tukey) [CT65]

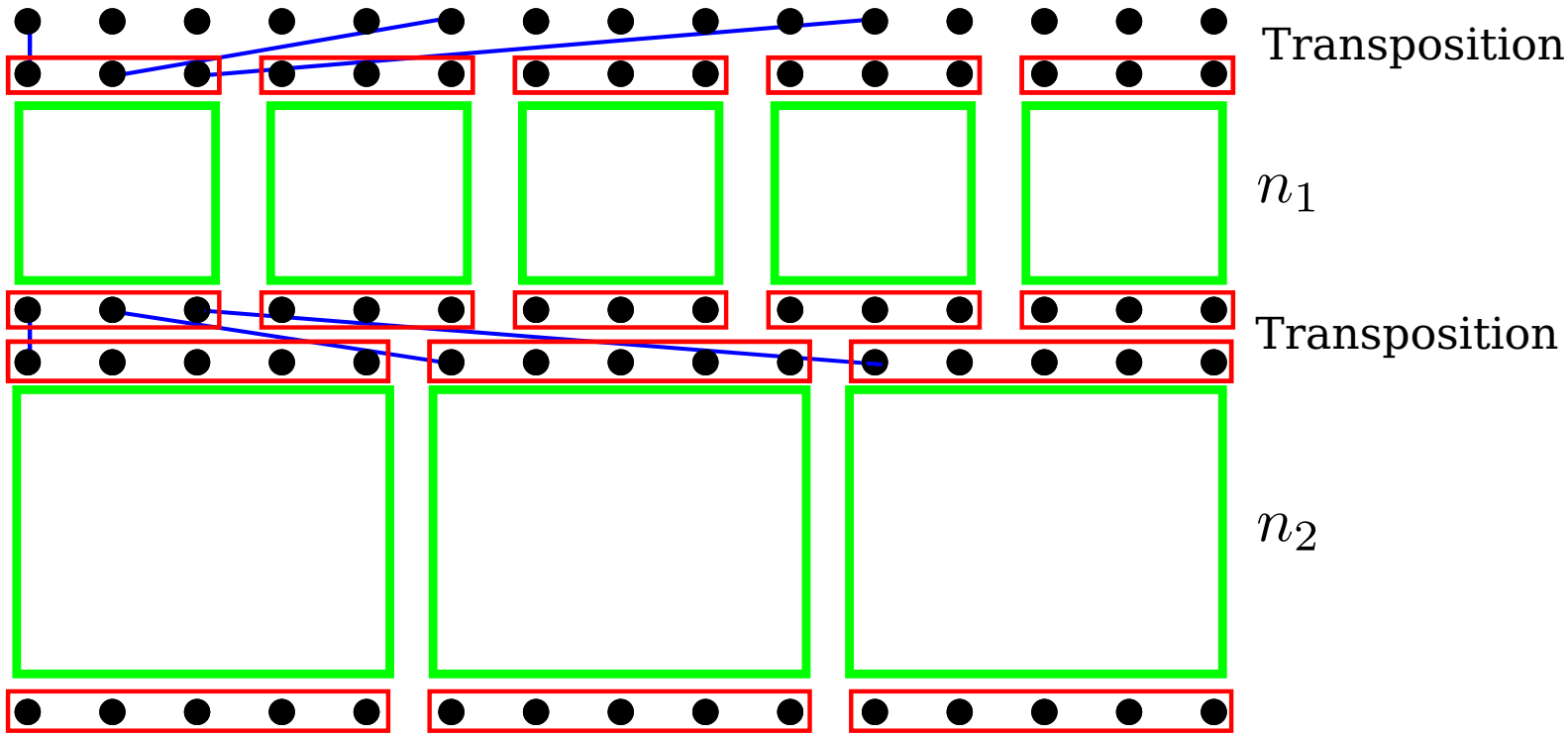
Définition de la transformée de Fourier discrète :

$$\hat{A}_k = A(\omega^k) = \sum_{i=0}^{n-1} A_i \omega^{ik}$$

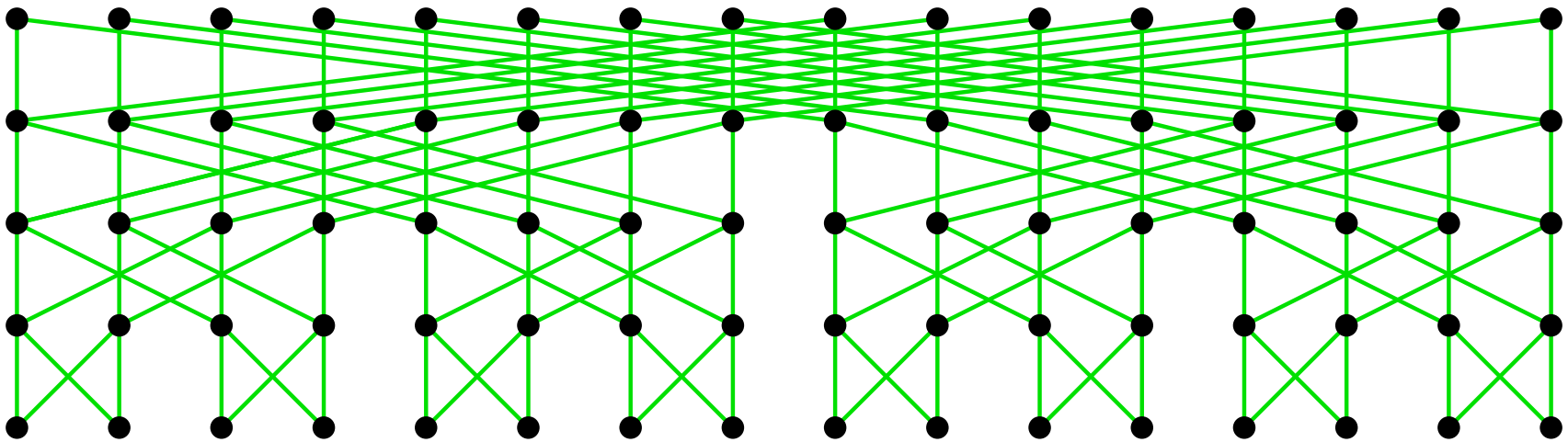
Si $n = n_1 n_2$, on peut écrire :

$$\hat{A}_{k_1+n_1 k_2} = \sum_{i=0}^{n_2-1} \omega^{ik_1} \cdot \left(\sum_{j=0}^{n_1-1} A_{i+n_2 j} \cdot (\omega^{n_2})^{jk_1} \right) \cdot (\omega^{n_1})^{ik_2}$$

