Quasi-Optimal Multiplication of Linear Differential Operators

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

[1]CNRS, INRIA, UPMC (France)

[2]INRIA (France)

[3]CNRS, École Polytechnique (France)

ACA 2012
June, 28th 2012

I Introduction

Product of Linear Differential Operators

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

$$\forall f \in \mathbb{K}[x], \qquad PQ \cdot f = P \cdot (Q \cdot f).$$

Product of Linear Differential Operators

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

$$\forall f \in \mathbb{K}[x], \qquad PQ \cdot f = P \cdot (Q \cdot f).$$

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 $\leqslant r$ with polynomial coefficients of degrees $\leqslant d$ (i.e bidegrees (d,r)):

- Naive algorithm: $\mathcal{O}(d^2 r^2 \min(d,r))$ 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 see next talk

Fast algorithms for LCLM or GCRD for operators of bidegrees (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 see next talk

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

Product of operators of bidegrees (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 (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,
to appear in *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 (d,r) in

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

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,
to appear in *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 (d,r) in

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

If $d = r^2$ the complexity becomes

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

Idea: Modify van der Hoeven algorithm using

- Instead of evaluations of operators on x^i, evaluations on $x^i\exp(\alpha x)$
- Use multipoint evaluations and interpolations
- Use fast algorithm for performing Hermite interpolation

1

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

II New Vision of van der Hoeven Algorithm

Skew Product: a Linear Algebra Problem

$$QP = \sum_{i=0}^{r} q_i(x)\partial^i P$$

Skew Product: a Linear Algebra Problem

$$QP = \sum_{i=0}^{r} q_i(x)\partial^i P$$

We define

$$\mathsf{S}_\ell(P) := \begin{pmatrix} P \\ \partial P \\ \vdots \\ \partial^{\ell-1} P \end{pmatrix},$$

then for Q of order r, we have,

$$\mathsf{S}_1(QP) = \mathsf{S}_1(Q)\,\mathsf{S}_{r+1}(P).$$

Skew Product: a Linear Algebra Problem

$$QP = \sum_{i=0}^{r} q_i(x)\partial^i P$$

We define

$$\mathsf{S}_\ell(P) := \begin{pmatrix} P \\ \partial P \\ \vdots \\ \partial^{\ell-1} P \end{pmatrix},$$

then for Q of order r, we have,

$$\mathsf{S}_1(QP) = \mathsf{S}_1(Q)\,\mathsf{S}_{r+1}(P).$$

More generally, for any $\ell \geqslant 0$, we have:

$$\mathsf{S}_\ell(QP) = \mathsf{S}_\ell(Q)\,\mathsf{S}_{r+\ell}(P).$$

Evaluations and Interpolation of Operator

$$\mathsf{S}_\ell(P) := \begin{pmatrix} P \\ \partial P \\ \vdots \\ \partial^{\ell-1} P \end{pmatrix} = \begin{pmatrix} p_0 & p_1 & \cdots & p_r & 0 & \cdots & 0 \\ p_0' & p_1' + p_0 & \cdots & p_r' + p_{r-1} & p_r & 0 & \vdots \\ \vdots & & \vdots & & & \ddots & 0 \\ p_0^{(\ell-1)} & & \cdots & & & \cdots & p_r \end{pmatrix}$$

Evaluations and Interpolation of Operator

$$\mathsf{S}_\ell(P) := \begin{pmatrix} P \\ \partial P \\ \vdots \\ \partial^{\ell-1} P \end{pmatrix} = \begin{pmatrix} p_0 & p_1 & \cdots & p_r & 0 & \cdots & 0 \\ p_0' & p_1' + p_0 & \cdots & p_r' + p_{r-1} & p_r & 0 & \vdots \\ \vdots & & \vdots & & & \ddots & 0 \\ p_0^{(\ell-1)} & & \cdots & & & \cdots & p_r \end{pmatrix}$$

If P is an operator of bidegree (d,r), we can compute P from $S_d(P)(0)$

Fast Computation of $S_\ell(P)(0)$

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

$$\begin{pmatrix} 1 & 0 & & & & 0 \\ 1 & 1 & 0 & & & 0 \\ 1 & 2 & 1 & 0 & & 0 \\ 1 & 3 & 3 & 1 & 0 & 0 \\ \vdots & & & & \ddots & \end{pmatrix} \begin{pmatrix} 0 & \cdots & 0 & p_0 & p_1 & \cdots & p_r \\ \vdots & 0 & p_0' & p_1' & \cdots & p_r' & 0 \\ 0 & \iddots & & & \iddots & 0 & \vdots \\ p_0^{(\ell-1)} & p_1^{(\ell-1)} & \dots & p_r^{(\ell-1)} & 0 & \vdots & 0 \end{pmatrix}$$

$$= \begin{pmatrix} & & & p_0 & p_1 & \cdots & p_r \\ & & p_0' & p_0 + p_1' & \cdots & p_r' + p_{r-1} & p_r \\ & \iddots & & & & & \vdots \\ p_0^{(\ell-1)} & & \dots & & & \dots & p_r \end{pmatrix}.$$

