Product of Linear Differential Operators by Evaluation and Interpolation

Nicolas Le Roux
nicolas.le_roux@inria.fr

INRIA Rocquencourt, Algorithms Project

March 17, 2008

Joint work with Alin Bostan and Frédéric Chyzak.
Statement of the Problem

Notations

- $\partial = \frac{d}{dX}$, $\mathbb{K}$ a field.
- $\mathbb{K}[X]\langle \partial \rangle$ the ring of skew polynomials in $(X, \partial)$.

Canonical form of $L \in \mathbb{K}[X]\langle \partial \rangle$: $L = \sum_{i=0}^{r_L} \sum_{j=0}^{d_L} l_{i,j} X^j \partial^i$.

$(r_L, d_L)$ is the bidegree of $L$.

Problem

Given $B, A \in \mathbb{K}[X]\langle \partial \rangle$ in canonical form, compute the canonical form of $BA$. 

Nicolas Le Roux  nicolas.le_roux@inria.fr
Review of Known Algorithms

- Naive Algorithms
Review of Known Algorithms

- **Naive Algorithms**
  - Using naive expansion and the Leibniz Formula:

    \[ BA = \sum_{0 \leq i, j, k, l \leq n} b_{i,j} a_{k,l} X^j \left( \partial^i X^l \right) \partial^k; \]

    The canonical form of \( \partial^i X^l \) is obtained by Leibniz formula:

    \[ \partial^i X^l = \sum_{m=0}^{	ext{min}(i, l)} (l)_m \binom{i}{m} X^{l-m} \partial^{i-m}. \]

    This leads to \( \mathcal{O}(n^5) \) operations in \( \mathbb{K} \).
Review of Known Algorithms

- **Naive Algorithms**
  - Using naive expansion and the Leibniz Formula: $O(n^5)$;
  - Using an iterative scheme and polynomial FFT:
    Write $B$ as $B = \sum_{i=0}^{n} b_i(X) \partial^i$ and expand $BA$ as
    
    $$BA = \sum_{i=0}^{n} b_i(X) (\partial^i A).$$

    $\partial^i A$ is computed iteratively using the formula
    
    $$\partial L = \frac{dL}{dX} + L \partial.$$  

    This leads to $O(n^2 M(n))$ operations in $\mathbb{K}$. 
   
   $M(n) = \text{number of operations in } \mathbb{K} \text{ for computing the product of two polynomials with degree } n.$
Review of Known Algorithms

- **Naive Algorithms**
  - Using naive expansion and the Leibniz Formula: $O(n^5)$;
  - Using an iterative scheme and polynomial FFT: $\tilde{O}(n^3)$;
  - Using Takayama’s Formula and polynomial FFT:

$$BA = \sum_{k=0}^{n} \frac{1}{k!} \left( \frac{d^k B}{d\partial^k} \star \frac{d^k A}{dX^k} \right)$$

$M(n^2)$ ops in $\mathbb{K}$

※ means commutative products in bidegree $(n, n)$. This leads to a $O(nM(n^2))$ operations in $\mathbb{K}$.
Review of Known Algorithms

- **Naive Algorithms**
  - Using naive expansion and the Leibniz Formula: $\mathcal{O}(n^5)$;
  - Using an iterative scheme and polynomial FFT: $\tilde{\mathcal{O}}(n^3)$;
  - Using Takayama’s Formula and polynomial FFT: $\tilde{\mathcal{O}}(n^3)$;

- **van der Hoeven’s Algorithm** [2002]:
  It reduces the product to a constant number of product of $n \times n$ matrices with entries in $\mathbb{K}$. 

