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

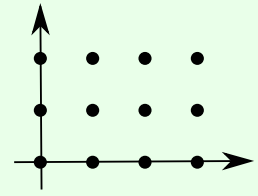
Email: eschost@uwo.ca

Context

Multiplication of multivariate polynomial:

Dense polynomials:

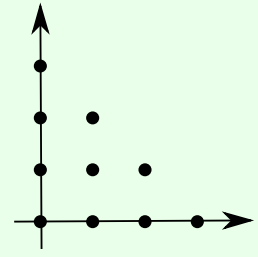
Reduction to univariate polynomial multiplication



$\Rightarrow$ quasi-optimal algorithm

Semi-dense polynomials:

Truncated Fourier Transform techniques



$\Rightarrow$ quasi-optimal algorithm

Structured polynomials:

- Invariant polynomials — *e.g.* $1+2\,(X+Y)+XY+3\,(X^2+Y^2)$
- Lattice polynomials — *e.g.* $1+Y^2+XY+X^2+X^2Y^2$

Our results

Existing body of works:

Crystallographic FFT:	[Ten Eyck, '73], [Kudlicki *et al.*, '07]
Equivariant FFT:	[Auslander *et al.*, '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 $G \subseteq \mathfrak{S}_n$	Cost of Lattice FFT
$\frac{1}{\lvert G\rvert}\mathsf{F}(d,n)+\mathcal{O}(d^n)$	$\frac{1}{\mathrm{vol}(\Lambda)}\mathsf{F}(\boldsymbol{d})+\mathcal{O}(d^n)$

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

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

$$\mathrm{FFT}_\omega : \left\{ \begin{array}{ccc} \mathbb{K}[\boldsymbol{x}]_d & \longrightarrow & \mathbb{K}^{\mathbb{N}_d^n} \\ P & \longmapsto & (P(\omega^{\boldsymbol{i}}))_{\boldsymbol{i} \in \mathbb{N}_d^n} \end{array} \right.$$

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

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

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

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

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

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

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

$$\mathsf{F}(d) := 3/2\, d \log(d)$$

Dense multivariate FFT

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

Direct multivariate FFT (Decimation-In-Time variant)

