Lazy multiplication of power series

By Joris van der Hoeven

Lab. : LIX, École polytechnique, France
Email: vdhoeven@lix.polytechnique.fr
Web : http://lix.polytechnique.fr:80/~vdhoeven/

ISSAC ’97, 21-7-1997, Hawaii
Definitions

\( \mathcal{C} \): effective field of constants
\[
f = f_0 + f_1 z + f_2 z^2 + \cdots \in \mathcal{C}[z]
\]
\[
g = g_0 + g_1 z + g_2 z^2 + \cdots \in \mathcal{C}[z]
\]
\[
h = fg
\]

Static multiplication algorithms
Given \( f_0, \cdots, f_n \) and \( g_0, \cdots, g_n \), we compute \( h_0, \cdots, h_{n-1} \).
Time complexity: \( M(n) = O(n \log n) \).
Space complexity: \( O(n) \).

Lazy multiplication algorithms
\( h_i \) is output as soon as \( f_0, \cdots, f_i \) and \( g_0, \cdots, g_i \) are known, where \( i \) goes from 0 to \( n \).
Time complexity: \( L(n) = O(M(n) \log n) \).
Space complexity: \( O(n) \).
Applications

Functional equations
Lazy multiplication algorithms allow the coefficients of $f$ and $g$ to depend on the result $h$; i.e. $f_n$ and $g_n$ depend on $f_0, \cdots, f_{n-1}, g_0, \cdots, g_{n-1}$ and $h_0, \cdots, h_{n-1}$.

Example: exponentiation
If $\varphi = \varphi_1 z + \varphi_2 z^2 + \cdots$, then $\psi = \exp \varphi$ satisfies

$$\psi' = \varphi' \psi \quad (\varphi_0 = 1).$$

Taking $f = \varphi', g = \psi$ and $h = \varphi' \psi$, we get

$$\psi = \int h.$$

Here $g_n = \varphi_n = \frac{1}{n} h_{n-1}$ indeed only depends on $h_0, \cdots, h_{n-1}$. 
Lazy multiplication

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_7z^7$</td>
<td>$f_0g_7z^7$</td>
<td>$f_1g_7z^8$</td>
<td>$f_2g_7z^9$</td>
<td>$f_3g_7z^{10}$</td>
<td>$f_4g_7z^{11}$</td>
<td>$f_5g_7z^{12}$</td>
<td>$f_6g_7z^{13}$</td>
<td>$f_7g_7z^{14}$</td>
<td>...</td>
</tr>
<tr>
<td>$g_6z^6$</td>
<td>$f_0g_6z^6$</td>
<td>$f_1g_6z^7$</td>
<td>$f_2g_6z^8$</td>
<td>$f_3g_6z^9$</td>
<td>$f_4g_6z^{10}$</td>
<td>$f_5g_6z^{11}$</td>
<td>$f_6g_6z^{12}$</td>
<td>$f_7g_6z^{13}$</td>
<td>...</td>
</tr>
<tr>
<td>$g_5z^5$</td>
<td>$f_0g_5z^5$</td>
<td>$f_1g_5z^6$</td>
<td>$f_2g_5z^7$</td>
<td>$f_3g_5z^8$</td>
<td>$f_4g_5z^9$</td>
<td>$f_5g_5z^{10}$</td>
<td>$f_6g_5z^{11}$</td>
<td>$f_7g_5z^{12}$</td>
<td>...</td>
</tr>
<tr>
<td>$g_4z^4$</td>
<td>$f_0g_4z^4$</td>
<td>$f_1g_4z^5$</td>
<td>$f_2g_4z^6$</td>
<td>$f_3g_4z^7$</td>
<td>$f_4g_4z^8$</td>
<td>$f_5g_4z^9$</td>
<td>$f_6g_4z^{10}$</td>
<td>$f_7g_4z^{11}$</td>
<td>...</td>
</tr>
<tr>
<td>$g_3z^3$</td>
<td>$f_0g_3z^3$</td>
<td>$f_1g_3z^4$</td>
<td>$f_2g_3z^5$</td>
<td>$f_3g_3z^6$</td>
<td>$f_4g_3z^7$</td>
<td>$f_5g_3z^8$</td>
<td>$f_6g_3z^9$</td>
<td>$f_7g_3z^{10}$</td>
<td>...</td>
</tr>
<tr>
<td>$g_2z^2$</td>
<td>$f_0g_2z^2$</td>
<td>$f_1g_2z^3$</td>
<td>$f_2g_2z^4$</td>
<td>$f_3g_2z^5$</td>
<td>$f_4g_2z^6$</td>
<td>$f_5g_2z^7$</td>
<td>$f_6g_2z^8$</td>
<td>$f_7g_2z^9$</td>
<td>...</td>
</tr>
<tr>
<td>$g_1z$</td>
<td>$f_0g_1z$</td>
<td>$f_1g_1z^2$</td>
<td>$f_2g_1z^3$</td>
<td>$f_3g_1z^4$</td>
<td>$f_4g_1z^5$</td>
<td>$f_5g_1z^6$</td>
<td>$f_6g_1z^7$</td>
<td>$f_7g_1z^8$</td>
<td>...</td>
</tr>
<tr>
<td>$g_0$</td>
<td>$f_0g_0$</td>
<td>$f_1g_0z$</td>
<td>$f_2g_0z^2$</td>
<td>$f_3g_0z^3$</td>
<td>$f_4g_0z^4$</td>
<td>$f_5g_0z^5$</td>
<td>$f_6g_0z^6$</td>
<td>$f_7g_0z^7$</td>
<td>...</td>
</tr>
</tbody>
</table>