Review of Known Algorithms

- Naive Algorithms
  - Using naive expansion and the Leibniz Formula: $O(n^5)$;
  - Using an iterative scheme and polynomial FFT: $\tilde{O}(n^3)$;
  - Using Takayama’s Formula and polynomial FFT: $\tilde{O}(n^3)$;
- van der Hoeven’s Algorithm [2002]: $O(n^3)$ and even better.
Results

- Product of dense skew polynomials in \( \mathbb{K}[X]\langle \partial \rangle \) with bidegree \((n, n)\).
  - Equivalence with product of \( n \times n \) matrices with entries in \( \mathbb{K} \);
  - An Evaluation-Interpolation algorithm reducing the number of product of \( n \times n \) matrices;
  - An optimal algorithm for product when characteristic of \( \mathbb{K} \) is small.

- Further complexity estimate when the skew polynomials are sparse.

- Experiments. Early prototype in Magma more efficient than the existing Magma product.

Nicolas Le Roux  nicolas.le_roux@inria.fr
Results

Number of $n \times n$ Matrix Products in van der Hoeven’s algorithm and ours

<table>
<thead>
<tr>
<th>$(a, b, c)$</th>
<th>vdH</th>
<th>Improved-vdH</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product by blocks</td>
<td>96</td>
<td>48</td>
<td>12</td>
</tr>
<tr>
<td>Zero blocks + Strassen product</td>
<td>48</td>
<td>12</td>
<td>8</td>
</tr>
</tbody>
</table>

- Evaluation of $B$ and $A$ resulting in two evaluation matrices of sizes $(an) \times (bn)$ and $(bn) \times (cn)$.
- Computation of the product of the two matrices:

\[
\begin{align*}
  a \{ \times \ b \} & \leq c
\end{align*}
\]
Results

Number of $n \times n$ Matrix Products in van der Hoeven’s algorithm and ours

<table>
<thead>
<tr>
<th>$(a, b, c)$</th>
<th>vdH</th>
<th>Improved-vdH</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product by blocks</td>
<td>96</td>
<td>48</td>
<td>12</td>
</tr>
<tr>
<td>Zero blocks + Strassen product</td>
<td>48</td>
<td>12</td>
<td>8</td>
</tr>
</tbody>
</table>

- Evaluation of $B$ and $A$ resulting in two evaluation matrices of sizes $(an) \times (bn)$ and $(bn) \times (cn)$.
- Computation of the product of the two matrices: $(abc)$ products of $n \times n$ matrices.
- Interpolation of $C$ from the $(an) \times (cn)$ matrix.
1 Review of van der Hoeven Algorithm
   - Product in $\mathbb{K}[X]\langle\theta\rangle$
   - Product in $\mathbb{K}[X]\langle\partial\rangle$

2 Equivalence with Matrix Product
   - Some Improvements of van der Hoeven Algorithm
   - Proving the Equivalence

3 A new Evaluation-Interpolation Algorithm
   - Reducing the size of the Evaluation Matrix
   - The Evaluation and Interpolation Steps

4 Further Discussions
   - When Equivalence Fails?
   - Miscellaneous
Steps of van der Hoeven’s algorithm

Given \( B, A \) in \( \mathbb{K}[X] \langle \theta \rangle \).

- Convert them as recurrence operators: this is done by rewriting \( B \) and \( A \) as elements of \( \mathbb{K}[X, X^{-1}] \langle \theta \rangle \);
  - \( \theta = X \partial \) plays the role of multiplication by \( k \);
  - \( X \) plays the role of the shift operator.
Steps of van der Hoeven’s algorithm

$B, A$ in $\mathbb{K}[X]\langle\partial\rangle$.

- convert them as recurrence operators: this is done by rewriting $B$ and $A$ as elements of $\mathbb{K}[X, X^{-1}]\langle\theta\rangle$;
- compute the evaluation matrices of $B$ and $A$;
- compute the product of the two previous matrices;
- recover $C = BA$ from the last matrix: $C$ is written as an element in $\mathbb{K}[X, X^{-1}]\langle\theta\rangle$;
- rewrite $C$ as an element of $\mathbb{K}[X]\langle\partial\rangle$. 

