

Structured FFT and TFT: symmetric and lattice polynomials

JORIS VAN DER HOEVEN

Laboratoire d'informatique
UMR 7161 CNRS
École polytechnique
91128 Palaiseau Cedex, France

Email: vdhoeven@lix.polytechnique.fr

ROMAIN LEBRETON

LIRMM
UMR 5506 CNRS
Université de Montpellier II
Montpellier, France

Email: lebreton@lirmm.fr

ÉRIC SCHOST

Computer Science Department
Western University
London, Ontario
Canada

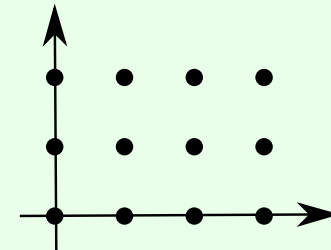
Email: eschost@uwo.ca

Multiplication of multivariate polynomial:

Dense polynomials:

Reduction to univariate polynomial multiplication

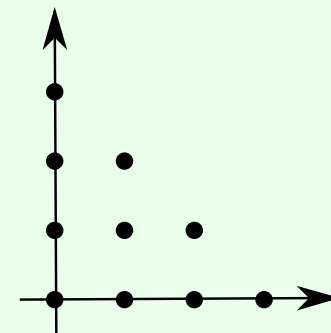
⇒ quasi-optimal algorithm



Semi-dense polynomials:

Truncated Fourier Transform techniques

⇒ quasi-optimal algorithm



Structured polynomials:

- Invariant polynomials
- Lattice polynomials

e.g. $1 + 2(X + Y) + XY + 3(X^2 + Y^2)$

e.g. $1 + Y^2 + XY + X^2 + X^2Y^2$

Our results

Existing body of works:

Crystallographic FFT:	[TEN EYCK, '73], [KUDLICKI <i>et al.</i> , '07]
Equivariant FFT:	[AUSLANDER <i>et al.</i> , '96], [JOHNSON, XU, '07]
FFT over lattices:	[GUESSOUM, MERSEREAU '86], [VINCE, ZHENG '07], [BERGMANN '12]

Theorem. (Hoeven, L., Schost, '13)

Cost of Symmetric FFT for $G \subseteq \mathfrak{S}_n$

$$\frac{1}{|G|} F(d, n) + \mathcal{O}(d^n)$$

Cost of Lattice FFT

$$\frac{1}{\text{vol}(\Lambda)} F(\mathbf{d}) + \mathcal{O}(d^n)$$

where F is the cost of the classical dense FFT.

Applications: Celestial mechanics (TRIP), ...

Outline

1. *The classical FFT*
 - a. Univariate FFT
 - b. Multivariate dense FFT

2. Invariant polynomials: the symmetric FFT

3. Lattice polynomials
 - a. Reduction of lattice polynomials to dense polynomials
 - b. Lattice polynomial multiplication

4. Conclusion

Invariant polynomial multiplication

Notations

- $\mathbb{K}[\mathbf{x}]_d := \mathbb{K}[x_1, \dots, x_n]_d$: polynomials in n variables of degree $< d = 2^\ell$
- \mathfrak{S}_n acts on $\mathbb{K}[\mathbf{x}]$: $P^\sigma(x_1, \dots, x_n) = P(x_{\sigma(1)}, \dots, x_{\sigma(n)})$ for $\sigma \in \mathfrak{S}_n$
- $\mathbb{K}[\mathbf{x}]^G$: invariant polynomials under $G \subseteq \mathfrak{S}_n$
- $\omega \in \mathbb{K}$: primitive d -th root of unity
- $\mathbb{N}_d^n := \{0, \dots, d-1\}^n$
- $[n]_s$ is the bit-reverse of n on s bits.

Example: $[13]_4 = [\overline{1101}^2]_4 = \overline{1011}^2 = 11$.

Objective

Compute efficiently the map

$$\text{FFT}_\omega: \begin{cases} \mathbb{K}[\mathbf{x}]_d & \longrightarrow & \mathbb{K}^{\mathbb{N}_d^n} \\ P & \longmapsto & (P(\omega^i))_{i \in \mathbb{N}_d^n} \end{cases}$$

when P is invariant under a group G of permutation.

Dense univariate FFT

Input: $P = \sum_{k=0}^{d-1} P_k x^k \in \mathbb{K}[x]_d$ with $d = 2^\ell$.