$\times f_0 + f_1z + f_2z^2 + f_3z^3 + f_4z^4 + f_5z^5 + f_6z^6 + f_7z^7 + \cdots$
More applications

Algebraic differential equations
Compute $f_n$, where $f$ solution of

$$\sum_{i_0, \ldots, i_r} P_{i_0, \ldots, i_r} f^{i_0} \cdots (f^{(r)})^{i_r} = 0,$$

with suitable initial conditions.
Our result $\Rightarrow$ solution in time $O(M(n) \log n)$.
Extension to systems of algebraic differential equations.

Brent and Kung: a statical $O(M(n))$ algorithm.
Time and space complexities depend badly on $r$.
Harder to implement the general case.

Difference equations

$$s(z) = 1 + z \frac{s(z)^3 + 2s(z^3)}{3},$$

$s_n$ can be computed in time $O(M(n) \log n)$.
Combinatorial interpretation: $s_n$ is the number of stereoisomers of alcohols of the form $C_n H_{2n+1} OH$. 
### Partial differential equations

\[
\frac{\partial f}{\partial x} = \frac{\partial^2 f}{\partial y^2} + e^x f^2,
\]

with \( f(0, y) = \sin y \). We have

\[
f = f_{0.0} + f_{1.0} x + f_{0.1} y + f_{2.0} x^2 + f_{1.1} xy + \cdots
\]

The coefficients \( f_{i,j} \) with \( 0 \leq i, j \leq n \) can be computed in time \( O(M(n)^2 \log n) \) (even in time \( O(M(n^2) \log n) \)).

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{0.9} )</td>
<td>( f_{1.9} )</td>
<td>( f_{2.9} )</td>
<td>( f_{3.9} )</td>
<td>( f_{4.9} )</td>
<td>( f_{5.9} )</td>
<td>( f_{6.9} )</td>
<td>( f_{7.9} )</td>
<td>( f_{8.9} )</td>
<td>( \cdots )</td>
</tr>
<tr>
<td>( f_{0.8} )</td>
<td>( f_{1.8} )</td>
<td>( f_{2.8} )</td>
<td>( f_{3.8} )</td>
<td>( f_{4.8} )</td>
<td>( f_{5.8} )</td>
<td>( f_{6.8} )</td>
<td>( f_{7.8} )</td>
<td>( f_{8.8} )</td>
<td>( \cdots )</td>
</tr>
<tr>
<td>( f_{0.7} )</td>
<td>( f_{1.7} )</td>
<td>( f_{2.7} )</td>
<td>( f_{3.7} )</td>
<td>( f_{4.7} )</td>
<td>( f_{5.7} )</td>
<td>( f_{6.7} )</td>
<td>( f_{7.7} )</td>
<td>( f_{8.7} )</td>
<td>( \cdots )</td>
</tr>
<tr>
<td>( f_{0.6} )</td>
<td>( f_{1.6} )</td>
<td>( f_{2.6} )</td>
<td>( f_{3.6} )</td>
<td>( f_{4.6} )</td>
<td>( f_{5.6} )</td>
<td>( f_{6.6} )</td>
<td>( f_{7.6} )</td>
<td>( f_{8.6} )</td>
<td>( \cdots )</td>
</tr>
<tr>
<td>( f_{0.5} )</td>
<td>( f_{1.5} )</td>
<td>( f_{2.5} )</td>
<td>( f_{3.5} )</td>
<td>( f_{4.5} )</td>
<td>( f_{5.5} )</td>
<td>( f_{6.5} )</td>
<td>( f_{7.5} )</td>
<td>( f_{8.5} )</td>
<td>( \cdots )</td>
</tr>
<tr>
<td>( f_{0.4} )</td>
<td>( f_{1.4} )</td>
<td>( f_{2.4} )</td>
<td>( f_{3.4} )</td>
<td>( f_{4.4} )</td>
<td>( f_{5.4} )</td>
<td>( f_{6.4} )</td>
<td>( f_{7.4} )</td>
<td>( f_{8.4} )</td>
<td>( \cdots )</td>
</tr>
<tr>
<td>( f_{0.3} )</td>
<td>( f_{1.3} )</td>
<td>( f_{2.3} )</td>
<td>( f_{3.3} )</td>
<td>( f_{4.3} )</td>
<td>( f_{5.3} )</td>
<td>( f_{6.3} )</td>
<td>( f_{7.3} )</td>
<td>( f_{8.3} )</td>
<td>( \cdots )</td>
</tr>
<tr>
<td>( f_{0.2} )</td>
<td>( f_{1.2} )</td>
<td>( f_{2.2} )</td>
<td>( f_{3.2} )</td>
<td>( f_{4.2} )</td>
<td>( f_{5.2} )</td>
<td>( f_{6.2} )</td>
<td>( f_{7.2} )</td>
<td>( f_{8.2} )</td>
<td>( \cdots )</td>
</tr>
<tr>
<td>( f_{0.1} )</td>
<td>( f_{1.1} )</td>
<td>( f_{2.1} )</td>
<td>( f_{3.1} )</td>
<td>( f_{4.1} )</td>
<td>( f_{5.1} )</td>
<td>( f_{6.1} )</td>
<td>( f_{7.1} )</td>
<td>( f_{8.1} )</td>
<td>( \cdots )</td>
</tr>
<tr>
<td>( f_{0.0} )</td>
<td>( f_{1.0} )</td>
<td>( f_{2.0} )</td>
<td>( f_{3.0} )</td>
<td>( f_{4.0} )</td>
<td>( f_{5.0} )</td>
<td>( f_{6.0} )</td>
<td>( f_{7.0} )</td>
<td>( f_{8.0} )</td>
<td>( \cdots )</td>
</tr>
</tbody>
</table>
Related results