Nicolas Le Roux  nicolas.le_roux@inria.fr
Action over $\mathbb{K}[X]$:

$\mathbb{K}[X]\langle \theta \rangle$ acts naturally (and faithfully) over $\mathbb{K}[X]$:
$L \in \mathbb{K}[X]\langle \theta \rangle$ defines a $\mathbb{K}$-linear map from $\mathbb{K}[X]$ to itself and is uniquely determined by this map. Therefore $L$ can be identified with an infinite matrix.

van der Hoeven’s observation:

Assume $L$ with bidegree $(n, n)$. $L$ is uniquely determined as a $\mathbb{K}$-linear map from $\mathbb{K}[X]_{\leq n}$ to $\mathbb{K}[X]_{\leq 2n}$.
van der Hoeven’s observation

Assume $L$ with bidegree $(n, n)$. $L$ is uniquely determined as a $\mathbb{K}$-linear map from $\mathbb{K}[X]_{\leq n}$ to $\mathbb{K}[X]_{\leq 2n}$.

Example

$L = (a + b X) + (c + d X) \theta$.
$L(1) = a + b X, \quad L(X) = (a + c) X + (b + d) X^2$

Associated matrix: $M^L = \begin{bmatrix} a + 0c & 0 \\ b + 0d & a + 1c \\ 0 & b + 1d \end{bmatrix}$
Action over $\mathbb{K}[X]$

van der Hoeven’s observation

Assume $L$ with bidegree $(n, n)$. $L$ is uniquely determined as a $\mathbb{K}$-linear map from $\mathbb{K}[X]_{\leq n}$ to $\mathbb{K}[X]_{\leq 2n}$.

Example

$L(1) = a + bX, \quad L(X) = (a + c)X + (b + d)X^2$

$L(X^2) = (a + 2c)X^2 + (b + 2d)X^3$

Associated matrix: $M^L_{4,3} = \begin{bmatrix}
  a + 0c & 0 & 0 \\
  b + 0d & a + 1c & 0 \\
  0 & b + 1d & a + 2c \\
  0 & 0 & b + 2d
\end{bmatrix}$
Scheme of the Algorithm

\[ \mathbb{K}[X] \leq 4n \quad \overset{B}{\longrightarrow} \quad \mathbb{K}[X] \leq 3n \quad \overset{A}{\longrightarrow} \quad \mathbb{K}[X] \leq 2n \]

\[ C = BA \]

\[ M^C = M^B_{4n,3n} \times M^A_{3n,2n} \]
Algorithm in Action when $n = 1$

Input:

\[ B = (e + f X) + (g + h X)\theta, \ A = (a + b X) + (c + d X)\theta \]

\[
\begin{bmatrix}
1 & 0 \\
1 & 1 \\
1 & 2
\end{bmatrix}
\begin{bmatrix}
a & b \\
c & d
\end{bmatrix}
\]
gives the multipoint-evaluation of polynomials.

Output:

\[ C = C_0(\theta) + XC_1(\theta) + X^2C_2(\theta) \]
Algorithm in Action when $n = 1$

Input:

\[ B = (e + f X) + (g + h X)\theta, \quad A = (a + b X) + (c + d X)\theta \]

We get

\[ M_{4,3}^A = \begin{bmatrix} a & 0 & 0 \\ b & a + c & 0 \\ 0 & b + d & a + 2c \\ 0 & 0 & b + 2d \end{bmatrix} \]

Output:

\[ C = C_0(\theta) + XC_1(\theta) + X^2C_2(\theta) \]
Algorithm in Action when $n = 1$

**Input:**

$$B = (e + fX) + (g + hX)\theta, \quad A = (a + bX) + (c + dX)\theta$$

$$\begin{bmatrix}
1 & 0 \\
1 & 1 \\
1 & 2 \\
1 & 3
\end{bmatrix}
\begin{bmatrix}
e & f \\
g & h
\end{bmatrix}$$

gives the multipoint-evaluation of polynomials.