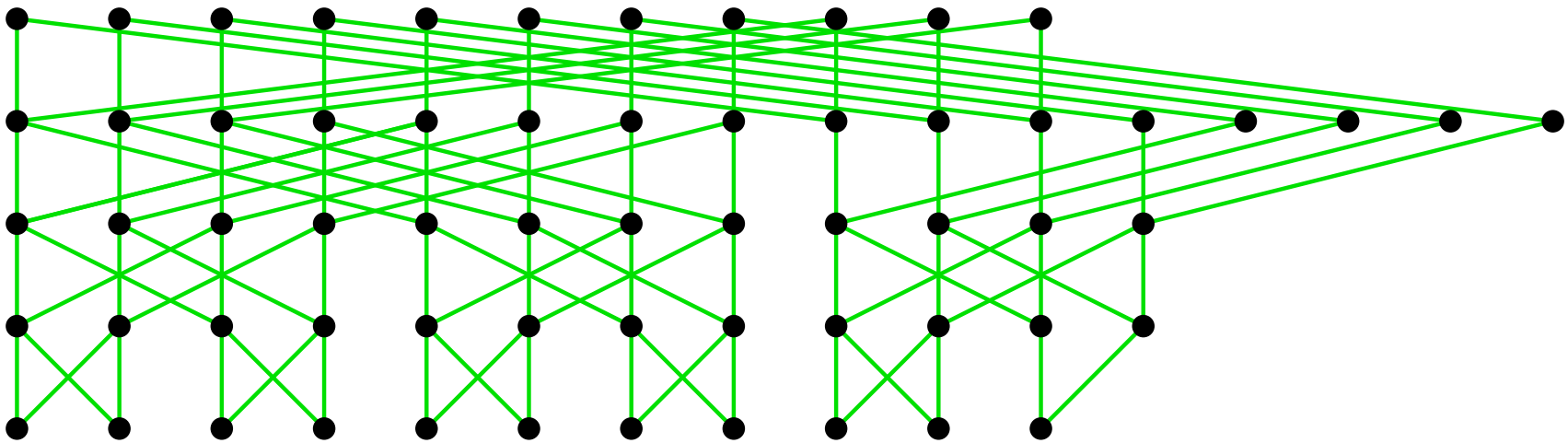
FFT d'ordre quelconque



FFT d'ordre une puissance de 2



TFT d'ordre une puissance de 2 [vdH04]



TFT d'ordre une puissance de 2 [vdH04]

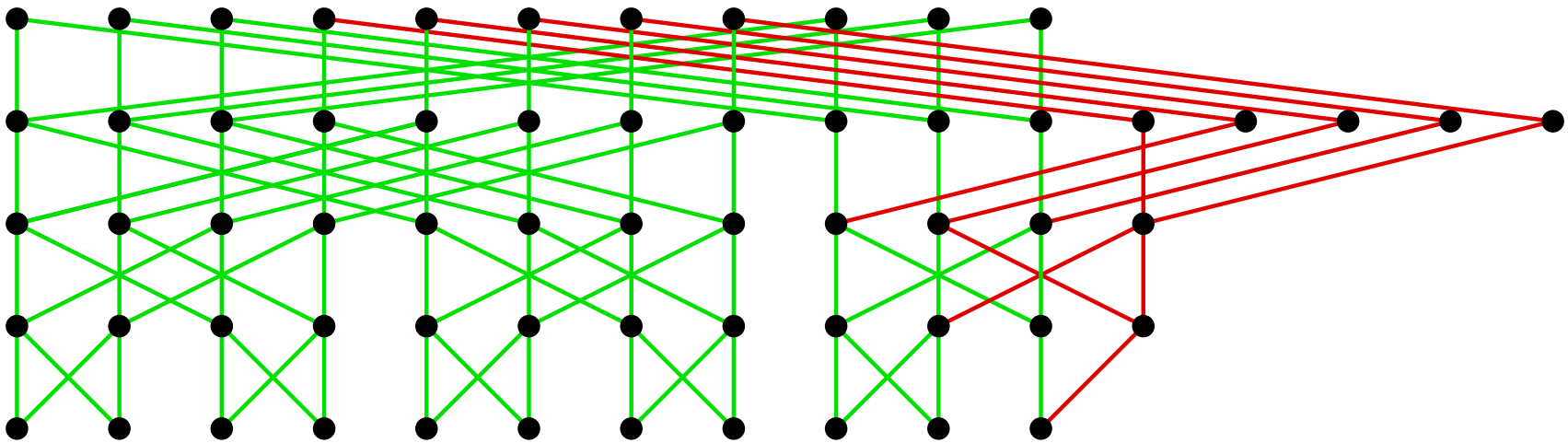
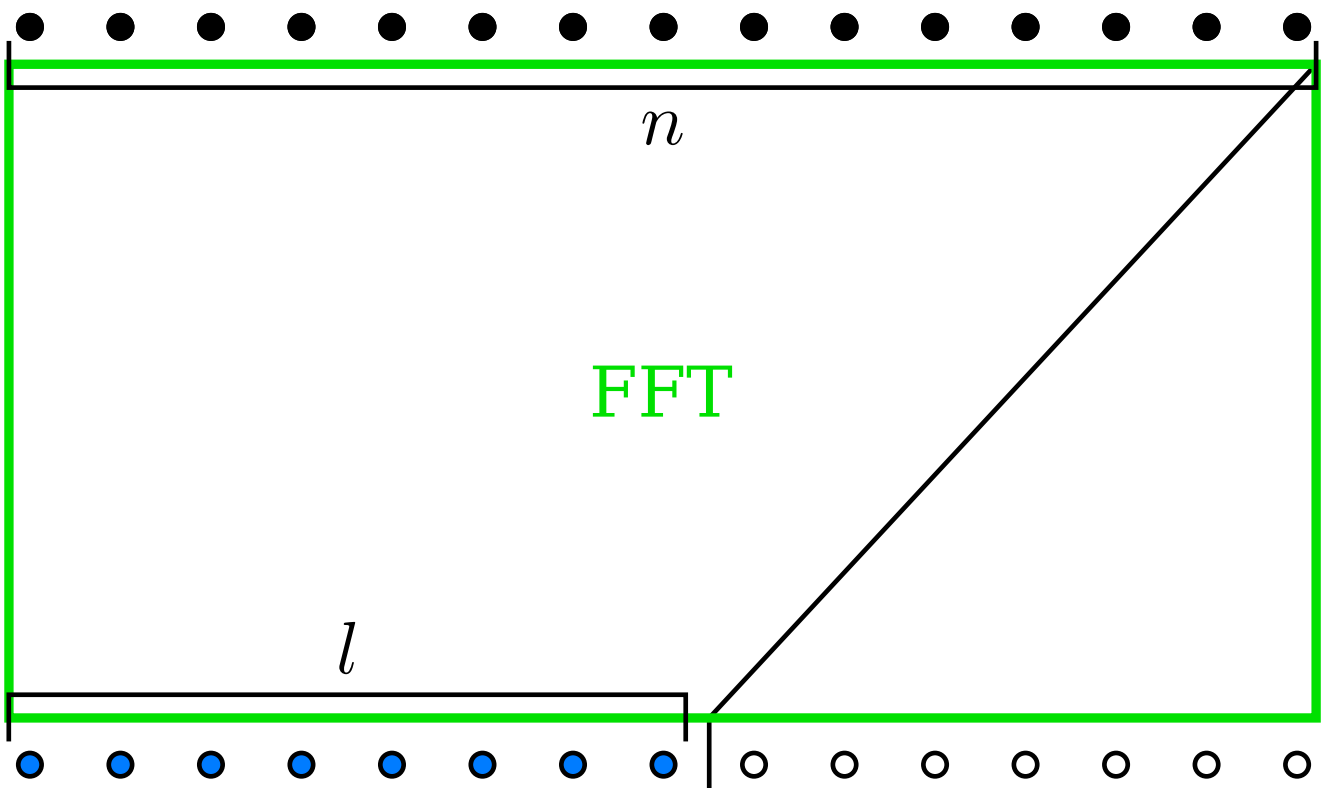


