Quasi-Optimal Multiplication of Linear Differential Operators

Alexandre Benoit [1], Alin Bostan [2] and Joris van der Hoeven [3]

[1] Éducation nationale (France)

[2] INRIA (France)

[3] CNRS, École Polytechnique (France)

Séminaire Caramel
November, 23th 2012

I Introduction

Product of Linear Differential Operators

L and K: linear differential operators with polynomial coefficients in $\mathbb{K}[x]\langle\partial\rangle$. The product KL is given by the relation of composition

$$\forall f \in \mathbb{K}[x], \qquad KL \cdot f = K \cdot (L \cdot f).$$

Product of Linear Differential Operators

L and K: linear differential operators with polynomial coefficients in $\mathbb{K}[x]\langle\partial\rangle$. The product KL is given by the relation of composition

$$\forall f \in \mathbb{K}[x], \qquad KL \cdot f = K \cdot (L \cdot f).$$

The commutation of this product is given by the Leibniz rule:

$$\partial x = x\partial + 1.$$

Complexity of the Product of Linear Differential Operators

The product of differential operator is a complexity yardstick.
The complexity of more involved, higher-level, operations on linear differential operators can be reduced to that of multiplication:

- LCLM, GCRD (van der Hoeven 2011)
- Hadamard product
- other closure properties for differential operators ...

Previous complexity results

Product of operators in $\mathbb{K}[x]\langle\partial\rangle$ of orders $< r$ with polynomial coefficients of degrees $< d$ (i.e bidegrees less than (d,r)):

- Naive algorithm: $\mathcal{O}(d^2r^2\min(d,r))$ ops
- Takayama algorithm: $\tilde{\mathcal{O}}(dr\min(d,r))$ ops
- Van der Hoeven algorithm (2002): $\mathcal{O}((d+r)^\omega)$ ops using evaluations and interpolations.

ω is a feasible exponent for matrix multiplication $(2 \leqslant \omega \leqslant 3)$
$\tilde{\mathcal{O}}$ indicates that polylogarithmic factors are neglected.

Complexities for Unballanced Product

van der Hoeven 2011 + bound given by Bostan et al (ISSAC 2012)

Fast algorithms for LCLM or GCRD for operators of bidegrees less than (r,r) can be reduced to the multiplication of operators with polynomial coefficients of bidegrees (r^2,r).

Complexities for Unballanced Product

van der Hoeven 2011 + bound given by Bostan et al (ISSAC 2012)

Fast algorithms for LCLM or GCRD for operators of bidegrees less than (r,r) can be reduced to the multiplication of operators with polynomial coefficients of bidegrees (r^2,r).

Product of operators of bidegrees less than (r^2,r)

- Naive algorithm: $\mathcal{O}(r^7)$ ops
- Takayama algorithm: $\tilde{\mathcal{O}}(r^4)$ ops
- Van der Hoeven algorithm: $\mathcal{O}(r^{2\omega})$ ops

Contributions: New Algorithm for Unbalanced Product

New algorithm[1] for the product of operators in $\mathbb{K}[x]\langle\partial\rangle$ of bidegree less than (d,r) in

$$\tilde{\mathcal{O}}(dr\min(d,r)^{\omega-2}).$$

1

[BenoitBostanvanderHoeven, 2012] B. and Bostan and van der Hoeven.
Quasi-Optimal Multiplication of Linear Differential Operators,
FOCS 2012.

Contributions: New Algorithm for Unbalanced Product

New algorithm[1] for the product of operators in $\mathbb{K}[x]\langle\partial\rangle$ of bidegree less than (d,r) in

$$\tilde{\mathcal{O}}(dr\min(d,r)^{\omega-2}).$$

In the important case $d \geqslant r$, this complexity reads $\tilde{\mathcal{O}}(dr^{\omega-1})$.
In particular, if $d = r^2$ the complexity becomes

$$\tilde{\mathcal{O}}(r^{\omega+1}) \text{ (instead of } \tilde{\mathcal{O}}(r^4)).$$

1

[BenoitBostanvanderHoeven, 2012] B. and Bostan and van der Hoeven.
Quasi-Optimal Multiplication of Linear Differential Operators,
FOCS 2012.