**Output:**

$$C = C_0(\theta) + XC_1(\theta) + X^2C_2(\theta)$$
Algorithm in Action when $n = 1$

Input:

$$B = (e + f X) + (g + h X)\theta, \quad A = (a + b X) + (c + d X)\theta$$

We get

$$M_{5,4}^B = \begin{bmatrix}
    e & 0 & 0 & 0 & 0 \\
    f & e + g & 0 & 0 & 0 \\
    0 & f + h & e + 2g & 0 & 0 \\
    0 & 0 & f + 2h & e + 3g & 0 \\
    0 & 0 & 0 & f + 3h & 0
\end{bmatrix}$$

Output:

$$C = C_0(\theta) + XC_1(\theta) + X^2 C_2(\theta)$$
Algorithm in Action when \( n = 1 \)

**Input:**

\[
B = (e + f X) + (g + h X)\theta, \quad A = (a + b X) + (c + d X)\theta
\]

**Output:**

\[
C = C_0(\theta) + XC_1(\theta) + X^2C_2(\theta)
\]

Nicolas Le Roux  
nicolas.le_roux@inria.fr
Algorithm in Action when $n = 1$

Input:

$B = (e + fX) + (g + hX)\theta$, $A = (a + bX) + (c + dX)\theta$

$$M^C = \begin{bmatrix}
C_0(0) & 0 & 0 \\
C_1(0) & C_0(1) & 0 \\
C_2(0) & C_1(1) & C_0(2) \\
0 & C_2(1) & C_1(2) \\
0 & 0 & C_2(2)
\end{bmatrix}$$

Output:

$C = C_0(\theta) + XC_1(\theta) + X^2C_2(\theta)$
Algorithm in Action when $n = 1$

Input:

\[ B = (e + f X) + (g + h X)\theta, \quad A = (a + b X) + (c + d X)\theta \]

\[
\begin{bmatrix}
1 & 0 & 0 \\
1 & 1 & 1 \\
1 & 2 & 4
\end{bmatrix}^{-1}
\begin{bmatrix}
C_0(0) & C_1(0) & C_2(0) \\
C_0(1) & C_1(1) & C_2(1) \\
C_0(2) & C_1(2) & C_2(2)
\end{bmatrix}
\]

gives the coefficients of $C$

Output:

\[ C = C_0(\theta) + XC_1(\theta) + X^2C_2(\theta) \]
Complexity estimates

- Vandermonde and inverse: $O(n^2)$.
- Matrix product: $n^\omega$, $2 \leq \omega \leq 3$.
  - naive product: $\omega = 3$;
  - Strassen algorithm: $\omega < 2.8$;
  - Coppersmith & Winograd algorithm: $\omega < 2.376$.
- van der Hoeven’s algorithm: $37n^\omega + O(n^2)$.
  - Evaluation: $(2 + 3)n^\omega + O(n^2)$;
  - $M^C$: $24n^\omega$;
  - Interpolation: $8n^\omega + O(n^2)$. 

Nicolas Le Roux  nicolas.le_roux@inria.fr  
Product of Linear Differential Operators
Extension to $\mathbb{K}[X, X^{-1}]\langle \theta \rangle$

\[ B = \sum_{i=0}^{n} \sum_{j=-n}^{n} b_{i,j} X^j \theta^i : \text{idem for } A. \]

\[ C = BA \]

\[ M^C = 6n \}

\[ \times 4n \}

\[ \mathbb{K}[X]_{\leq 4n} \}

\[ \mathbb{K}[X^{-1}]_{\leq 2n} \}

\[ \mathbb{K}[X]_{\leq 3n} \}

\[ \mathbb{K}[X^{-1}]_{\leq n} \}

\[ \mathbb{K}[X]_{\leq 2n} \]
Conversion between $\mathbb{K}[X]\langle \partial \rangle$ and $\mathbb{K}[X, X^{-1}]\langle \theta \rangle$

$L \in \mathbb{K}[X]\langle \partial \rangle$: 