Table des matières

Principe général



Transformations atomiques

TFT d'ordre premier

- Calcul direct (méthode de Horner)
- FFT complète (Rader) pour p grand

Algorithme récursif

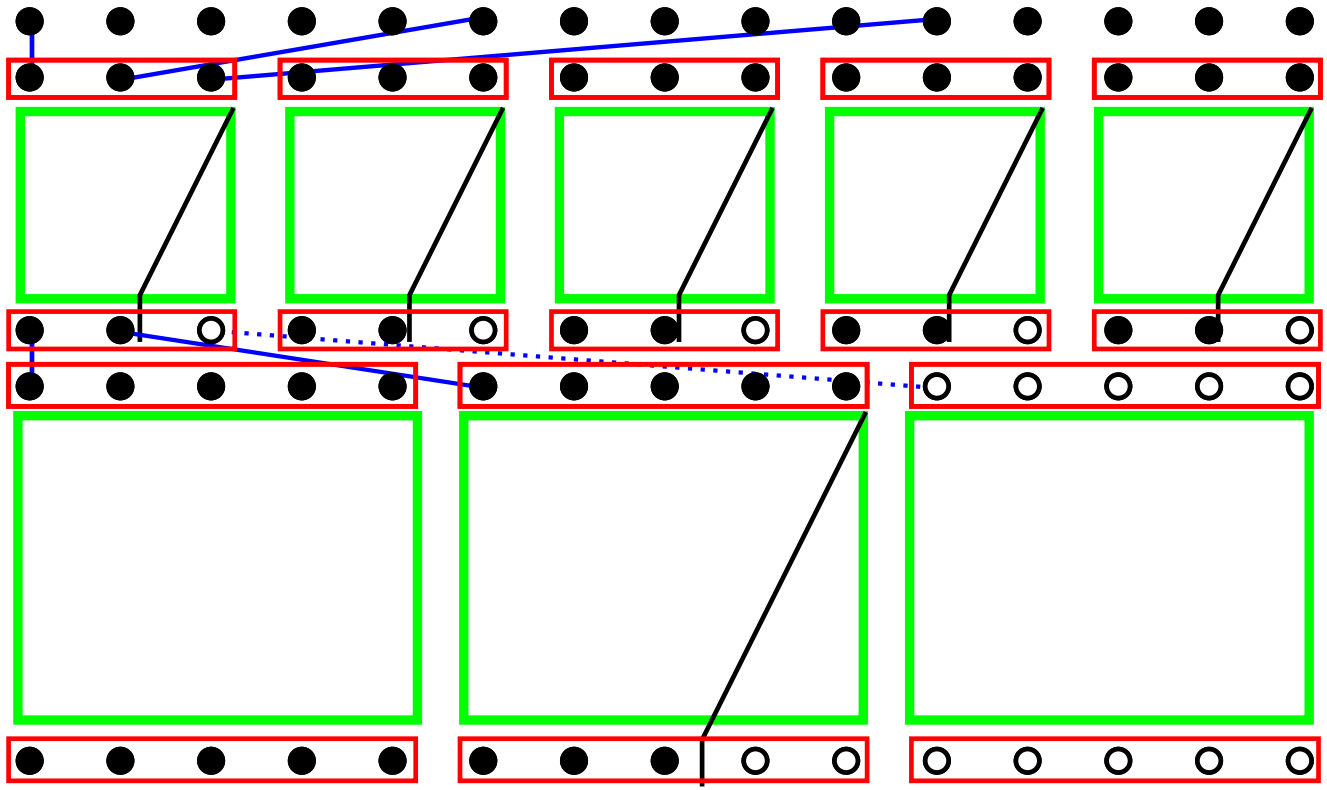
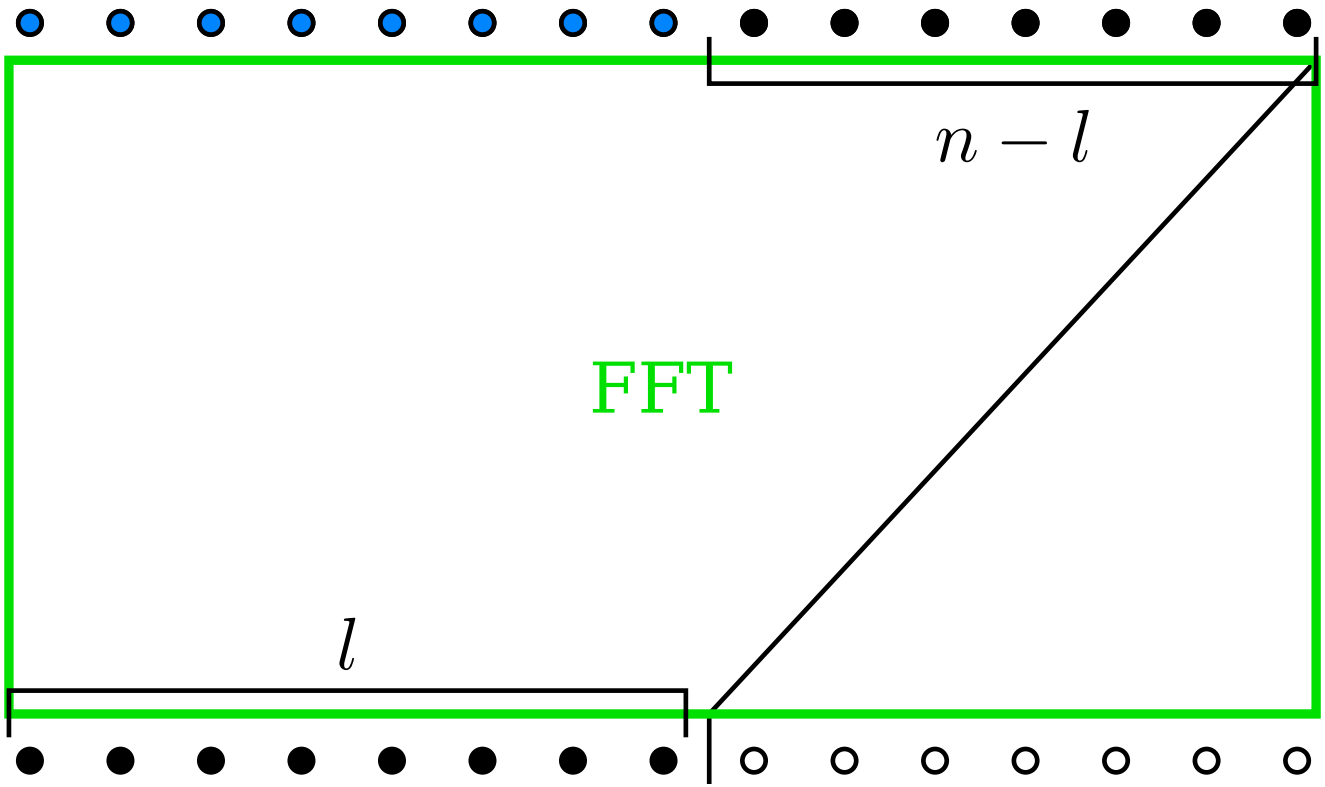
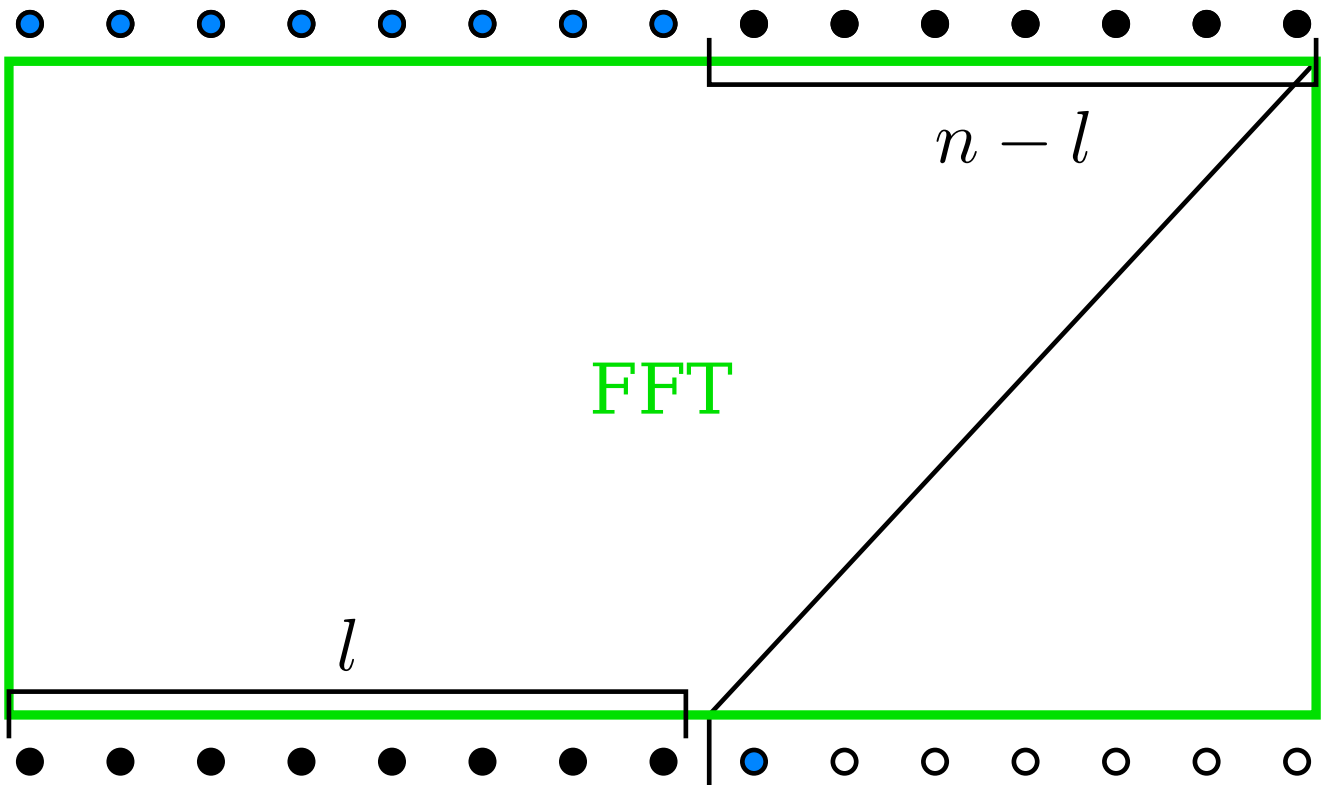