Outline of the proof

Main ideas

- Use an evaluation-interpolation strategy on the point $x^i \exp(\alpha x)$
- Use fast algorithm for performing Hermite interpolation
- $(d, r) \xleftrightarrow{\text{reflection}} (r, d)$ allows us to assume that $r \geqslant d$

II The van der Hoeven Algorithm

Skew Product: a Linear Algebra Problem

Recall : L is an operator of bidegree less than (d,r)
$L(x^\ell) \in \mathbb{K}[x]_{d+\ell-1}$.
$L(x^\ell)_i$ is defined by :

$$L(x^\ell) = L(x^\ell)_0 + L(x^\ell)_1 x + \cdots + L(x^\ell)_{d+\ell-1} x^{d+\ell-1}$$

Skew Product: a Linear Algebra Problem

Recall : L is an operator of bidegree less than (d,r)
$L(x^\ell) \in \mathbb{K}[x]_{d+\ell-1}$.
$L(x^\ell)_i$ is defined by :

$$L(x^\ell) = L(x^\ell)_0 + L(x^\ell)_1 x + \cdots + L(x^\ell)_{d+\ell-1} x^{d+\ell-1}$$

We define :

$$\Phi_L^{k+d,k} \quad = \quad \begin{pmatrix} L(1)_0 & \cdots & L(x^{k-1})_0 \\ \vdots & & \vdots \\ L(1)_{k+d-1} & \cdots & L(x^{k-1})_{k+d-1} \end{pmatrix} \in \mathbb{K}^{(k+d)\times k}$$

we clearly have

$$\Phi_{KL}^{k+2d,k} = \Phi_K^{k+2d,k+d} \Phi_L^{k+d,k}, \quad \text{for all } k \geqslant 0.$$

Study of Φ_L

We denote $L = l_0(\partial) + x l_1(\partial) + \cdots + x^{d-1} l_{d-1}(\partial)$ ($l_i \in \mathbb{K}[\partial]_r$)

$$\Phi_L^{k+d,k} := \begin{pmatrix} l_0(0) & l_0'(0) & \cdots & l_0^{(k-1)}(0) \\ l_1(0) & (l_1' + l_0)(0) & & \\ \vdots & \vdots & \vdots & \vdots \\ l_{d-1}(0) & (l_{d-1}' + l_{d-2})(0) & & \\ 0 & l_{d-1}(0) & & \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & l_{d-1}(0) \end{pmatrix}$$

Study of Φ_L

We denote $L = l_0(\partial) + x l_1(\partial) + \cdots + x^{d-1} l_{d-1}(\partial)$ ($l_i \in \mathbb{K}[\partial]_r$)

$$\Phi_L^{k+d,k} := \begin{pmatrix} l_0(0) & l_0'(0) & \cdots & l_0^{(k-1)}(0) \\ l_1(0) & (l_1' + l_0)(0) & & \\ \vdots & \vdots & \vdots & \vdots \\ l_{d-1}(0) & (l_{d-1}' + l_{d-2})(0) & & \\ 0 & l_{d-1}(0) & & \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & l_{d-1}(0) \end{pmatrix}$$

If L is an operator of bidegree (r,d), we can compute L from $\Phi_L^{r+d,r}$

Algorithm Using Evaluations-Interpolation

KL is an operator of bidegree less than $(2d,2r)$.
Then the operator KL can be recovered from the matrix

$$\Phi_{KL}^{2r+2d,2r}$$

Algorithm Using Evaluations-Interpolation

KL is an operator of bidegree less than $(2d,2r)$.
Then the operator KL can be recovered from the matrix

$$\Phi_{KL}^{2r+2d,2r}$$

We deduce an algorithm to compute KL.

1. (Evaluation) Computation of $\Phi_K^{2r+2d,2r+d}$ and of $\Phi_L^{2r+d,2r}$ from K and L.
2. (Inner multiplication) Computation of the matrix product.
3. (Interpolation) Recovery of KL from $\Phi_{KL}^{2r+2d,2r}$.