$L = \sum_{i=0}^{n} \sum_{j=0}^{n} l_{i,j} X^j \partial^i \longleftrightarrow L = \sum_{i=0}^{n} \sum_{j=-n}^{n} \tilde{l}_{i,j} X^j \theta^i.$

- Write $L = \sum_{i=0}^{n} \sum_{j=0}^{n} l_{i,j} X^{j-i} (X^i \partial^i);$  

- Apply the change of bases:

$$
\begin{bmatrix}
1 \\
X \partial \\
\vdots \\
X^n \partial^n
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & \ldots & 0 \\
\ast & 1 & \ldots & \vdots \\
\vdots & \vdots & \ddots & 1 \\
\ast & \ldots & \ast & 1
\end{bmatrix}
\begin{bmatrix}
1 \\
\theta \\
\vdots \\
\theta^n
\end{bmatrix}
$$

\text{Construction in $\mathcal{O}(n^2)$ ops}

- Return $L = \sum_{i=0}^{n} \sum_{j=-n}^{n} \tilde{l}_{i,j} X^j \theta^i.$
1. **Review of van der Hoeven Algorithm**
   - Product in $\mathbb{K}[X]⟨\theta⟩$
   - Product in $\mathbb{K}[X]⟨\partial⟩$

2. **Equivalence with Matrix Product**
   - Some Improvements of van der Hoeven Algorithm
   - Proving the Equivalence

3. **A new Evaluation-Interpolation Algorithm**
   - Reducing the size of the Evaluation Matrix
   - The Evaluation and Interpolation Steps

4. **Further Discussions**
   - When Equivalence Fails?
   - Miscellaneous
Statement of the Result

Theorem (Bostan, Chyzak & Le Roux 2008)

Computing the product $C = BA$ where $B, A \in \mathbb{K}[X][\theta]$ (resp. $\mathbb{K}[X][\partial]$) with bidegree $(n, n)$ is equivalent to computing the product $MN$ where $M, N$ are $n \times n$ matrices with entries in $\mathbb{K}$.

Already proved. $C = BA$ reduces to a constant number of products $NM$ plus $O(n^2)$ other operations in $\mathbb{K}$ [van der Hoeven 2002]. To prove the converse. Interpolation, Evaluation, Conversion tasks can be done efficiently: $\tilde{O}(n^2)$. 
Improving Evaluation, Interpolation and conversion Steps

Evaluation and interpolation of $P = P_0 + \ldots + P_d X^d \in \mathbb{K}[X]$ at $(d+1)$ distincts points of $\mathbb{K}$: $\mathcal{O}(M(d) \log(d))$ operations in $\mathbb{K}$. It reduces to $\tilde{\mathcal{O}}(d)$ using fast multiplication.

- Evaluation matrices of $B$ and $A$: $\mathcal{O}(nM(n) \log(n))$ ops in $\mathbb{K}$.
- Computation of $C$ from $M^C$: $\mathcal{O}(nM(n) \log(n))$ ops in $\mathbb{K}$.

**Theorem (Bostan & Schost 2005)**

Conversions $L = \sum_{i=0}^{n} \sum_{j=0}^{n} l_{i,j} X^j \partial^i \leftrightarrow L = \sum_{i=0}^{n} \sum_{j=-n}^{n} \tilde{l}_{i,j} X^j \theta^i$ can be done using $2nM(n) \log(n) + \mathcal{O}(nM(n))$ operations in $\mathbb{K}$. 