Table des matières

Principe général



Principe général



Transformations atomiques

Problème : résoudre l'équation matricielle (ici pour $n = 5$, $l = 3$)

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3^{(?)} \\ b_4^{(?)} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 \\ 1 & \omega^2 & \omega^4 & \omega & \omega^3 \\ 1 & \omega^3 & \omega & \omega^4 & \omega^2 \\ 1 & \omega^4 & \omega^3 & \omega^2 & \omega \end{pmatrix} \begin{pmatrix} a_0^{(?)} \\ a_1^{(?)} \\ a_2^{(?)} \\ a_3 \\ a_4 \end{pmatrix}$$

Transformations atomiques

Problème : résoudre l'équation matricielle par blocs

$$\begin{pmatrix} B_1 \\ B_2^{(?)} \end{pmatrix} = \begin{pmatrix} V & W \\ W^\top & \tilde{V} \end{pmatrix} \cdot \begin{pmatrix} A_1^{(?)} \\ A_2 \end{pmatrix}$$

Transformations atomiques

Problème : résoudre l'équation matricielle par blocs

$$\begin{pmatrix} B_1 \\ B_2^{(?)} \end{pmatrix} = \begin{pmatrix} V & W \\ W^\top & \tilde{V} \end{pmatrix} \cdot \begin{pmatrix} A_1^{(?)} \\ A_2 \end{pmatrix}$$

$$A_1 = V^{-1}(B_1 - WA_2)$$

Transformations atomiques

Problème : résoudre l'équation matricielle par blocs

$$\begin{pmatrix} B_1 \\ B_2^{(?)} \end{pmatrix} = \begin{pmatrix} V & W \\ W^\top & \tilde{V} \end{pmatrix} \cdot \begin{pmatrix} A_1^{(?)} \\ A_2 \end{pmatrix}$$

$$A_1 = V^{-1}(B_1 - WA_2)$$
$$B_2 = W^\top A_1 + \tilde{V} A_2$$

Exemple

Cas $n = 2$ et $l = 1$

$$\begin{pmatrix} b_0 \\ b_1^{(?)} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} a_0^{(?)} \\ a_1 \end{pmatrix}$$

Exemple

Cas $n = 2$ et $l = 1$

$$\begin{pmatrix} b_0 \\ b_1^{(?)} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} a_0^{(?)} \\ a_1 \end{pmatrix}$$

Alors

$$a_0 = b_0 - a_1$$

Exemple

Cas $n = 2$ et $l = 1$

$$\begin{pmatrix} b_0 \\ b_1^{(?)} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} a_0^{(?)} \\ a_1 \end{pmatrix}$$