Algorithm Using Evaluations-Interpolation

KL is an operator of bidegree less than $(2d,2r)$.
Then the operator KL can be recovered from the matrix

$$\Phi_{KL}^{2r+2d,2r}$$

We deduce an algorithm to compute KL.

1. (Evaluation) Computation of $\Phi_K^{2r+2d,2r+d}$ and of $\Phi_L^{2r+d,2r}$ from K and L.
2. (Inner multiplication) Computation of the matrix product. $\mathcal{O}((d+r)^\omega)$ ops
3. (Interpolation) Recovery of KL from $\Phi_{KL}^{2r+2d,2r}$.

Fast Evaluation and Interpolation

A remark from Bostan, Chyzak and Le Roux (ISSAC 2008)

$$\begin{pmatrix} & & & l_0 & l_1 & \cdots & l_d \\ & & l_0' & l_0 + l_1' & \cdots & l_d' + l_{d-1} & l_d \\ & \iddots & & & & & \vdots \\ l_0^{(\ell-1)} & & \cdots & & & \cdots & l_d \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & & & & 0 \\ 1 & 1 & 0 & & & 0 \\ 1 & 2 & 1 & 0 & & 0 \\ 1 & 3 & 3 & 1 & 0 & 0 \\ \vdots & & & & \ddots & \end{pmatrix} \begin{pmatrix} 0 & \cdots & 0 & l_0 & l_1 & \cdots & l_d \\ \vdots & 0 & l_0' & l_1' & \cdots & l_d' & 0 \\ 0 & \iddots & & & \iddots & 0 & \vdots \\ l_0^{(\ell-1)} & l_1^{(\ell-1)} & \cdots & l_d^{(\ell-1)} & 0 & \vdots & 0 \end{pmatrix}$$

Fast Evaluation and Interpolation

A remark from Bostan, Chyzak and Le Roux (ISSAC 2008)

$$\begin{pmatrix} & & & l_0 & l_1 & \cdots & l_d \\ & & l'_0 & l_0 + l'_1 & \cdots & l'_d + l_{d-1} & l_d \\ & \iddots & & & & & \vdots \\ l_0^{(\ell-1)} & & \cdots & & & \cdots & l_d \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & & & & 0 \\ 1 & 1 & 0 & & & 0 \\ 1 & 2 & 1 & 0 & & 0 \\ 1 & 3 & 3 & 1 & 0 & 0 \\ \vdots & & & & \ddots & \end{pmatrix} \begin{pmatrix} 0 & \cdots & 0 & l_0 & l_1 & \cdots & l_d \\ \vdots & 0 & l'_0 & l'_1 & \cdots & l'_d & 0 \\ 0 & \iddots & & & \iddots & 0 & \vdots \\ l_0^{(\ell-1)} & l_1^{(\ell-1)} & \cdots & l_d^{(\ell-1)} & 0 & \vdots & 0 \end{pmatrix}$$

Applications:

- Computation of $\Phi_L^{r+d,r}$ from L in $\mathcal{O}((r+d)^\omega)$ (in $\tilde{\mathcal{O}}(rd)$ using structured matrices)

Fast Evaluation and Interpolation

A remark from Bostan, Chyzak and Le Roux (ISSAC 2008)

$$\begin{pmatrix} & & & l_0 & l_1 & \cdots & l_d \\ & & l'_0 & l_0 + l'_1 & \cdots & l'_d + l_{d-1} & l_d \\ & \iddots & & & & & \vdots \\ l_0^{(\ell-1)} & & \cdots & & & \cdots & l_d \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & & & & 0 \\ 1 & 1 & 0 & & & 0 \\ 1 & 2 & 1 & 0 & & 0 \\ 1 & 3 & 3 & 1 & 0 & 0 \\ \vdots & & & & \ddots & \end{pmatrix} \begin{pmatrix} 0 & \cdots & 0 & l_0 & l_1 & \cdots & l_d \\ \vdots & 0 & l'_0 & l'_1 & \cdots & l'_d & 0 \\ 0 & \iddots & & & \iddots & 0 & \vdots \\ l_0^{(\ell-1)} & l_1^{(\ell-1)} & \cdots & l_d^{(\ell-1)} & 0 & \vdots & 0 \end{pmatrix}$$

Applications:

- Computation of $\Phi_L^{r+d,r}$ from L in $\mathcal{O}((r+d)^\omega)$ (in $\tilde{\mathcal{O}}(rd)$ using structured matrices)
- Computation of L from $\phi_{r+d,r}(L)$ in $\mathcal{O}((r+d)^\omega)$ (in $\tilde{\mathcal{O}}(rd)$ using structured matrices)

Complexity of van der Hoeven Algorithm

Easy bound: If L and K are of bidegrees less than (d,r), KL is of bidegree less than $(2d,2r)$.