Nicolas Le Roux  nicolas.le_roux@inria.fr

Product of Linear Differential Operators
Sketch of the proof

- Product of $(n+1) \times (n+1)$ matrices: constant number of products of lower triangular matrices;
- Assume $M, N$ are $(n+1) \times (n+1)$ lower triangular matrices. We construct from $M$ (resp. $N$) $B$ (resp. $A$) in $\mathbb{K}[X] \langle \theta \rangle$ with bidegree $(n, n)$: $\tilde{O}(n^2)$ ops in $\mathbb{K}$;

$$\tilde{M} = \begin{bmatrix} 0 & M & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & 0 \end{bmatrix} \quad \text{Interpolation} \quad B$$
Sketch of the proof

- Product of \((n+1) \times (n+1)\) matrices: constant number of products of lower triangular matrices;
- Assume \(M, N\) are \((n+1) \times (n+1)\) lower triangular matrices. We construct from \(M\) (resp. \(N\)) \(B\) (resp. \(A\)) in \(\mathbb{K}[X]\langle\theta\rangle\) with bidegree \((n, n)\): \(\tilde{O}(n^2)\) ops in \(\mathbb{K}\);
- Compute the product \(C = BA\) (by one of existing algorithms);
- Compute \(M^C\): \(\tilde{O}(n^2)\) ops in \(\mathbb{K}\).

Claim. \(MN\) is the upper left \((n+1) \times (n+1)\) submatrix of \(M^C\). Consequently: \(MN\) is computed by one product \(C = BA\) in \(\mathbb{K}[X]\langle\theta\rangle\) plus \(\tilde{O}(n^2)\) other operations in \(\mathbb{K}\).
Sketch of the proof

Proof of the claim

\[ M^C = M^B_{4n,3n} \times M^A_{3n,2n} \]
1. Review of van der Hoeven Algorithm
   - Product in $\mathbb{K}[X]\langle \theta \rangle$
   - Product in $\mathbb{K}[X]\langle \partial \rangle$

2. Equivalence with Matrix Product
   - Some Improvements of van der Hoeven Algorithm
   - Proving the Equivalence

3. A new Evaluation-Interpolation Algorithm
   - Reducing the size of the Evaluation Matrix
   - The Evaluation and Interpolation Steps

4. Further Discussions
   - When Equivalence Fails?
   - Miscellaneous
An Interpolation Result

Let \( L \in \mathbb{K}[X]\langle \partial \rangle \) with bidegree \((n, n)\).
Define a \( \mathbb{K} \)-linear endomorphism of \( \mathbb{K}[X]_{\leq n} \) by
\[
P \mapsto L(P) \mod X^{n+1}.
\]
Denote by \( M^{L} \) the corresponding matrix.

**Theorem (Bostan, Chyzak & Le Roux 2008)**

\[
L = \sum_{i=0}^{n} \sum_{j=0}^{n} l_{i,j} X^j \partial^i \text{ is uniquely determined by } M^{L}.
\]
Sketch of the Proof

Define \( C^L = \begin{bmatrix}
    l_{0,0} & \ldots & \ldots & \ldots & l_{n,0} \\
    \vdots & \ddots & \ddots & \ddots & \vdots \\
    l_{0,\ell} & \ldots & \ldots & \ldots & l_{n,0} \\
    \vdots & \ddots & \ddots & \ddots & \vdots \\
    l_{n,0} & \ldots & l_{n-\ell,n} & \ldots & l_{n,n}
\end{bmatrix} \).

We associate to each diagonal a polynomial the coefficients of which in the falling factorials are the entries of the diagonal: for instance when \( 0 \leq \ell \leq n \),

\[
\mathcal{L}_\ell(T) = \sum_{j=0}^{n-\ell} l_{j,j+\ell} T(T - 1) \ldots (T - j + 1).
\]
Why these polynomials $L_\ell$?
They appear from the following observation ($0 \leq \ell \leq n$):

$$X^i \partial^i (X^k) = \frac{k!}{(k-i)!} X^k = \theta (\theta - 1) \ldots (\theta - i + 1)(X^k).$$

Thus $X^i \partial^i = \theta (\theta - 1) \ldots (\theta - i + 1)$.
It follows that $L$ can be rewritten as:

$$L = \sum_{\ell=0}^{n} X^\ell L_\ell(\theta) + \sum_{\ell=-n}^{-1} L_\ell(\theta) \partial^\ell.$$
Sketch of the Proof