Direct Univariate FFT (Decimation-In-Time variant)

– Iterative in-place algorithm

$$(\mathbf{c}_k^0)_{0 \leq k < d} := (P_k)_{0 \leq k < d} \xrightarrow{\text{butterflies}} \mathbf{c}^1 \xrightarrow{\text{butterflies}} \cdots \xrightarrow{\text{butterflies}} (\mathbf{c}_k^\ell)_{0 \leq k < d} = (P(\omega^{[k]_s}))_{0 \leq k < d}$$

– Butterflies of step s with $1 \leq s \leq \ell$

$$\begin{pmatrix} \mathbf{c}_{i\delta+j}^s \\ \mathbf{c}_{(i+1)\delta+j}^s \end{pmatrix} = \begin{pmatrix} 1 & \omega^{[i]_s \delta} \\ 1 & -\omega^{[i]_s \delta} \end{pmatrix} \begin{pmatrix} \mathbf{c}_{i\delta+j}^{s-1} \\ \mathbf{c}_{(i+1)\delta+j}^{s-1} \end{pmatrix}$$

where $\delta := 2^{\ell-s}$.

– \mathbf{B}^s is the *global* matrix of butterflies: $\mathbf{c}^s = \mathbf{B}^s \mathbf{c}^{s-1}$.

Arithmetic complexity of univariate FFT for $d = 2^\ell$

$$F(d) := 3/2 d \log(d)$$

Dense multivariate FFT

Input: $P = \sum_{\mathbf{k} \in \mathbb{N}_d^n} P_{\mathbf{k}} x^{\mathbf{k}} \in \mathbb{K}[x_1, \dots, x_n]_d$ with $d = 2^\ell$.

Direct multivariate FFT (Decimation-In-Time variant)

– Iterative in-place algorithm

$$(\mathbf{c}_{\mathbf{k}}^0)_{\mathbf{k} \in \mathbb{N}_d^n} := (P_{\mathbf{k}})_{\mathbf{k} \in \mathbb{N}_d^n} \xrightarrow{\text{butterflies}} \mathbf{c}^1 \xrightarrow{\text{butterflies}} \dots \xrightarrow{\text{butterflies}} (\mathbf{c}_{\mathbf{k}}^\ell)_{\mathbf{k} \in \mathbb{N}_d^n} = (P(\omega^{[\mathbf{k}]_s}))_{\mathbf{k} \in \mathbb{N}_d^n}$$

– Multivariate butterfly of step s with $1 \leq s \leq \ell$:

$$\mathbf{B}^s = \mathbf{B}^{s,n} \dots \mathbf{B}^{s,1}$$

where $\mathbf{B}^{s,i}$ is the s -th step of the univariate FFT on x_i .