Functional composition and reversion
Brent and Kung:

– Static $O(M(n)\sqrt{n \log n})$ composition and reversion algorithms in characteristic zero.

– $O(M(n))$ algorithm for static left composition with differential algebraic function.

van der Hoeven:

– Static $O(M(n) \log n)$ right composition with algebraic power series.

– Lazy $O(L(n) \log n)$ right composition with algebraic power series.

– Lazy $O(L(n) \sqrt{n \log n})$ composition and reversion algorithms in characteristic zero.
Premature computations

If the first $2^{p+1}$ coefficients of $f$ and $g$ are known, then the multiplication

$$\Pi_{2^p,2^p} = (f_{2^p} z^{2^p} + \cdots + f_{2^p+1-1} z^{2^{p+1}-1})$$

$$= (g_{2^p} z^{2^p} + \cdots + g_{2^p+1-1} z^{2^{p+1}-1})$$

can be performed prematurely.

If the first $n = (k + 1)2^p$ coefficients of $f$ and $g$ are known, with $k \in \{2, 3, \cdots\}$ and $p \geq 1$, then the multiplications

$$\Pi_{2^p,k2^p} = (f_{2^p} z^{2^p} + \cdots + f_{2^p+1-1} z^{2^{p+1}-1})$$

$$= (g_{k2^p} z^{k2^p} + \cdots + g_{(k+1)2^p-1} z^{(k+1)2^p-1})$$

and

$$\Pi_{k2^p,2^p} = (f_{k2^p} z^{k2^p} + \cdots + f_{(k+1)2^p-1} z^{(k+1)2^p-1})$$

$$= (g_{2^p} z^{2^p} + \cdots + g_{2^p+1-1} z^{2^{p+1}-1})$$

can be performed prematurely.
Algorithm C. Input $n \in \mathbb{N}$. Output $h_n$.

A: extendable array which contains $h_0, h_1, \cdots$ whose entries are initialized by 0. We assume that $h_0, \cdots, h_{n-1}$ have been computed.

C1. [Border]
If $n = 0$, then set $A[0] := f_0g_0$.
Otherwise, set $A[n] := A[n] + f_0g_n + f_ng_0$.

C2. [Diagonal]
If $n = 2^{p+1}$ for some $p \geq 0$, then compute $\Pi_{2p,2p}$ and set $A[i] := A[i] + \Pi_{2p,2p,i}$ for all $2^{p+1} \leq i \leq 2^{p+2} - 2$.

C3. [Main]
For each $k \geq 2$ and $p \geq 0$ such that $n = (k + 1)2^p$, do the following:

- Compute $\Pi_{2p,k2p}$ and set $A[i] := A[i] + \Pi_{2p,k2p,i}$ for all $(k + 1)2^p \leq i \leq (k + 3)2^p - 2$.
- Compute $\Pi_{k2p,2p}$ and set $A[i] := A[i] + \Pi_{k2p,2p,i}$ for all $(k + 1)2^p \leq i \leq (k + 3)2^p - 2$. 