- Evaluation of $\Phi_L^{2r+d,2r}$ and $\Phi_K^{2r+2d,2r}$
- Matrix multiplication $\Phi_{KL}^{2r+2d,2r} = \Phi_K^{2d} \cdot \ \Phi_L^{2d+r}$
- Interpolation. From $\Phi_{KL}^{2d+2r,2r}$ to the coefficients of KL

Complexity of van der Hoeven Algorithm

Easy bound: If L and K are of bidegrees less than (d,r), KL is of bidegree less than $(2d,2r)$.

- Evaluation of $\Phi_L^{2r+d,2r}$ and $\Phi_K^{2r+2d,2r}$ $\tilde{\mathcal{O}}(rd)$ ops
- Matrix multiplication $\Phi_{KL}^{2r+2d,2r} = \Phi_K^{2d} \cdot \ \Phi_L^{2d+r}$ $\mathcal{O}((d+r)^\omega)$ ops
- Interpolation. From $\Phi_{KL}^{2d+2r,2r}$ to the coefficients of KL $\tilde{\mathcal{O}}(dr)$ ops

Complexity of van der Hoeven Algorithm

Easy bound: If L and K are of bidegrees less than (d,r), KL is of bidegree less than $(2d,2r)$.

- Evaluation of $\Phi_L^{2r+d,2r}$ and $\Phi_K^{2r+2d,2r}$ $\tilde{\mathcal{O}}(rd)$ ops
- Matrix multiplication $\Phi_{KL}^{2r+2d,2r} = \Phi_K^{2d} \cdot \Phi_L^{2d+r}$ $\mathcal{O}((d+r)^\omega)$ ops
- Interpolation. From $\Phi_{KL}^{2d+2r,2r}$ to the coefficients of KL $\tilde{\mathcal{O}}(dr)$ ops

Complexity of van der Hoeven algorithm if $d = r$: $\tilde{\mathcal{O}}(r^\omega)$ ☺

Complexity of van der Hoeven Algorithm

Easy bound: If L and K are of bidegrees less than (d,r), KL is of bidegree less than $(2d,2r)$.

- Evaluation of $\Phi_L^{2r+d,2r}$ and $\Phi_K^{2r+2d,2r}$ $\tilde{\mathcal{O}}(rd)$ ops
- Matrix multiplication $\Phi_{KL}^{2r+2d,2r} = \Phi_K^{2d} \cdot \ \Phi_L^{2d+r}$ $\mathcal{O}((d+r)^\omega)$ ops
- Interpolation. From $\Phi_{KL}^{2d+2r,2r}$ to the coefficients of KL $\tilde{\mathcal{O}}(dr)$ ops

Complexity of van der Hoeven algorithm if $d = r$: $\tilde{\mathcal{O}}(r^\omega)$ ☺

Complexity of van der Hoeven algorihtm if $d = r^2$: $\tilde{\mathcal{O}}(r^{2\omega})$ ☹

III New Algorithm for the Unbalanced Product ($r > d$)

Operate on Exponential Polynomials

- L also operates on $\mathbb{K}[x]e^{\alpha x}$ for every $\alpha \in \mathbb{K}$

Operate on Exponential Polynomials

- L also operates on $\mathbb{K}[x]e^{\alpha x}$ for every $\alpha \in \mathbb{K}$
- More specifically, writing

$$L = \sum_i L_i(x)\partial^i$$

we have:

$$\begin{aligned} L(Pe^{\alpha x}) &= L_{\ltimes\alpha}(P) \\ L_{\ltimes\alpha} &= \sum_i L_i(x)(\partial+\alpha)^i \end{aligned}$$