Arithmetic complexity of multivariate FFT for $d = 2^\ell$

$$F(d, n) := 3/2 d^n \log(d^n)$$

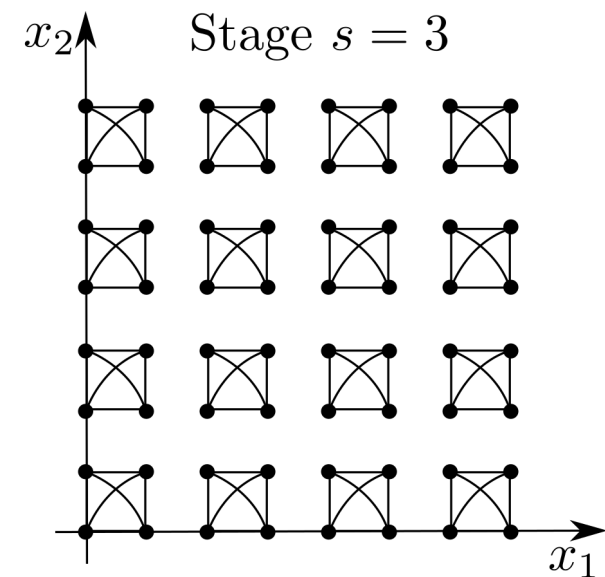
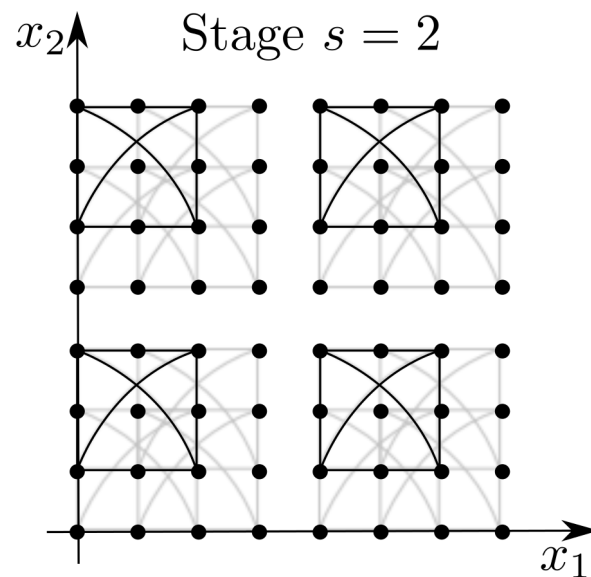
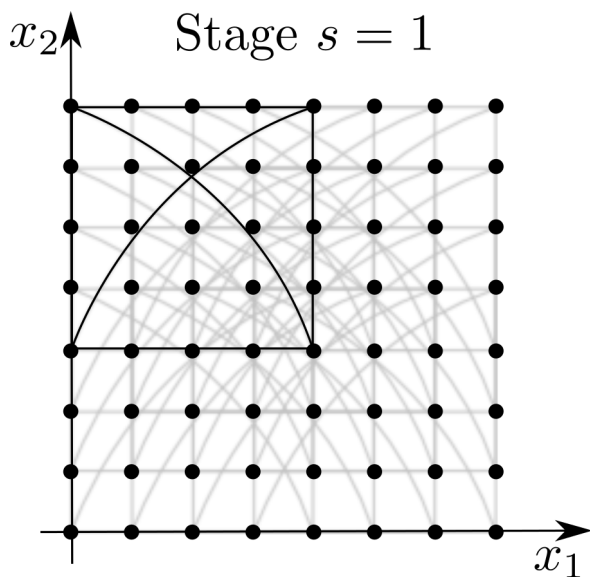
Dense multivariate FFT - Example

Example for $\mathbb{K}[x_1, x_2]_8$:

– 2-dimensional butterfly: At step 2, for any $j_1, j_2 \in \mathbb{N}_2$,

$$\begin{pmatrix} c_{(j_1, j_2)}^1 \\ c_{(2+j_1, j_2)}^1 \\ c_{(j_1, 2+j_2)}^1 \\ c_{(2+j_1, 2+j_2)}^1 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}}_{\mathbf{B}^2 = \mathbf{B}^{2,2} \cdot \mathbf{B}^{2,1}} \begin{pmatrix} c_{(j_1, j_2)}^0 \\ c_{(2+j_1, j_2)}^0 \\ c_{(j_1, 2+j_2)}^0 \\ c_{(2+j_1, 2+j_2)}^0 \end{pmatrix}$$

2-dimensional FFT:



Outline

1. The classical FFT
 - a. Univariate FFT
 - b. Multivariate dense FFT

2. *Invariant polynomials: the symmetric FFT*

3. Lattice polynomials
 - a. Reduction of lattice polynomials to dense polynomials
 - b. Lattice polynomial multiplication

4. Conclusion

Symmetric multivariate FFT

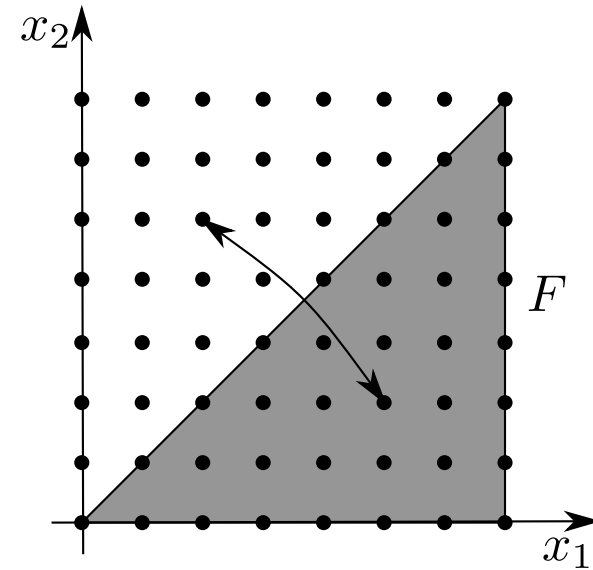
Symmetry on the input coefficients

- Example:

$$P_{(\mathbf{k}_1, \mathbf{k}_2)} = P_{(\mathbf{k}_2, \mathbf{k}_1)} \text{ for } P \in \mathbb{K}[x_1, x_2]^{\mathfrak{S}_2}$$