Ended. Whew!

\[ C^L = \begin{bmatrix} l_{0,\ell} & \ldots & \end{bmatrix} \]
\[ M^L = \begin{bmatrix} \mathcal{L}_\ell(0) & \ldots \end{bmatrix} \]
Scheme of the Algorithm

\[
B \quad \text{Evaluation mod } X^{2n+1} \quad M^B_{2n+1,3n+1} \quad \times \quad M^A_{3n+1,2n+1} \quad = \quad M^C_{2n+1,2n+1}
\]

\[
A \quad \text{Evaluation} \quad 0 \quad \times \quad 0 \quad = \quad 0
\]

\[
BA \quad \text{Interpolation}
\]
Fast Computation and Interpolation of the Evaluation Matrix

\[ L \in \mathbb{K}[X]\langle \partial \rangle \text{ with bidegree } (n, n). \]

**Computation of \( M^L \).** Its entries are computed diagonal by diagonal. Assume for instance \( 0 \leq \ell \leq n \).

We define \( \tilde{L}_\ell(T) = \sum_{i=0}^{n-\ell} \lambda_{i,i+\ell} T^i \).

**Claim.** The entries of the \( \ell \)th subdiagonal of \( M^L \) are the coefficients of the truncated series at order \( n - \ell + 1 \) of:

\[ (\tilde{L}_\ell(T) \exp(T)) \odot \left( \sum_{k \geq 0} k! T^k \right). \]
Theorem (Bostan, Chyzak & Le Roux 2008)

The product $BA$ where $B, A \in \mathbb{K}[X][\partial]$ with bidegree $(n, n)$ can be done in $8 \, n^\omega + O(nM(n))$ operations in $\mathbb{K}$.

- Evaluation matrices of $B$ and $A$: $O(nM(n))$;
- Computation of $M^C$: $8 \, n^\omega$;
- Recover $C$ from $M^C$: $O(nM(n))$. 
Computation of $M^C$ in $8n^\omega$ Operations in $\mathbb{K}$

\[ M^C = M^B_{2n+1,3n+1} \]

\[ M^A_{3n+1,2n+1} = 0 \]

\[ M^C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \]
Computation of $M^C$ in $8n^\omega$ Operations in $K$

\[
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix} \times
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}
\]
\[
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix} \times
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}
\]
\[
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix} \times
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}
\]
\[
\text{cost: } 7n^\omega
\]
\[
\text{cost: } n^\omega
\]
## Experimental Issues

<table>
<thead>
<tr>
<th>( n )</th>
<th>( S^2 )</th>
<th>( vdB )</th>
<th>( \text{Iter} )</th>
<th>( \text{Tak} )</th>
<th>( \text{Sq} )</th>
<th>( \text{Rect} )</th>
<th>( \text{Interp} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>0.50</td>
<td>1.08</td>
<td>2.25</td>
<td>5.61</td>
<td>0.01</td>
<td>0.08</td>
<td>1.10</td>
</tr>
<tr>
<td>160</td>
<td>2.24</td>
<td>4.52</td>
<td>19.07</td>
<td>67.73</td>
<td>0.07</td>
<td>0.35</td>
<td>9.22</td>
</tr>
<tr>
<td>320</td>
<td>12.2</td>
<td>24.4</td>
<td>187</td>
<td>926</td>
<td>0.27</td>
<td>1.63</td>
<td>75.6</td>
</tr>
<tr>
<td>640</td>
<td>88.1</td>
<td>172</td>
<td>2604</td>
<td>( \infty )</td>
<td>1.26</td>
<td>9.40</td>
<td>770</td>
</tr>
<tr>
<td>1280</td>
<td>1961</td>
<td>( \infty )</td>
<td>( \infty )</td>
<td>( \infty )</td>
<td>7.24</td>
<td>59.1</td>
<td>( \infty )</td>
</tr>
<tr>
<td>80</td>
<td>9.93</td>
<td>28.4</td>
<td>6.99</td>
<td>24.3</td>
<td>0.01</td>
<td>0.07</td>
<td>0.93</td>
</tr>
<tr>
<td>160</td>
<td>128</td>
<td>498</td>
<td>118</td>
<td>725</td>
<td>0.05</td>
<td>0.27</td>
<td>6.89</td>
</tr>
<tr>
<td>320</td>
<td>2164</td>
<td>( \infty )</td>
<td>2492</td>
<td>( \infty )</td>
<td>0.24</td>
<td>4.37</td>
<td>51.4</td>
</tr>
</tbody>
</table>