– Iterative in-place algorithm

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

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

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

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

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

$$\mathsf{F}(d, n) := 3/2\, d^n \log\left(d^n\right)$$

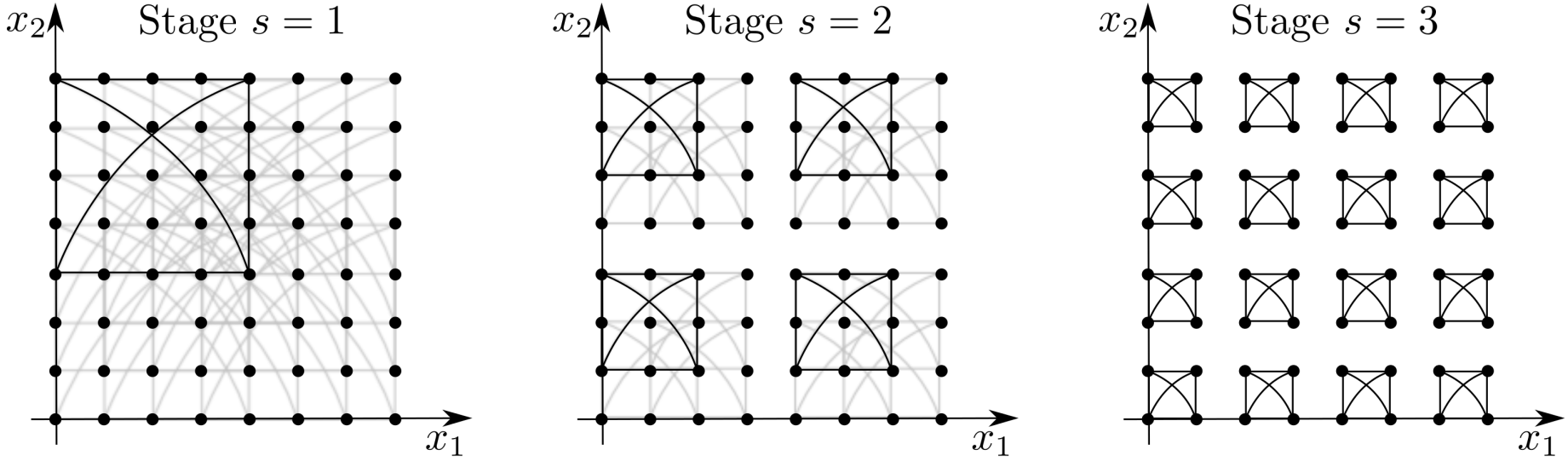
Dense multivariate FFT - Example

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

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

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

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

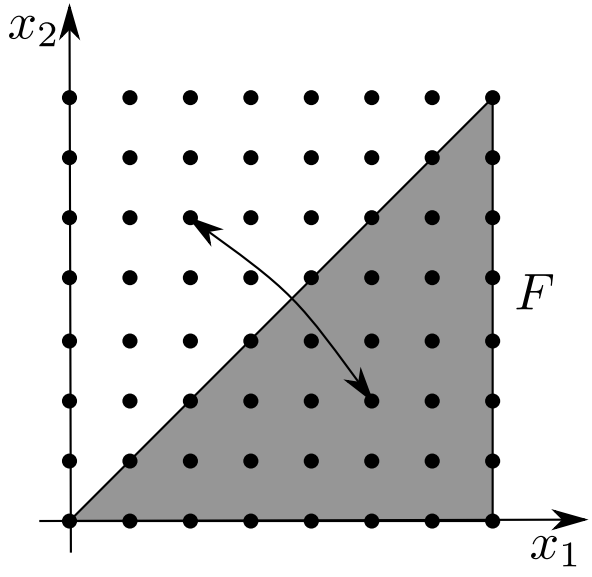
Symmetry on the input coefficients

- Example:

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

- Fundamental domain

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



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

If $\boldsymbol{c}^0$ is G-symmetric and $G \subseteq \mathfrak{S}_n$ then any $\boldsymbol{c}^s$ is G-symmetric.

Symmetric multivariate FFT

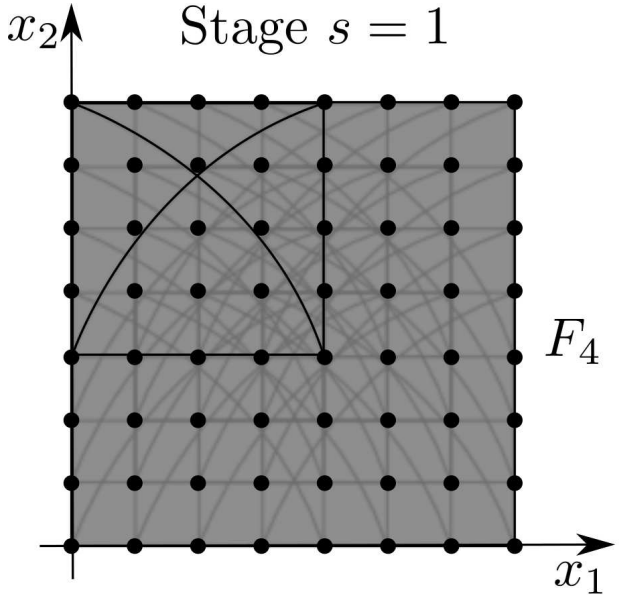
Observation:

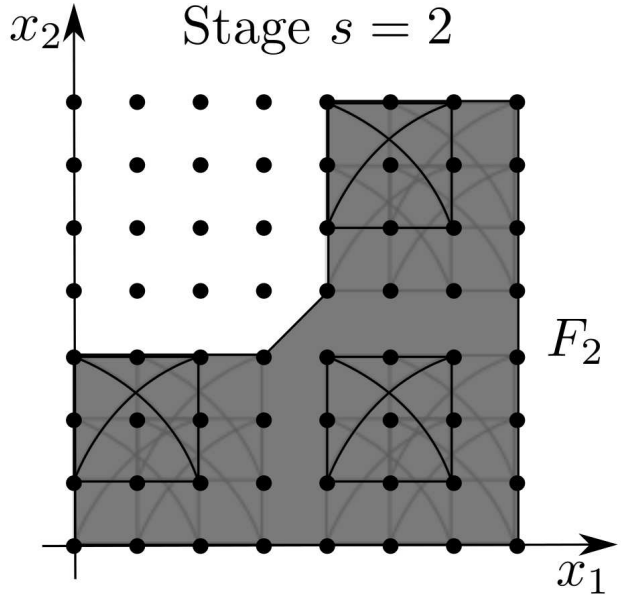
$\boldsymbol{c}^s$ is G-symmetric $\Rightarrow$ we need only $\boldsymbol{c}^s_{\boldsymbol{k}}$ for $\boldsymbol{k} \in F$.

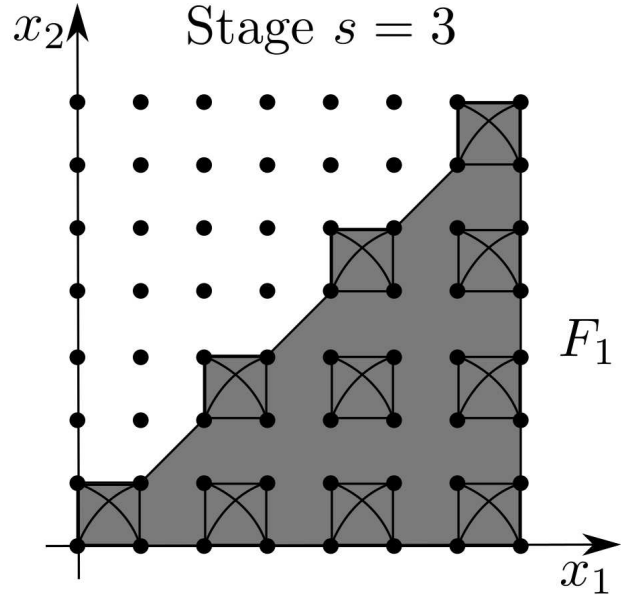
Algorithm SymmetricFFT:

"Compute only butterflies that intersect F"

Example:







Symmetric FFT - Cost

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

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

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

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 $\Lambda := \boldsymbol{\lambda}_1 \mathbb{Z} + \cdots + \boldsymbol{\lambda}_n \mathbb{Z} \subseteq \mathbb{Z}^n$

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

Objective

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

using FFT techniques.