- Fundamental domain

$$F := \{\mathbf{i} \in \mathbb{N}_d^n : \forall g \in G, \mathbf{i} \geq_{\text{lex}} g(\mathbf{i})\}$$



Lemma. (Hoeven, L., Schost, '13)

If $\mathbf{c}^0 = (P_{\mathbf{k}})_{\mathbf{k} \in \mathbb{N}_d^n}$ is G -symmetric and $G \subseteq \mathfrak{S}_n$ then any \mathbf{c}^s is G -symmetric.

Symmetric multivariate FFT

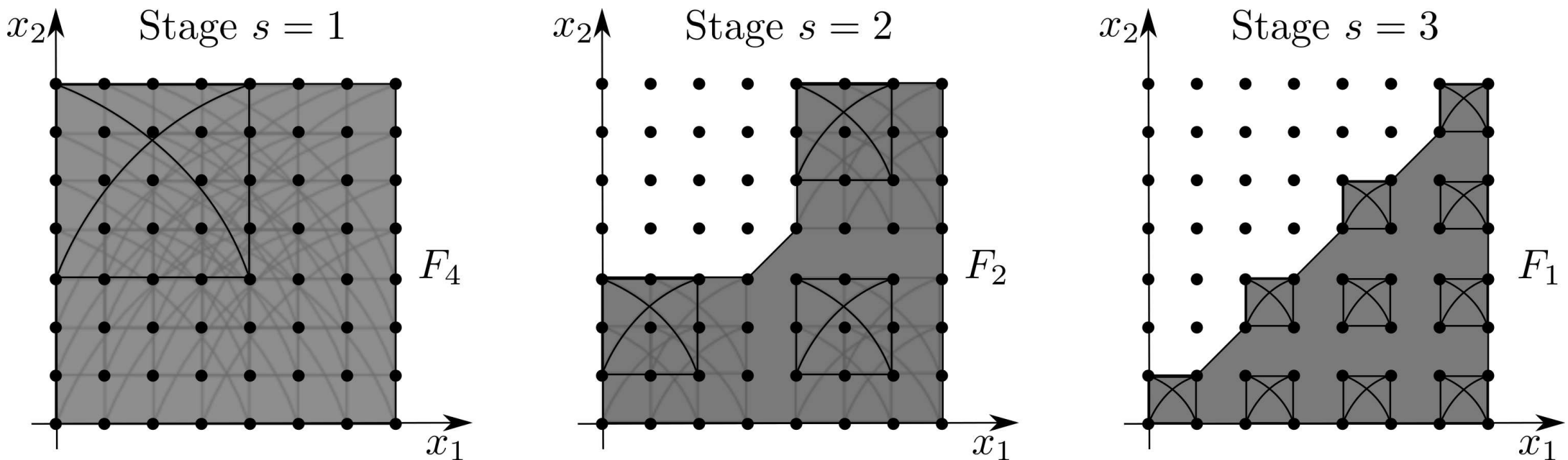
Observation:

c^s is G -symmetric \Rightarrow we need only c_k^s for $k \in F$.

Algorithm SymmetricFFT:

“Compute only butterflies that intersect F ”

Example:



Symmetric FFT - Cost

Theorem. (Hoeven, L., Schost, '13)

For fixed n and G , and for $d \rightarrow \infty$, the symmetric FFT can be computed in

$$\frac{1}{|G|} F(d, n) + \mathcal{O}(d^n)$$

arithmetic operations in \mathbb{K} .

Remark: Operations with groups are counted $\mathcal{O}(1)$.

Outline

1. The classical FFT
 - a. Univariate FFT
 - b. Multivariate dense FFT
2. Invariant polynomials: the symmetric FFT
3. *Lattice polynomials*
 - a. Reduction of lattice polynomials to dense polynomials
 - b. Lattice polynomial multiplication
4. Conclusion

Lattice polynomials

Notations.