Operate on Exponential Polynomials

- L also operates on $\mathbb{K}[x]e^{\alpha x}$ for every $\alpha \in \mathbb{K}$
- More specifically, writing

$$L = \sum_i L_i(x)\partial^i$$

we have:

$$\begin{aligned} L(Pe^{\alpha x}) &= L_{\ltimes\alpha}(P) \\ L_{\ltimes\alpha} &= \sum_i L_i(x)(\partial+\alpha)^i \end{aligned}$$

$$\Phi^{k+d,k}_{L_{\ltimes\alpha}} := \begin{pmatrix} l_0(\alpha) & l_0'(\alpha) & \cdots & l_0^{(k-1)}(\alpha) \\ l_1(\alpha) & (l_1'+l_0)(\alpha) & & \\ \vdots & \vdots & \vdots & \vdots \\ l_{d-1}(\alpha) & (l_{d-1}'+l_{d-2})(\alpha) & & \\ 0 & l_{d-1}(\alpha) & & \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & l_{d-1}(\alpha) \end{pmatrix}$$

Hermite Evaluations and Interpolations

Sur la formule d'interpolation de *Lagrange*.

(Extrait d'une lettre de M. *Ch. Hermite* à M. *Borchardt*.)

Je me suis proposé de trouver un polynôme entier $F(x)$ de degré $n-1$, satisfaisant aux conditions suivantes:

$$\begin{array}{llll} F(a)=f(a), & F'(a)=f'(a), & \ldots & F^{\alpha-1}(a)=f^{\alpha-1}(a), \\ F(b)=f(b), & F'(b)=f'(b), & \ldots & F^{\beta-1}(b)=f^{\beta-1}(b), \\ \ldots & \ldots & \ldots & \ldots \\ F(l)=f(l), & F'(l)=f'(l), & \ldots & F^{\lambda-1}(l)=f^{\lambda-1}(l) \end{array}$$

où $f(x)$ est une fonction donnée. En supposant:

$$\alpha+\beta+\cdots+\lambda = n$$

la question comme on voit est déterminée, et conduira à une généralisation de la formule de *Lagrange* sur laquelle je présenterai quelques remarques.

Hermite Evaluations and Interpolations

Sur la formule d'interpolation de *Lagrange*.

(Extrait d'une lettre de M. *Ch. Hermite* à M. *Borchardt*.)

Je me suis proposé de trouver un polynôme entier $F(x)$ de degré $n-1$, satisfaisant aux conditions suivantes:

$$\begin{array}{llll} F(a)=f(a), & F'(a)=f'(a), & \ldots & F^{\alpha-1}(a)=f^{\alpha-1}(a), \\ F(b)=f(b), & F'(b)=f'(b), & \ldots & F^{\beta-1}(b)=f^{\beta-1}(b), \\ \cdots & \cdots & \cdots & \cdots \\ F(l)=f(l), & F'(l)=f'(l), & \ldots & F^{\lambda-1}(l)=f^{\lambda-1}(l) \end{array}$$

où $f(x)$ est une fonction donnée. En supposant:

$$\alpha+\beta+\cdots+\lambda = n$$

la question comme on voit est déterminée, et conduira à une généralisation de la formule de *Lagrange* sur laquelle je présenterai quelques remarques.

Fast algorithm for evaluations and interpolations in $\tilde{\mathcal{O}}(n)$ ops (Chin (1976)).

Hermite Evaluations and Interpolations

Sur la formule d'interpolation de *Lagrange*.

(Extrait d'une lettre de M. *Ch. Hermite* à M. *Borchardt*.)

Je me suis proposé de trouver un polynôme entier $F(x)$ de degré $n-1$, satisfaisant aux conditions suivantes:

$$\begin{array}{llll} F(a) = f(a), & F'(a) = f'(a), & \ldots & F^{\alpha-1}(a) = f^{\alpha-1}(a), \\ F(b) = f(b), & F'(b) = f'(b), & \ldots & F^{\beta-1}(b) = f^{\beta-1}(b), \\ \ldots & \ldots & \ldots & \ldots \\ F(l) = f(l), & F'(l) = f'(l), & \ldots & F^{\lambda-1}(l) = f^{\lambda-1}(l) \end{array}$$

où $f(x)$ est une fonction donnée. En supposant:

$$\alpha + \beta + \cdots + \lambda = n$$

la question comme on voit est déterminée, et conduira à une généralisation de la formule de *Lagrange* sur laquelle je présenterai quelques remarques.