Fast Computation of $S_\ell(P)(0)$

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

$$\begin{pmatrix} 1 & 0 & & & & 0 \\ 1 & 1 & 0 & & & 0 \\ 1 & 2 & 1 & 0 & & 0 \\ 1 & 3 & 3 & 1 & 0 & 0 \\ \vdots & & & & \ddots & \end{pmatrix} \begin{pmatrix} 0 & \cdots & 0 & p_0 & p_1 & \cdots & p_r \\ \vdots & 0 & p_0' & p_1' & \cdots & p_r' & 0 \\ 0 & \iddots & & & \iddots & 0 & \vdots \\ p_0^{(\ell-1)} & p_1^{(\ell-1)} & \cdots & p_r^{(\ell-1)} & 0 & \vdots & 0 \end{pmatrix}$$

$$= \begin{pmatrix} & & & p_0 & p_1 & \cdots & p_r \\ & & p_0' & p_0 + p_1' & \cdots & p_r' + p_{r-1} & p_r \\ & \iddots & & & & & \vdots \\ p_0^{(\ell-1)} & & \cdots & & & \cdots & p_r \end{pmatrix}.$$

Applications:

- Computation of P from $S_d(P)(0)$ in $\mathcal{O}((r+d)^\omega)$ (in $\tilde{\mathcal{O}}(rd)$ using structured matrices)

Fast Computation of $S_\ell(P)(0)$

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

$$\begin{pmatrix} 1 & 0 & & & & 0 \\ 1 & 1 & 0 & & & 0 \\ 1 & 2 & 1 & 0 & & 0 \\ 1 & 3 & 3 & 1 & 0 & 0 \\ \vdots & & & & \ddots & \end{pmatrix} \begin{pmatrix} 0 & \cdots & 0 & p_0 & p_1 & \cdots & p_r \\ \vdots & 0 & p_0' & p_1' & \cdots & p_r' & 0 \\ 0 & \iddots & & & \iddots & 0 & \vdots \\ p_0^{(\ell-1)} & p_1^{(\ell-1)} & \cdots & p_r^{(\ell-1)} & 0 & \vdots & 0 \end{pmatrix}$$

$$= \begin{pmatrix} & & & p_0 & p_1 & \cdots & p_r \\ & & p_0' & p_0 + p_1' & \cdots & p_r' + p_{r-1} & p_r \\ & \iddots & & & & & \vdots \\ p_0^{(\ell-1)} & & \cdots & & & \cdots & p_r \end{pmatrix}.$$

Applications:

- Computation of P from $S_d(P)(0)$ in $\mathcal{O}((r+d)^\omega)$ (in $\tilde{\mathcal{O}}(rd)$ using structured matrices)
- Computation of $S_d(P)(0)$ from P in $\mathcal{O}((r+d)^\omega)$ (in $\tilde{\mathcal{O}}(rd)$ using structured matrices)

Complexity of van der Hoeven Algorithm

Easy bound: If P and Q are of bidegrees (r,d), QP is of bidegree $(2r,2d)$.

- Evaluation of $\mathsf{S}_{2d+r}(P)(0)$ and $\mathsf{S}_{2d}(Q)(0)$
- Matrix multiplication $\mathsf{S}_{2d}(QP)(0) = \mathsf{S}_{2d}(Q)(0) \cdot \mathsf{S}_{2d+r}(P)(0)$
- Interpolation. From $\mathsf{S}_{2d}(QP)(0)$ to the coefficients of QP

Complexity of van der Hoeven Algorithm

Easy bound: If P and Q are of bidegrees (r,d), QP is of bidegree $(2r,2d)$.

- Evaluation of $\mathsf{S}_{2d+r}(P)(0)$ and $\mathsf{S}_{2d}(Q)(0)$ $\tilde{\mathcal{O}}(r(r+d))$ ops
- Matrix multiplication $\mathsf{S}_{2d}(QP)(0) = \mathsf{S}_{2d}(Q)(0) \cdot \mathsf{S}_{2d+r}(P)(0)$ $\mathcal{O}((d+r)^{\omega})$ ops
- Interpolation. From $\mathsf{S}_{2d}(QP)(0)$ to the coefficients of QP $\tilde{\mathcal{O}}(dr)$ ops

Complexity of van der Hoeven Algorithm

Easy bound: If P and Q are of bidegrees (r,d), QP is of bidegree $(2r,2d)$.