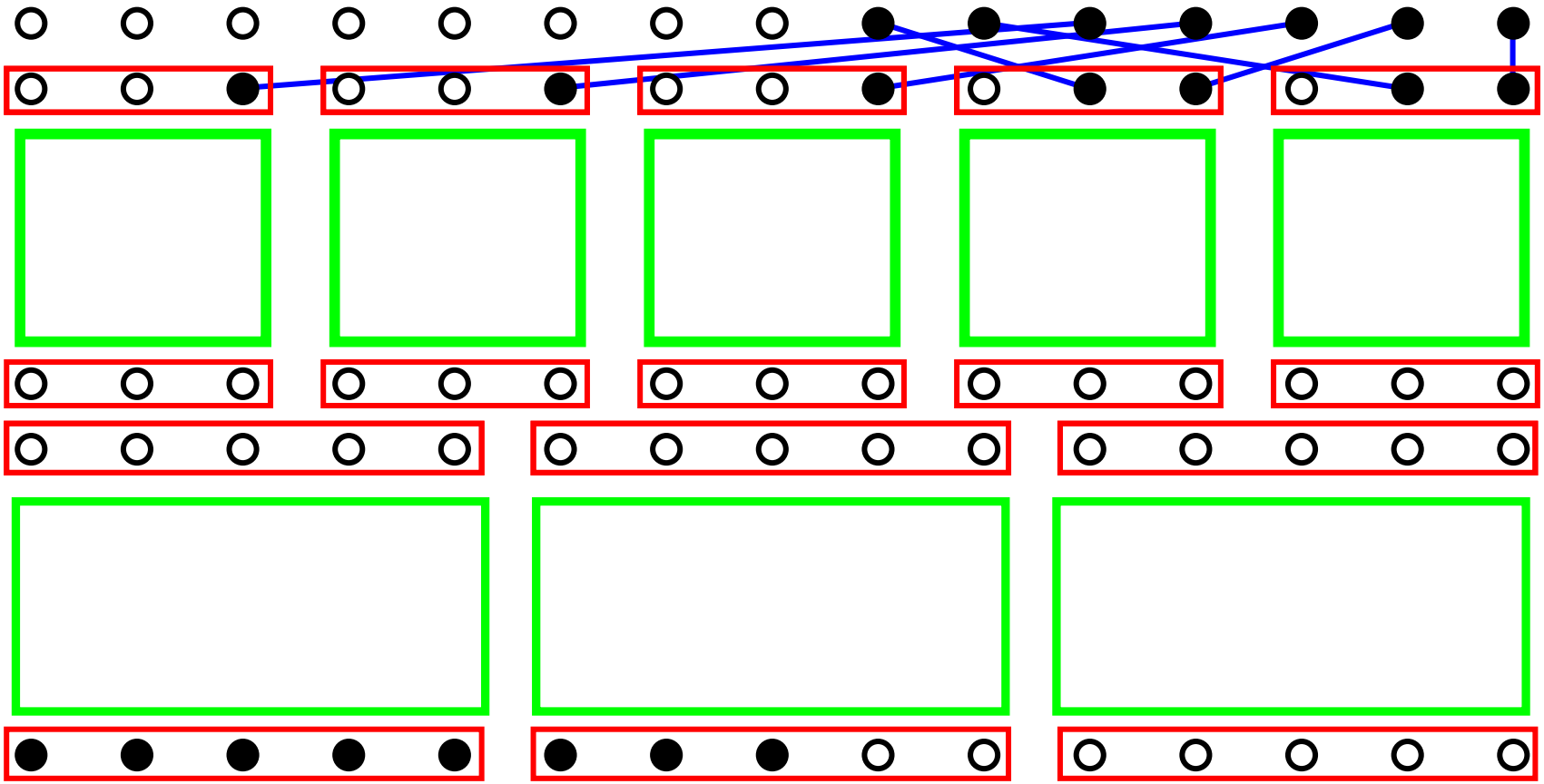
Alors

$$a_0 = b_0 - a_1$$

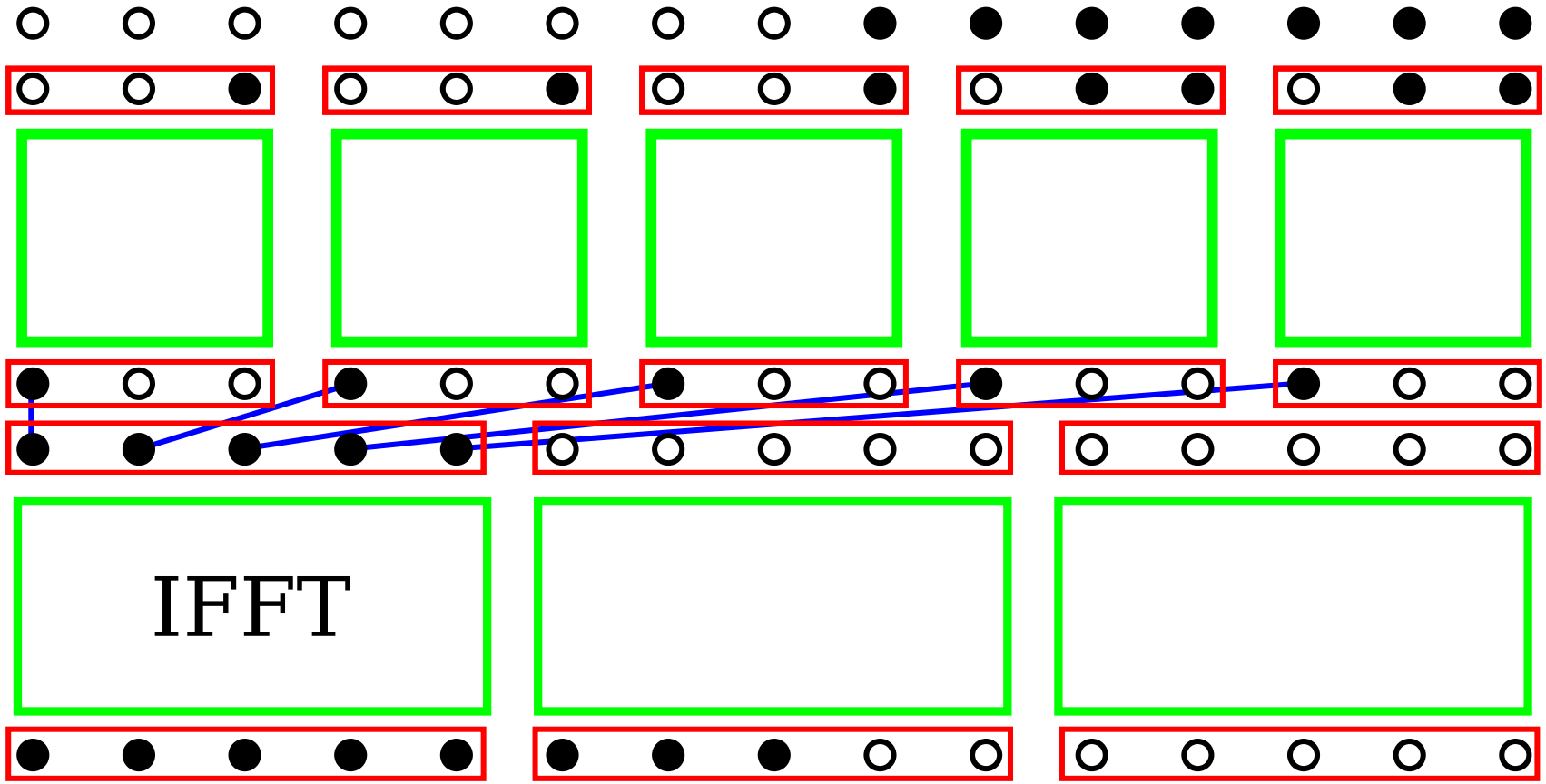
Puis

$$b_1 = b_0 - 2a_1$$