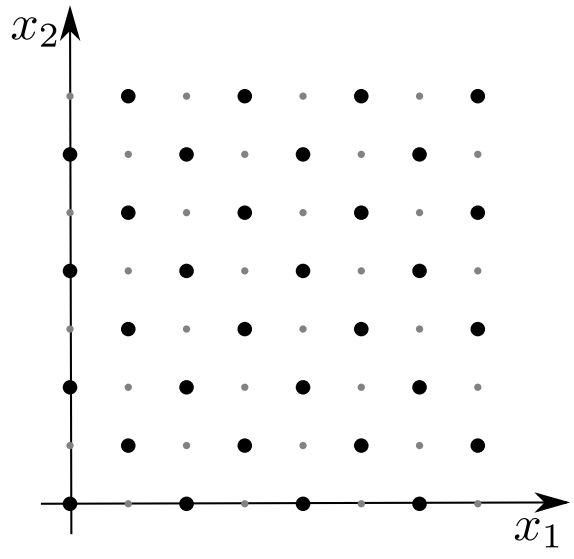


Figure. Example of lattice support Λ

Remark. Lattice polynomials are invariant polynomials

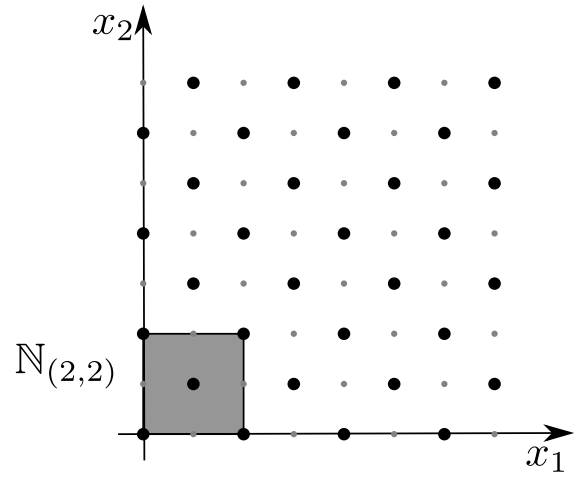
Example: Let $\Lambda = (2,0)\,\mathbb{Z} + (1,1)\,\mathbb{Z} \subseteq \mathbb{Z}^2$.

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

Lattice FFT - Basic domain

Basic domain:

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



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

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

$$\begin{array}{rccc} \Phi\colon & \mathbb{K}[\boldsymbol{y}]/(y_1^{q_1}-1, \ldots, y_n^{q_n}-1) & \longrightarrow & \mathbb{K}[\boldsymbol{x}]_\Lambda/(x_1^{p_1}-1, \ldots, x_n^{p_n}-1) \\ & y_i & \longmapsto & \boldsymbol{x}^{\boldsymbol{\lambda}'_i} \end{array}.$$

Corollary.

Lattice multiplication modulo $(x_1^{p_1}-1, \ldots, x_n^{p_n}-1)$	$\Leftrightarrow$	Dense multiplication modulo $(y_1^{q_1}-1, \ldots, y_n^{q_n}-1)$

Lattice FFT - General case

Algorithm LatticeProduct

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

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

1. ℓ FFT steps on P_1 and P_2:

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

2. ...

Proposition.

Let $\boldsymbol{c}_{\boldsymbol{k}}^s$ be the coefficients after s steps of Decimation-In-Frequency FFT. Then

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

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

Lattice FFT - General case

Algorithm LatticeProduct

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

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

1. ℓ FFT steps on P_1 and P_2:

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

2. ...

+ Compute only butterflies included in Λ

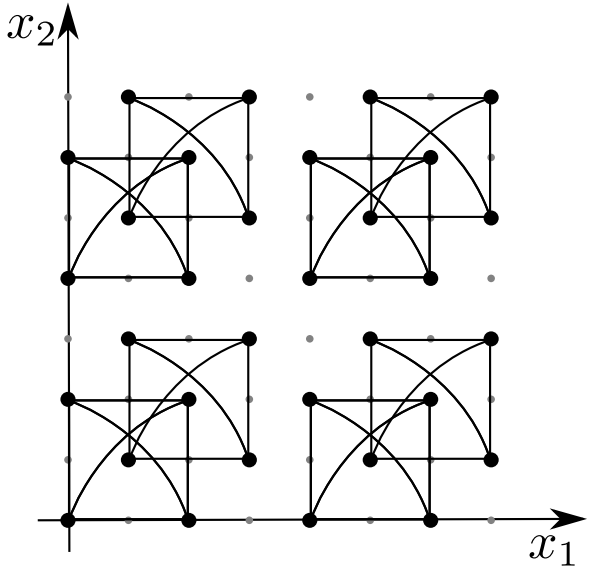


Figure. Butterflies on the lattice

Lattice FFT - General case

Algorithm LatticeProduct

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

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

1. ℓ FFT steps on P_1 and P_2:

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

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

3. ...

Earlier Corollary.

Lattice multiplication modulo $(x_1^{p_1}-1,\ldots,x_n^{p_n}-1)$ $\Leftrightarrow$ Dense multiplication modulo $(y_1^{q_1}-1,\ldots,y_n^{q_n}-1)$

Lattice FFT - General case

Algorithm LatticeProduct

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

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

1. ℓ FFT steps on P_1 and P_2:

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

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

3. Inverse FFT

Lattice FFT - Cost

Algorithm LatticeProduct

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

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

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

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

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

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

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

  $\rightsquigarrow$ 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