- Evaluation of $\mathsf{S}_{2d+r}(P)(0)$ and $\mathsf{S}_{2d}(Q)(0)$ $\tilde{\mathcal{O}}(r(r+d))$ ops
- Matrix multiplication $\mathsf{S}_{2d}(QP)(0) = \mathsf{S}_{2d}(Q)(0) \cdot \mathsf{S}_{2d+r}(P)(0)$ $\mathcal{O}((d+r)^\omega)$ ops
- Interpolation. From $\mathsf{S}_{2d}(QP)(0)$ to the coefficients of QP $\tilde{\mathcal{O}}(dr)$ ops

Complexity of van der Hoeven Algorithm

Easy bound: If P and Q are of bidegrees (r,d), QP is of bidegree $(2r,2d)$.

- Evaluation of $\mathsf{S}_{2d+r}(P)(0)$ and $\mathsf{S}_{2d}(Q)(0)$ $\tilde{\mathcal{O}}(r(r+d))$ ops
- Matrix multiplication $\mathsf{S}_{2d}(QP)(0) = \mathsf{S}_{2d}(Q)(0) \cdot \mathsf{S}_{2d+r}(P)(0)$ $\mathcal{O}((d+r)^{\omega})$ ops
- Interpolation. From $\mathsf{S}_{2d}(QP)(0)$ to the coefficients of QP $\tilde{\mathcal{O}}(dr)$ ops

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

Complexity of van der Hoeven Algorithm

Easy bound: If P and Q are of bidegrees (r,d), QP is of bidegree $(2r,2d)$.

- Evaluation of $\mathsf{S}_{2d+r}(P)(0)$ and $\mathsf{S}_{2d}(Q)(0)$ $\tilde{\mathcal{O}}(r(r+d))$ ops
- Matrix multiplication $\mathsf{S}_{2d}(QP)(0) = \mathsf{S}_{2d}(Q)(0) \cdot \mathsf{S}_{2d+r}(P)(0)$ $\mathcal{O}((d+r)^{\omega})$ ops
- Interpolation. From $\mathsf{S}_{2d}(QP)(0)$ to the coefficients of QP $\tilde{\mathcal{O}}(dr)$ 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

Product using Multipoint Evaluations and Interpolation

New idea ($d > r$ and $d/r \in \mathbb{N}$):
We replace one multiplication of big matrices by several multiplications of smaller matrices

Product using Multipoint Evaluations and Interpolation

New idea ($d > r$ and $d/r \in \mathbb{N}$):
We replace one multiplication of big matrices by several multiplications of smaller matrices

- Evaluations of $\mathsf{S}_{2r}(P)(0), \mathsf{S}_{2r}(P)(1), \cdots, \mathsf{S}_{2r}(P)(2d/r-1)$ and $\mathsf{S}_r(Q)(0), \mathsf{S}_r(Q)(1), \cdots, \mathsf{S}_r(Q)(2d/r-1)$

Product using Multipoint Evaluations and Interpolation

New idea ($d > r$ and $d/r \in \mathbb{N}$):
We replace one multiplication of big matrices by several multiplications of smaller matrices

- Evaluations of $\mathsf{S}_{2r}(P)(0), \mathsf{S}_{2r}(P)(1), \cdots, \mathsf{S}_{2r}(P)(2d/r-1)$ and $\mathsf{S}_{r}(Q)(0), \mathsf{S}_{r}(Q)(1), \cdots, \mathsf{S}_{r}(Q)(2d/r-1)$
- Matrix multiplications: For all i, $\mathsf{S}_{r}(QP)(i) = \mathsf{S}_{r}(Q)(i) \cdot \mathsf{S}_{2r}(P)(i)$.