Lattice $\Lambda := \lambda_1 \mathbb{Z} + \dots + \lambda_n \mathbb{Z} \subseteq \mathbb{Z}^n$

$\mathbb{K}[\mathbf{x}]_\Lambda := \{P \in \mathbb{K}[\mathbf{x}] : \text{support}(P) \subseteq \Lambda\}$

Objective

Multiply efficiently polynomials in $\mathbb{K}[\mathbf{x}]_\Lambda$

using FFT techniques.

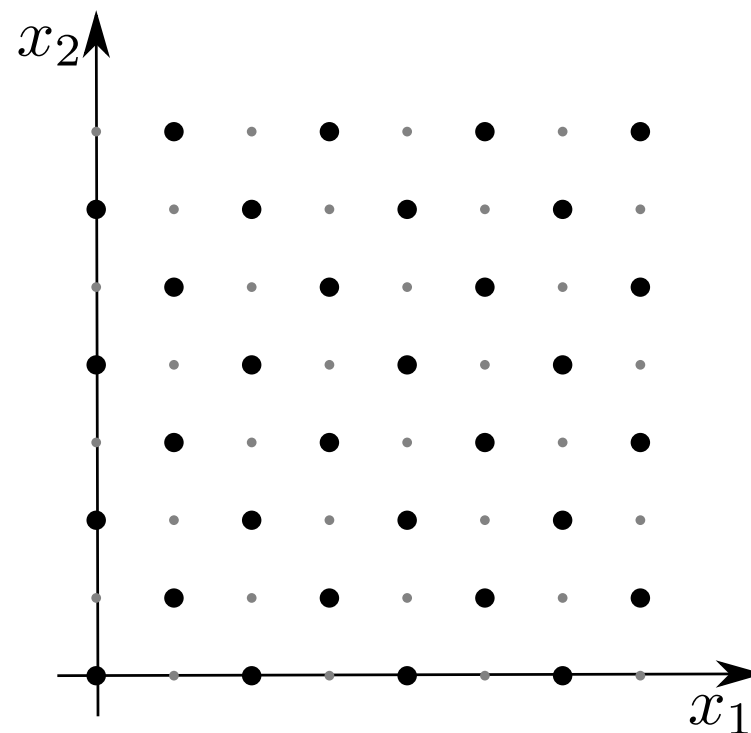


Figure. Example of lattice support Λ

Remark. Lattice polynomials are invariant polynomials

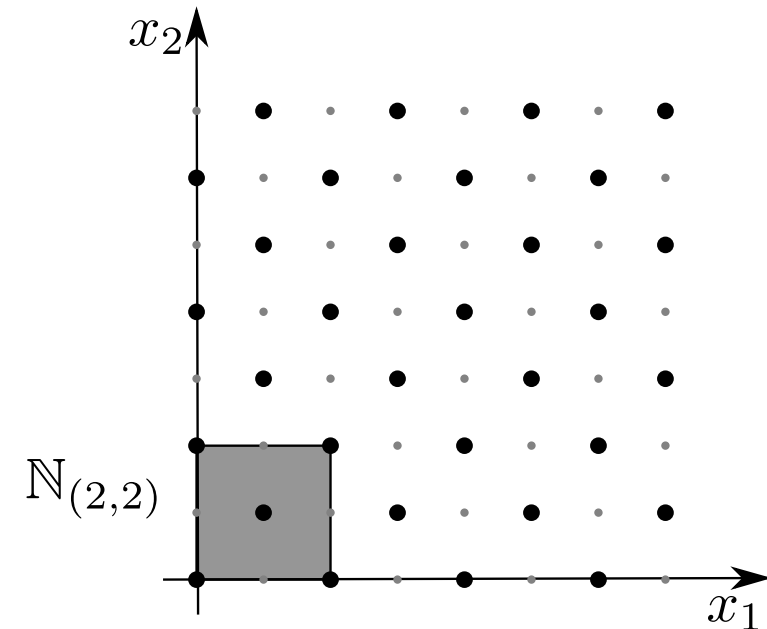
Example: Let $\Lambda = (2, 0) \mathbb{Z} + (1, 1) \mathbb{Z} \subseteq \mathbb{Z}^2$.

Then $P \in \mathbb{K}[\mathbf{x}]_\Lambda \iff P(x_1, x_2) = P(-x_1, -x_2)$.

Lattice FFT - Basic domain

Basic domain:

$\mathbb{N}_{\mathbf{p}} := \mathbb{N}_{p_1} \times \cdots \times \mathbb{N}_{p_n}$ where $p_i > 0$ minimal s.t. $x_i^{p_i} \in \mathbb{K}[\mathbf{x}]_{\Lambda}$.