Fast algorithm for evaluations and interpolations in $\tilde{\mathcal{O}}(n)$ ops (Chin (1976)).

Application:
Evaluations and interpolation of $\Phi^{2d,d}_{L_{\ltimes \alpha_i}}$ for $i \in [0..r/d-1]$ in $\tilde{\mathcal{O}}(rd)$ ops

Product using Multipoint Evaluations and Interpolation

We suppose $r > d$

Idea

For $p = \lceil r/d \rceil$, choose distinct $\alpha_0, \dots, \alpha_{p-1}$, and let L operates on

$$\mathbb{V}_k \quad = \quad \mathbb{K}[x]_k e^{\alpha_0 x} \oplus \cdots \oplus \mathbb{K}[x]_k e^{\alpha_{p-1} x}$$

We replace one multiplication of big matrices by several multiplications of smaller matrices

Product using Multipoint Evaluations and Interpolation

We suppose $r > d$

Idea

For $p = \lceil r/d \rceil$, choose distinct $\alpha_0, \ldots, \alpha_{p-1}$, and let L operates on

$$\mathbb{V}_k = \mathbb{K}[x]_k e^{\alpha_0 x} \oplus \cdots \oplus \mathbb{K}[x]_k e^{\alpha_{p-1} x}$$

We replace one multiplication of big matrices by several multiplications of smaller matrices

- Evaluations of $\Phi^{3d,2d}_{L_{\ltimes \alpha_i}}$ and $\Phi^{4d,3d}_{K_{\ltimes \alpha_i}}$, for i from 0 to $p-1$ ($\tilde{O}(dr)$)

Product using Multipoint Evaluations and Interpolation

We suppose $r > d$

Idea

For $p = \lceil r/d \rceil$, choose distinct $\alpha_0, \ldots, \alpha_{p-1}$, and let L operates on

$$\mathbb{V}_k \quad = \quad \mathbb{K}[x]_k e^{\alpha_0 x} \oplus \cdots \oplus \mathbb{K}[x]_k e^{\alpha_{p-1} x}$$

We replace one multiplication of big matrices by several multiplications of smaller matrices

- Evaluations of $\Phi^{3d,2d}_{L_{\ltimes \alpha_i}}$ and $\Phi^{4d,3d}_{K_{\ltimes \alpha_i}}$, for i from 0 to $p-1$ ($\tilde{\mathcal{O}}(dr)$)
- Matrix multiplications: For all i, $\Phi^{4d,2d}_{KL_{\ltimes \alpha_i}} = \Phi^{4d,3d}_{K_{\ltimes \alpha_i}} \cdot \Phi^{3d,2d}_{L_{\ltimes \alpha_i}}$.

Product using Multipoint Evaluations and Interpolation

We suppose $r > d$

Idea

For $p = \lceil r/d \rceil$, choose distinct $\alpha_0, \ldots, \alpha_{p-1}$, and let L operates on

$$\mathbb{V}_k \quad = \quad \mathbb{K}[x]_k e^{\alpha_0 x} \oplus \cdots \oplus \mathbb{K}[x]_k e^{\alpha_{p-1} x}$$

We replace one multiplication of big matrices by several multiplications of smaller matrices

- Evaluations of $\Phi^{3d,2d}_{L_{\ltimes \alpha_i}}$ and $\Phi^{4d,3d}_{K_{\ltimes \alpha_i}}$, for i from 0 to $p-1$ ($\tilde{\mathcal{O}}(dr)$)
- Matrix multiplications: For all i, $\Phi^{4d,2d}_{KL_{\ltimes \alpha_i}} = \Phi^{4d,3d}_{K_{\ltimes \alpha_i}} \cdot \Phi^{3d,2d}_{L_{\ltimes \alpha_i}}$. $\mathcal{O}(rd^{\omega-1})$
  r/d multiplications of matrices of size $d \times d$ (instead of $r \times d$)