Product using Multipoint Evaluations and Interpolation

New idea ($d > r$ and $d/r \in \mathbb{N}$):
We replace one multiplication of big matrices by several multiplications of smaller matrices

- Evaluations of $\mathsf{S}_{2r}(P)(0), \mathsf{S}_{2r}(P)(1), \cdots, \mathsf{S}_{2r}(P)(2d/r-1)$ and $\mathsf{S}_r(Q)(0), \mathsf{S}_r(Q)(1), \cdots, \mathsf{S}_r(Q)(2d/r-1)$
- Matrix multiplications: For all i, $\mathsf{S}_r(QP)(i) = \mathsf{S}_r(Q)(i) \cdot \mathsf{S}_{2r}(P)(i)$.
  $\mathcal{O}(dr^{\omega-1})$
  d/r multiplications of matrices of size $r \times r$ (instead of $r \times d$)

Product using Multipoint Evaluations and Interpolation

New idea ($d > r$ and $d/r \in \mathbb{N}$):
We replace one multiplication of big matrices by several multiplications of smaller matrices

- Evaluations of $\mathsf{S}_{2r}(P)(0), \mathsf{S}_{2r}(P)(1), \cdots, \mathsf{S}_{2r}(P)(2d/r-1)$ and $\mathsf{S}_r(Q)(0), \mathsf{S}_r(Q)(1), \cdots, \mathsf{S}_r(Q)(2d/r-1)$
- Matrix multiplications: For all i, $\mathsf{S}_r(QP)(i) = \mathsf{S}_r(Q)(i) \cdot \; \mathsf{S}_{2r}(P)(i)$.
  $\mathcal{O}(dr^{\omega-1})$
  d/r multiplications of matrices of size $r \times r$ (instead of $r \times d$)
- Interpolations. From $\mathsf{S}_r(QP)(0), \mathsf{S}_r(QP)(1), \cdots, \mathsf{S}_r(QP)(2d/r-1)$ to the coefficients of QP

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:

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

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

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

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:

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

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 $\mathsf{S}_r(P)(i)$ for $i \in [0..2d/r-1]$ in $\tilde{\mathcal{O}}(rd)$ ops

New Algorithm when $d > r$

- Evaluations of $\mathsf{S}_r(P)(0), \mathsf{S}_r(P)(1), \cdots, \mathsf{S}_{2r}(P)(2d/r-1)$ and $\mathsf{S}_r(Q)(0), \mathsf{S}_r(Q)(1), \cdots, \mathsf{S}_r(Q)(2d/r-1)$
- Matrix multiplications: For all i $\mathsf{S}_r(QP)(i) = \mathsf{S}_r(Q)(i) \cdot \ \mathsf{S}_{2r}(P)(i)$ $\mathcal{O}(dr^{\omega-1})$
- Interpolation. From $\mathsf{S}_r(QP)(0), \mathsf{S}_r(QP)(1), \cdots, \mathsf{S}_r(QP)(2d/r-1)$ to the coefficients of QP

New Algorithm when $d > r$

- Evaluations of $\mathsf{S}_r(P)(0), \mathsf{S}_r(P)(1), \cdots, \mathsf{S}_{2r}(P)(2d/r-1)$ and $\mathsf{S}_r(Q)(0), \mathsf{S}_r(Q)(1), \cdots, \mathsf{S}_r(Q)(2d/r-1)$ $\tilde{\mathcal{O}}(dr)$
- Matrix multiplications: For all i $\mathsf{S}_r(QP)(i) = \mathsf{S}_r(Q)(i) \cdot \mathsf{S}_{2r}(P)(i)$ $\mathcal{O}(dr^{\omega-1})$
- Interpolation. From $\mathsf{S}_r(QP)(0), \mathsf{S}_r(QP)(1), \cdots, \mathsf{S}_r(QP)(2d/r-1)$ to the coefficients of QP $\tilde{\mathcal{O}}(dr)$

New Algorithm when $d > r$

- Evaluations of $\mathsf{S}_r(P)(0), \mathsf{S}_r(P)(1), \cdots, \mathsf{S}_{2r}(P)(2d/r-1)$ and $\mathsf{S}_r(Q)(0), \mathsf{S}_r(Q)(1), \cdots, \mathsf{S}_r(Q)(2d/r-1)$ $\tilde{\mathcal{O}}(dr)$
- Matrix multiplications: For all i $\mathsf{S}_r(QP)(i) = \mathsf{S}_r(Q)(i) \cdot \ \mathsf{S}_{2r}(P)(i)$ $\mathcal{O}(dr^{\omega-1})$
- Interpolation. From $\mathsf{S}_r(QP)(0), \mathsf{S}_r(QP)(1), \cdots, \mathsf{S}_r(QP)(2d/r-1)$ to the coefficients of QP $\tilde{\mathcal{O}}(dr)$

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

Application when $r > d$

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

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

Application when $r > d$

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

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

Application, new algorithm for the product when $r > d$

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

Application when $r > d$

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

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

Application, new algorithm for the product when $r > d$

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

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

IV 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 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 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