Proposition. (Hoeven, L., Schost, '13)

There exists a basis $(\lambda'_1, \dots, \lambda'_n)$ of Λ and $q_1 \mid q_2 \mid \dots \mid q_n$ s.t. Φ is a \mathbb{K} -algebra isomorphism

$$\Phi: \mathbb{K}[\mathbf{y}] / (y_1^{q_1} - 1, \dots, y_n^{q_n} - 1) \longrightarrow \mathbb{K}[\mathbf{x}]_{\Lambda} / (x_1^{p_1} - 1, \dots, x_n^{p_n} - 1)$$

$$y_i \longmapsto x^{\lambda'_i}$$

Corollary.

Lattice multiplication
modulo $(x_1^{p_1} - 1, \dots, x_n^{p_n} - 1)$

\Leftrightarrow

Dense multiplication
modulo $(y_1^{q_1} - 1, \dots, y_n^{q_n} - 1)$

Lattice FFT - General case

Algorithm LatticeProduct

Input: $P_1, P_2 \in \mathbb{K}[\mathbf{x}]_{\Lambda, d/2}$ with $d_i = p_i 2^\ell$

Output: $P_1 \cdot P_2 \in \mathbb{K}[\mathbf{x}]_{\Lambda, d}$

1. ℓ FFT steps on P_1 and P_2 :

$$\mathbb{K}[\mathbf{x}]_{\Lambda, d} / (x_1^{d_1} - 1, \dots, x_n^{d_n} - 1) \longrightarrow (\mathbb{K}[\mathbf{x}]_{\Lambda, p} / (x_1^{p_1} - 1, \dots, x_n^{p_n} - 1))^{2^{n\ell}}$$

2. ...

Proposition.

Let c_k^s be the coefficients after s steps of Decimation-In-Frequency FFT. Then

$$\Phi: \begin{cases} \mathbb{K}[\mathbf{x}]_d / (x_1^d - 1, \dots, x_n^d - 1) & \rightarrow [\mathbb{K}[\mathbf{x}]_\delta / (x_1^\delta - 1, \dots, x_n^\delta - 1)]^{2^{ns}} \\ P & \mapsto \left(\sum_{j \in \mathbb{N}_\delta^n} c_{i\delta+j}^s \mathbf{x}^j \right)_{i \in \mathbb{N}_{2^s}^n} \end{cases}$$

is a \mathbb{K} -algebra isomorphism where $\delta = d/2^s$.

Lattice FFT - General case

Algorithm LatticeProduct

Input: $P_1, P_2 \in \mathbb{K}[\mathbf{x}]_{\Lambda, d/2}$ with $d_i = p_i 2^\ell$

Output: $P_1 \cdot P_2 \in \mathbb{K}[\mathbf{x}]_{\Lambda, d}$

1. ℓ FFT steps on P_1 and P_2 :

$$\mathbb{K}[\mathbf{x}]_{\Lambda, d} / (x_1^{d_1} - 1, \dots, x_n^{d_n} - 1) \longrightarrow (\mathbb{K}[\mathbf{x}]_{\Lambda, p} / (x_1^{p_1} - 1, \dots, x_n^{p_n} - 1))^{2^{n\ell}}$$

2. ...

+ Compute only butterflies included in Λ

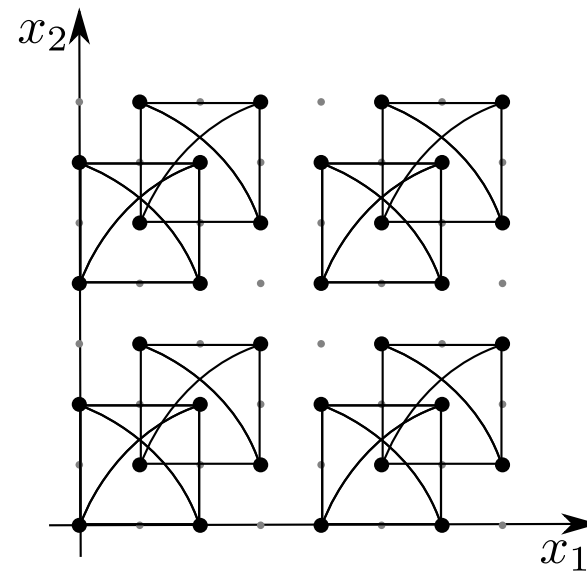


Figure. Butterflies on the lattice

Lattice FFT - General case

Algorithm LatticeProduct

Input: $P_1, P_2 \in \mathbb{K}[\mathbf{x}]_{\Lambda, d/2}$ with $d_i = p_i 2^\ell$