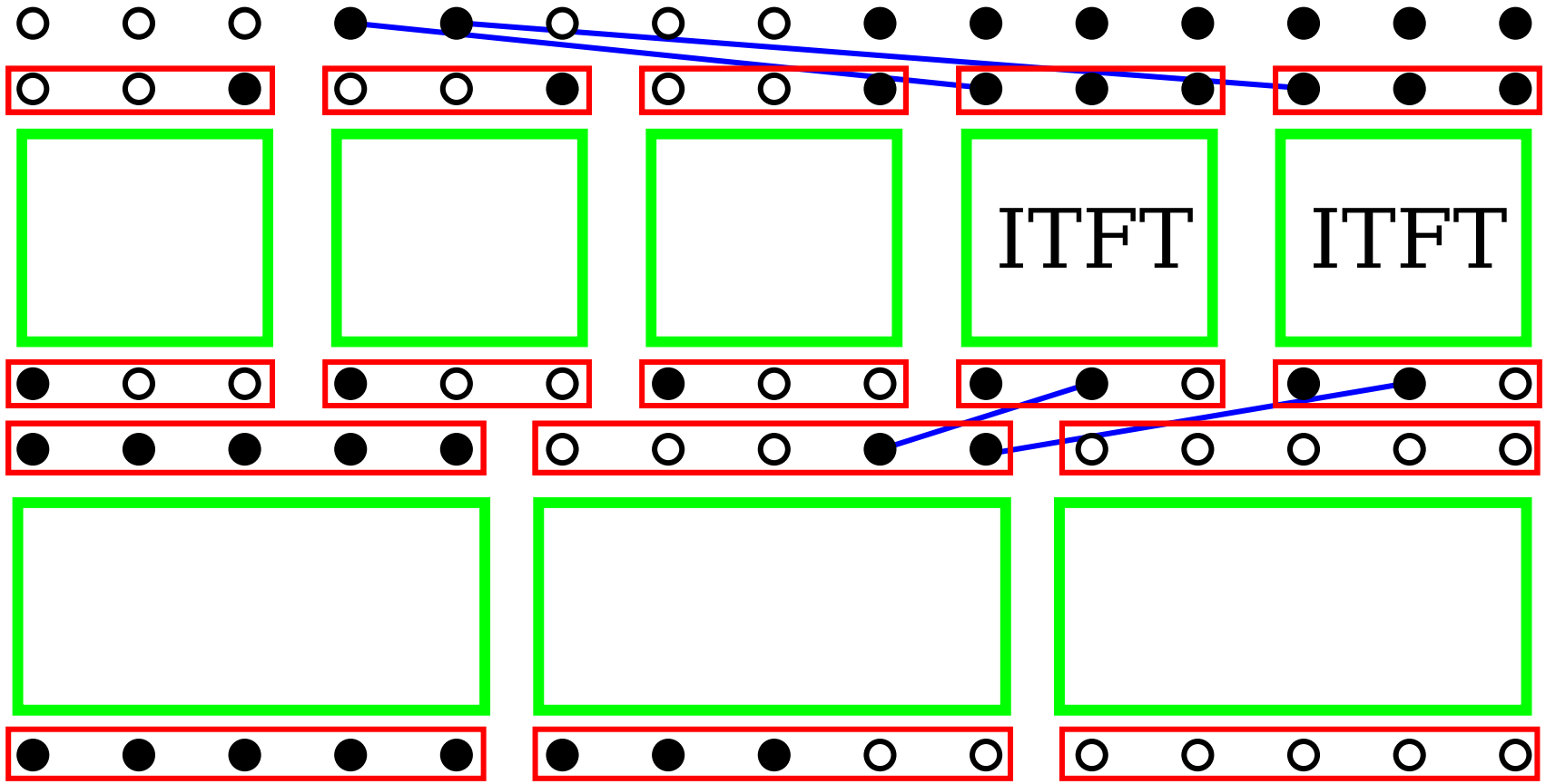
Algorithme récursif



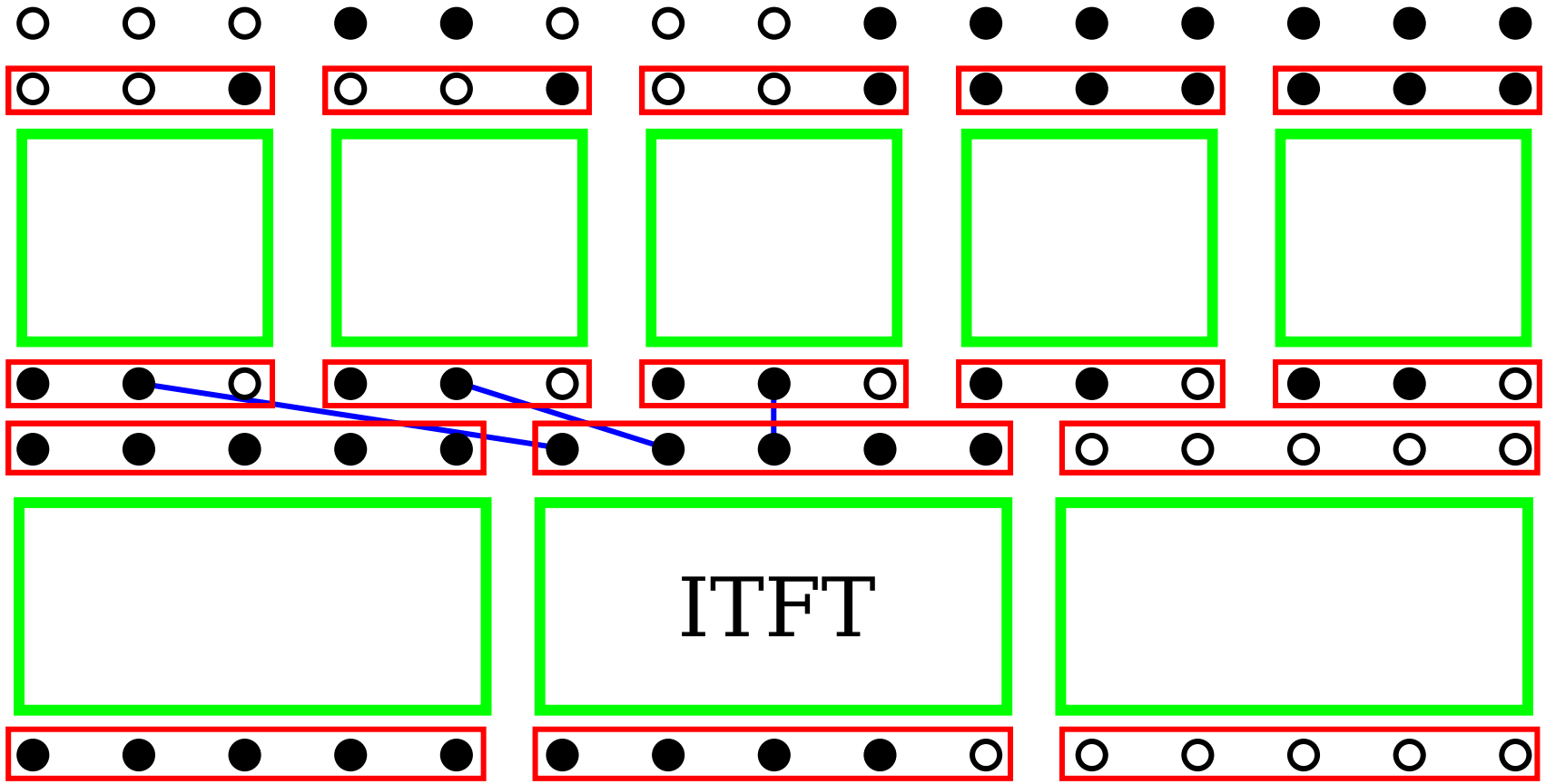
Algorithme récursif



Algorithme récursif



Algorithme récursif