Computation modulo 4294967291 and over \( \mathbb{Q} \).
Review of van der Hoeven Algorithm
- Product in $\mathbb{K}[X]\langle \theta \rangle$
- Product in $\mathbb{K}[X]\langle \partial \rangle$

Equivalence with Matrix Product
- Some Improvements of van der Hoeven Algorithm
- Proving the Equivalence

A new Evaluation-Interpolation Algorithm
- Reducing the size of the Evaluation Matrix
- The Evaluation and Interpolation Steps

Further Discussions
- When Equivalence Fails?
- Miscellaneous
Small Positive Characteristic $p$

**Important.** $\theta X^p = X^p \theta$.

**Theorem (Bostan, Chyzak & Le Roux 2008)**

Let $B, A \in \mathbb{K}[X] \langle \theta \rangle$ with bidegree $(n, n)$. Then $BA$ can be computed using $\tilde{O}(pn^2)$ operations in $\mathbb{K}$.

- Write $A$ and $B$ under the form $A = \sum_{v=0}^{p-1} A_v(X^p, \theta)X^v$ and $B = \sum_{u=0}^{p-1} X^u B_u(X^p, \theta)$. $O(nM(n) \log(n))$ ops in $\mathbb{K}$;
- Compute the commutative bivariate products with bidegree $(n/p, n)$. $C_{u,v} = A_v B_u$, for $0 \leq u, v < p$. $p^2M(n^2/p)$ ops in $\mathbb{K}$;
- Return $\sum_{u,v=0}^{p-1} X^u C_{u,v}(X^p, \theta)X^v$ in canonical form.
Unbalanced Bidegree Case

\( B, A \in K[X, \partial] \) with bidegree \((d, r)\).

**Case when \( d << r \).** For instance \( r = d^2 \).
Improved vdH and ours: constant number of \( d^2 \times d^2 \) by \( d^2 \times n \) matrix products. \( O(d^{\omega+2}) \) ops in \( K \).
Iterative scheme (slightly modified): \( \tilde{O}(d^4) \) ops in \( K \).

**Case when \( d >> r \).** For instance \( d = r^2 \).

- Improved vdH: constant number of \( r^2 \times r^2 \) by \( r^2 \times r^2 \) matrix products. \( O(r^{2\omega}) \) ops in \( K \).
- Ours: constant number of \( r \times r^2 \) by \( r^2 \times r^2 \) matrix products. \( O(r^{\omega+2}) \) ops in \( K \).
Iterative scheme: \( \tilde{O}(r^4) \) ops in \( K \).
Product of Sparse Linear Differential Operators

The discussion derives from the following case.

Special case

\[ \partial^i A \] where \( 0 \leq i \leq n \), \( A \) dense with bidegree \((n, n)\)

- Compute \( M^A_{n+1,2n+1} \);
  \( \mathcal{O}(nM(n)) \) operations in \( \mathbb{K} \).
- Compute the \( i \)th derivative of each column of \( M^A_{n+1,2n+1} \);
  \( \mathcal{O}(n^2) \) operations in \( \mathbb{K} \). It gives \( M^{\partial^i A} \).
- Recover \( \partial^i A \) from \( M^{\partial^i A} \);
  \( \mathcal{O}(nM(n)) \)