Product using Multipoint Evaluations and Interpolation

We suppose $r > d$

Idea

For $p = \lceil r/d \rceil$, choose distinct $\alpha_0, \ldots, \alpha_{p-1}$, and let L operates on

$$\mathbb{V}_k \quad = \quad \mathbb{K}[x]_k e^{\alpha_0 x} \oplus \cdots \oplus \mathbb{K}[x]_k e^{\alpha_{p-1} x}$$

We replace one multiplication of big matrices by several multiplications of smaller matrices

- Evaluations of $\Phi^{3d,2d}_{L_{\ltimes \alpha_i}}$ and $\Phi^{4d,3d}_{K_{\ltimes \alpha_i}}$, for i from 0 to $p-1$ ($\tilde{\mathcal{O}}(dr)$)
- Matrix multiplications: For all i, $\Phi^{4d,2d}_{KL_{\ltimes \alpha_i}} = \Phi^{4d,3d}_{K_{\ltimes \alpha_i}} \cdot \Phi^{3d,2d}_{L_{\ltimes \alpha_i}}$. $\mathcal{O}(rd^{\omega-1})$
  r/d multiplications of matrices of size $d \times d$ (instead of $r \times d$)
- Interpolations. From $\Phi^{4d,2d}_{KL_{\ltimes \alpha_i}}$ to KL $\tilde{\mathcal{O}}(dr)$

Product using Multipoint Evaluations and Interpolation

We suppose $r > d$

Idea

For $p = \lceil r/d \rceil$, choose distinct $\alpha_0, \ldots, \alpha_{p-1}$, and let L operates on

$$\mathbb{V}_k = \mathbb{K}[x]_k e^{\alpha_0 x} \oplus \cdots \oplus \mathbb{K}[x]_k e^{\alpha_{p-1} x}$$

We replace one multiplication of big matrices by several multiplications of smaller matrices

- Evaluations of $\Phi^{3d,2d}_{L_{\ltimes \alpha_i}}$ and $\Phi^{4d,3d}_{K_{\ltimes \alpha_i}}$, for i from 0 to $p-1$ ($\tilde{\mathcal{O}}(dr)$)
- Matrix multiplications: For all i, $\Phi^{4d,2d}_{KL_{\ltimes \alpha_i}} = \Phi^{4d,3d}_{K_{\ltimes \alpha_i}} \cdot \Phi^{3d,2d}_{L_{\ltimes \alpha_i}}$. $\mathcal{O}(rd^{\omega-1})$
  r/d multiplications of matrices of size $d \times d$ (instead of $r \times d$)
- Interpolations. From $\Phi^{4d,2d}_{KL_{\ltimes \alpha_i}}$ to KL $\tilde{\mathcal{O}}(dr)$

Conplexity of algorithm when $r > d$: $\tilde{\mathcal{O}}(rd^{\omega-1})$ arithmetic operations

IV Reflexion for the Case when $d > r$

Computing the Reflexion

The reflexion φ is the morphism from $\mathbb{K}[x]\langle\partial\rangle$ to itself such that:

$$\varphi(\partial) = x, \qquad \varphi(x) = -\partial.$$

Computing the Reflexion

The reflexion φ is the morphism from $\mathbb{K}[x]\langle\partial\rangle$ to itself such that:

$$\varphi(\partial) = x, \qquad \varphi(x) = -\partial.$$

Given

$$L \ = \ \sum_{i,j} p_{i,j}\partial^j x^i, \text{ compute } q_{i,j} \text{ with } L \ = \ \sum_{i,j} q_{i,j} x^i \partial^j.$$

We have :

$$\varphi(L) \ = \ \sum_{i,j} (-1)^j p_{i,j} x^j \partial^i.$$

Computing the Reflexion

The reflexion φ is the morphism from $\mathbb{K}[x]\langle\partial\rangle$ to itself such that:

$$\varphi(\partial) = x, \qquad \varphi(x) = -\partial.$$

Given

$$L = \sum_{i,j} p_{i,j}\partial^j x^i, \text{ compute } q_{i,j} \text{ with } L = \sum_{i,j} q_{i,j} x^i \partial^j.$$

We have :

$$\varphi(L) = \sum_{i,j} (-1)^j p_{i,j} x^j \partial^i.$$

Theorem

Given $L \in \mathbb{K}[x,\partial]_{d,r}$, we may compute $\varphi(L)$ in time $\tilde{O}(\min(dr, rd))$.