Algorithme récursif

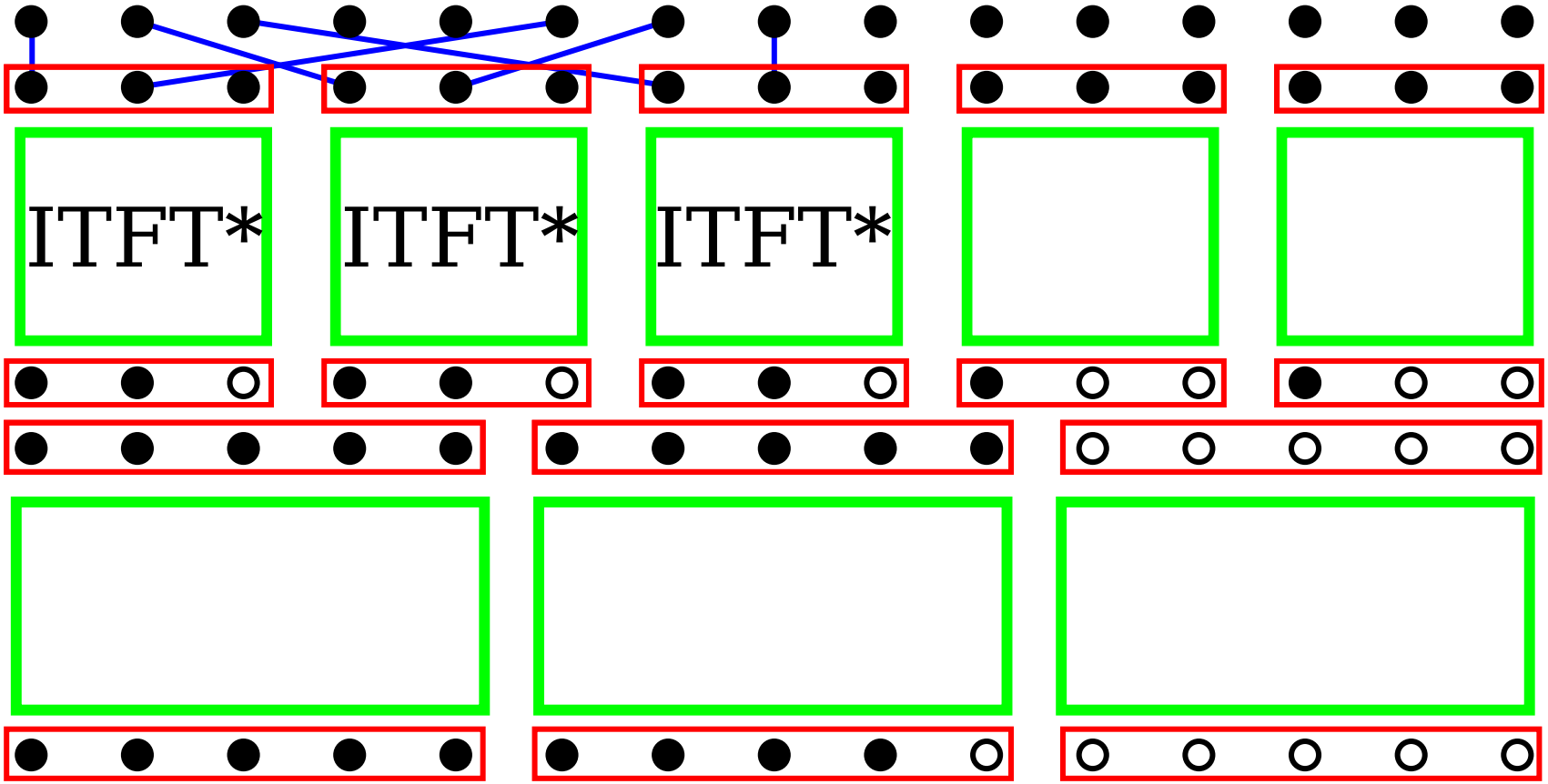
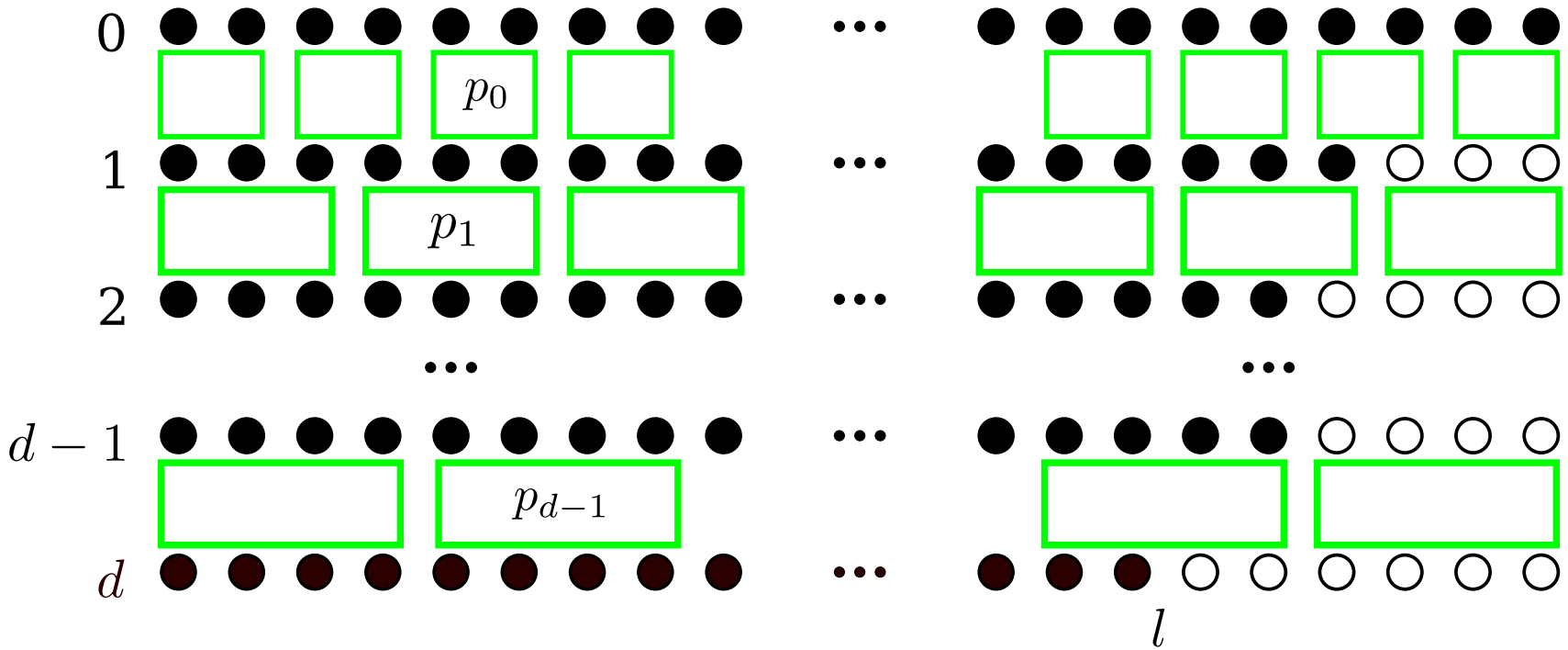
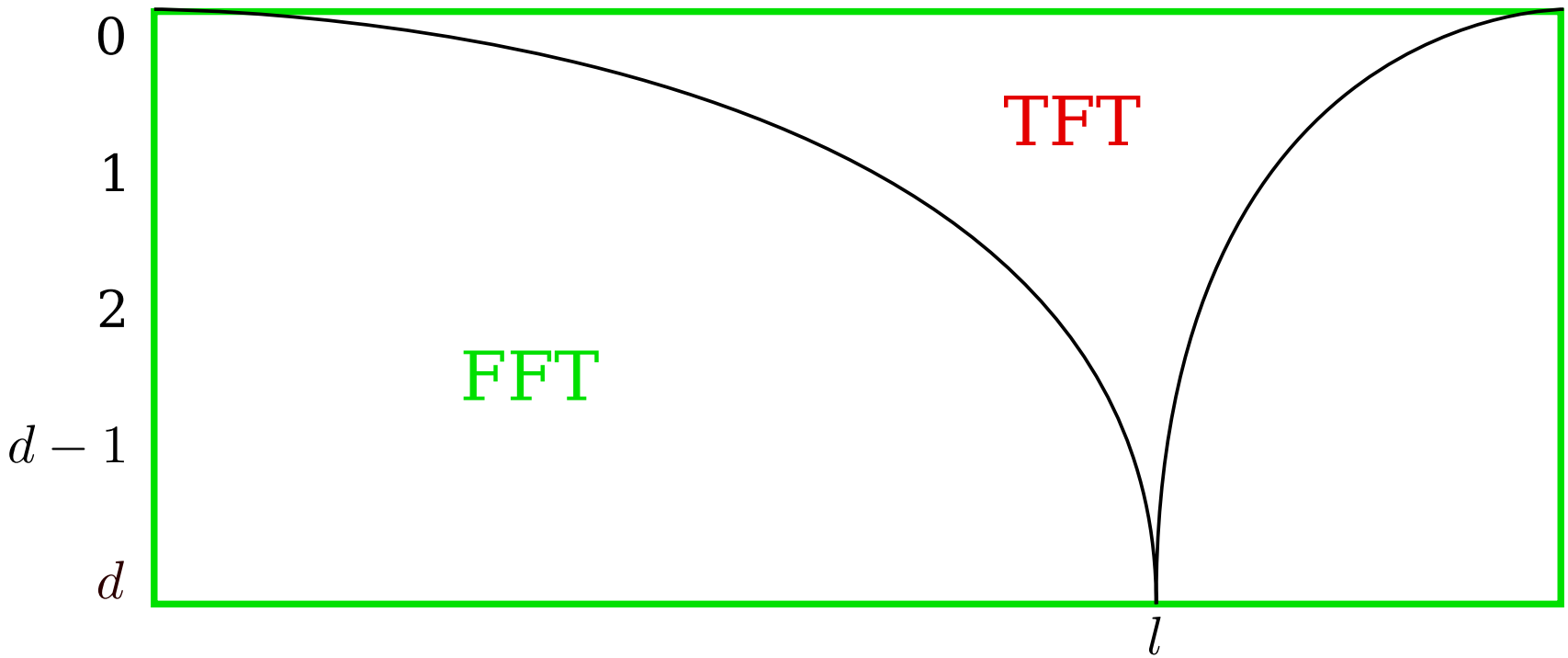