Output: $P_1 \cdot P_2 \in \mathbb{K}[\mathbf{x}]_{\Lambda, d}$

1. ℓ FFT steps on P_1 and P_2 :

$$\mathbb{K}[\mathbf{x}]_{\Lambda, d} / (x_1^{d_1} - 1, \dots, x_n^{d_n} - 1) \longrightarrow (\mathbb{K}[\mathbf{x}]_{\Lambda, p} / (x_1^{p_1} - 1, \dots, x_n^{p_n} - 1))^{2^{n\ell}}$$

2. $2^{n\ell}$ multiplications in $\mathbb{K}[\mathbf{x}]_{\Lambda, p} / (x_1^{p_1} - 1, \dots, x_n^{p_n} - 1) \simeq \mathbb{K}[\mathbf{y}] / (y_1^{q_1} - 1, \dots, y_n^{q_n} - 1)$

3. ...

Earlier Corollary.

Lattice multiplication

modulo $(x_1^{p_1} - 1, \dots, x_n^{p_n} - 1)$

\Leftrightarrow

Dense multiplication

modulo $(y_1^{q_1} - 1, \dots, y_n^{q_n} - 1)$

Lattice FFT - General case

Algorithm LatticeProduct

Input: $P_1, P_2 \in \mathbb{K}[\mathbf{x}]_{\Lambda, d/2}$ with $d_i = p_i 2^\ell$

Output: $P_1 \cdot P_2 \in \mathbb{K}[\mathbf{x}]_{\Lambda, d}$

1. ℓ FFT steps on P_1 and P_2 :

$$\mathbb{K}[\mathbf{x}]_{\Lambda, d} / (x_1^{d_1} - 1, \dots, x_n^{d_n} - 1) \longrightarrow (\mathbb{K}[\mathbf{x}]_{\Lambda, p} / (x_1^{p_1} - 1, \dots, x_n^{p_n} - 1))^{2^{n\ell}}$$

2. $2^{n\ell}$ multiplications in $\mathbb{K}[\mathbf{x}]_{\Lambda, p} / (x_1^{p_1} - 1, \dots, x_n^{p_n} - 1) \simeq \mathbb{K}[\mathbf{y}] / (y_1^{q_1} - 1, \dots, y_n^{q_n} - 1)$

3. Inverse FFT

Lattice FFT - Cost

Algorithm LatticeProduct

Input: $P_1, P_2 \in \mathbb{K}[\mathbf{x}]_{\Lambda, d/2}$ with $d_i = p_i 2^\ell$

Output: $P_1 \cdot P_2 \in \mathbb{K}[\mathbf{x}]_{\Lambda, d}$

1. ℓ FFT steps on P_1 and P_2
2. $2^{n\ell}$ dense multiplications in $\mathbb{K}[\mathbf{y}]/(y_1^{q_1} - 1, \dots, y_n^{q_n} - 1)$ after isomorphism
3. Inverse FFT

Theorem. (Hoeven, L., Schost, '13)

For a fixed lattice Λ and $d \rightarrow \infty$, Algorithm LatticeProduct costs

$$\mathsf{T}(\mathbf{d}) = \frac{1}{\text{vol}(\Lambda)} \mathsf{F}(\mathbf{d}) + \mathcal{O}(d^n)$$

where $\mathsf{F}(d)$ is the cost of the classical variant of LatticeProduct.

Remark: Operations with lattices are counted $\mathcal{O}(1)$.

Conclusion

Conclusion:

- Quasi-optimal algorithms for multiplication of
 - Invariant polynomials with $G \subseteq \mathfrak{S}_n$
 - Lattice polynomials
- No implementation yet
 - ↪ Needs an improved study of operations with groups and lattices

Perspectives:

- Symmetric / Lattice Truncated Fourier Transforms
- More general groups $G \not\subseteq \mathfrak{S}_n$
- Other evaluation-interpolation models: Karatsuba and Toom-Cook

Thank you
for your attention