Computing the Reflexion

The reflexion φ is the morphism from $\mathbb{K}[x]\langle\partial\rangle$ to itself such that:

$$\varphi(\partial) = x, \qquad \varphi(x) = -\partial.$$

Given

$$L = \sum_{i,j} p_{i,j}\partial^j x^i, \text{ compute } q_{i,j} \text{ with } L = \sum_{i,j} q_{i,j} x^i \partial^j.$$

We have :

$$\varphi(L) = \sum_{i,j} (-1)^j p_{i,j} x^j \partial^i.$$

Theorem

Given $L \in \mathbb{K}[x,\partial]_{d,r}$, we may compute $\varphi(L)$ in time $\tilde{O}(\min(dr, rd))$.

Proof :

- Show that

$$i! q_{i,j} = \sum_{k \geqslant 0} \binom{j+k}{k} (i+k)! p_{i+k,j+k}$$

- Reduce to the computation of $\tilde{O}(d+r)$ Taylor shifts of length $\min(d,r)$.

New Algorithm for the Case when $d > r$

Idea : if L is an operator of bidegree (d,r), then $\varphi(L)$ is an operator of bidegree (r,d).

New Algorithm for the Case when $d > r$

Idea : if L is an operator of bidegree (d,r), then $\varphi(L)$ is an operator of bidegree (r,d).

Application: New algorithm

- compute the canonical forms (x at left and ∂ at right) of $\varphi(L)$ and $\varphi(K)$,
  new algorithm in $\tilde{\mathcal{O}}(dr)$
- compute the product $M = \varphi(L)\varphi(K)$ of operators $\varphi(L)$ and $\varphi(K)$
  in $\mathcal{O}(r^{\omega-1}d)$ using the previous algorithm
- return the (canonical form of the) operator $KL = \varphi^{-1}(M)$
  new algorithm in $\tilde{\mathcal{O}}(dr)$

New Algorithm for the Case when $d > r$

Idea : if L is an operator of bidegree (d,r), then $\varphi(L)$ is an operator of bidegree (r,d).
Application: New algorithm

- compute the canonical forms (x at left and ∂ at right) of $\varphi(L)$ and $\varphi(K)$,

  new algorithm in $\tilde{\mathcal{O}}(dr)$
- compute the product $M = \varphi(L)\varphi(K)$ of operators $\varphi(L)$ and $\varphi(K)$

  in $\mathcal{O}(r^{\omega-1}d)$ using the previous algorithm
- return the (canonical form of the) operator $KL = \varphi^{-1}(M)$

  new algorithm in $\tilde{\mathcal{O}}(dr)$

Complexity of the product in $\tilde{\mathcal{O}}(r^{\omega-1}d)$ arithmetic operations when $d > r$

V Conclusion

Contribution: better algorithm for the product of differential operator:

- Previous: $\mathcal{O}((d+r)^\omega)$ arithmetic operations
- New algorithm: $\mathcal{O}(rd\min(r,d)^{\omega-2})$ arithmetic operations

Contribution: better algorithm for the product of differential operator:

- Previous: $\mathcal{O}((d+r)^\omega)$ arithmetic operations
- New algorithm: $\mathcal{O}(rd\min(r,d)^{\omega-2})$ arithmetic operations

The same algorithm works also for product of , θ operators, recurrence operators or q-difference operators.

Contribution: better algorithm for the product of differential operator:

- Previous: $\mathcal{O}((d+r)^\omega)$ arithmetic operations
- New algorithm: $\mathcal{O}(rd\min(r,d)^{\omega-2})$ arithmetic operations

The same algorithm works also for product of , θ operators, recurrence operators or q-difference operators.

Perspective: Use of this fast product to improve algorithms to compute:

- differential operator canceling Hadamard product of series
- differential operator canceling product of series
- differential operator obtained by substitution with an algebraic function