Table des matières

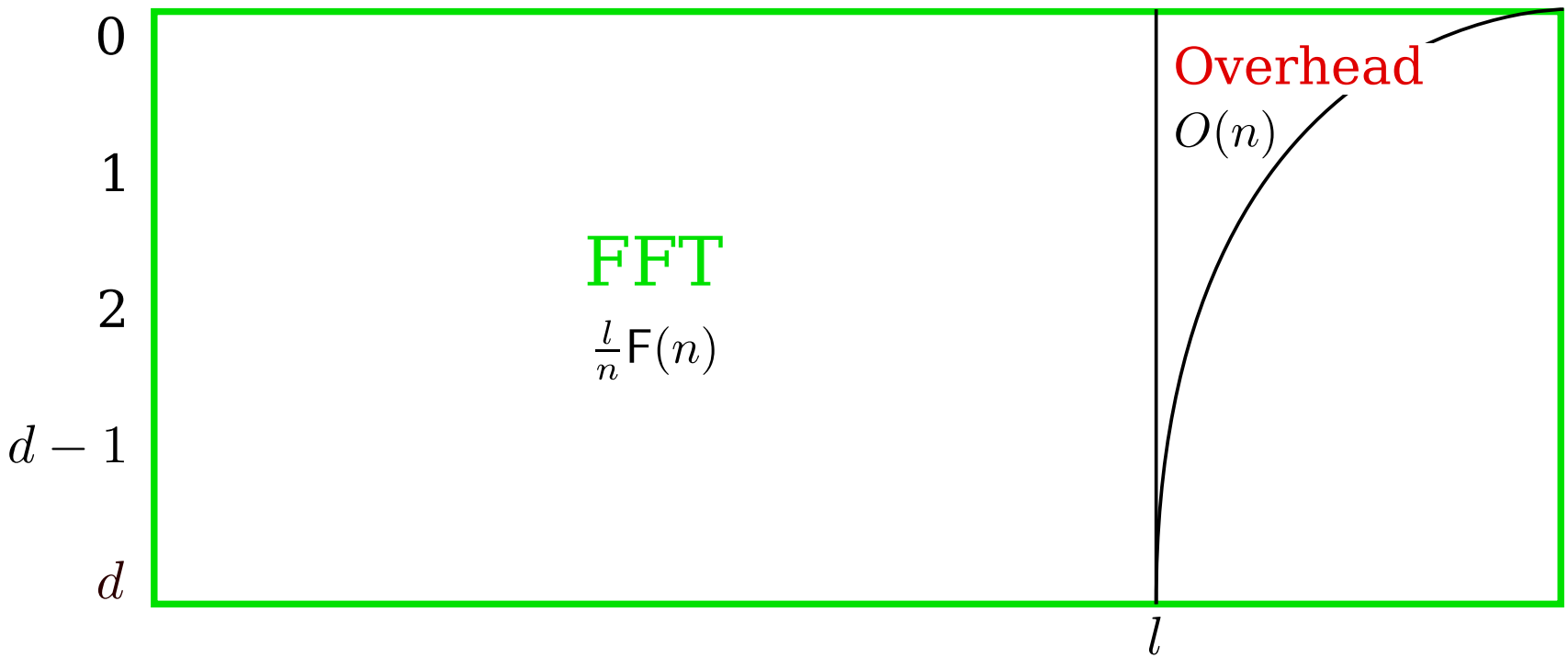
Complexité



Complexité



Complexité



Questions ?

Merci de votre attention

Références

[CT65] James W. Cooley and John W. Tukey.
An algorithm for the machine calculation of complex Fourier series.
19(90) :297–301, April 1965.

[Lar16] Robin Larrieu.
The Truncated Fourier Transform for mixed radices.
working paper or preprint, December 2016.

[vdH04] J. van der Hoeven.
The truncated Fourier transform and applications.
In J. Gutierrez, editor, *Proc. ISSAC 2004*, pages 290–296, Univ. of Cantabria, Santander, Spain, July 4–7 2004.