HOW TO COMPUTE FAST A FUNCTION AND ALL ITS DERIVATIVES

A variation on the theorem of Baur–Strassen

Jacques Morgenstern
The University of Nice
Dept. of Mathematics, Parc Valrose
06034 NICE Cedex, France

Abstract : This note is about a simple and algorithmic proof of the striking result of BAUR–STRASSEN [1] showing that the complexity of the evaluation of a rational function of several variables and all its derivatives is bounded above by three times the complexity of the evaluation of the function itself.

Introduction

The simplicity of the result of BAUR–STRASSEN [1] forces one to believe that a more direct proof by induction should be at hands. My inability to reproduce their proof clearly (after a good meal that is) in a seminar in front of Allan BORODIN and Claude CHRISTEN stimulated my research.

The first time derivatives of algorithms were used to find algorithms for derivatives was in STRASSEN [1] and MORGENSTERN [3][4] independently. But the importance of the paper by BAUR–STRASSEN relies on the possibility to use earlier results about lower bounds on the complexity of several functions to prove lower bounds for single functions.

The present idea gives more insight in the structure of algoritnms and is constructive.

Notation : The notation is mainly the one used in [1] :

K is an infinite field, F a rational function of n variables $x_1,x_2,\dots,x_n$ and $\tilde{F}$ a rational function of n+1 variables $x_1,x_2,\dots,x_n,y$.

The set of partial derivatives is noted $F' = \{\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \dots, \frac{\partial F}{\partial x_n}\}$.

Let o be any arithmetical operation; it is an <u>essential</u> operation if the two operands depend upon the variables or if it is a division and the second operand depends on the variables.

Let A be an algorithm computing F from $x_1,x_2,\dots,x_n,K$ that is

$$A = \{g_1,g_2,\dots,g_u\} \text{ where } g_k = g_{k_1} \circ g_{k_2}, \quad k_1,k_2 < k$$
$$\text{or } g_k \in \{x_1,\dots,x_n\} \cup K.$$

and let

- $s(F)$ be the number of essential multiplication/division in A
- $m(F)$ be the total number of m/d in A
- $T(F)$ the total number of essential operations in A
- $\theta(F)$ the total number of operations in A.

<u>Theorem</u> (BAUR–STRASSEN)

From each algorithm A computing F one can derive an algorithm A' computing F and F' such that

$$\text{(I)} \quad \begin{cases} s(F,F') \leqslant 3s(F) \\ m(F,F') \leqslant 3m(F) \\ T(F,F') \leqslant 5T(F) \\ \theta(F,F') \leqslant 5\theta(F) \end{cases}$$

Those inequalities are <u>independent</u> of the number n of variables.

<u>Proof</u> : by induction on the lenght of the algorithm.

Let us look at the first operation of the algorithm A and let g_1 be the result.

Define $\tilde{F}$ a function of n+1 variables such that

$$\text{(II)} \qquad F(x_1,x_2,\dots,x_n) = \tilde{F}(x_1,x_2,\dots,x_n, g_1(x_1,x_2,\dots,x_n))$$

Namely A induces an algorithm $\tilde{A}$ which computes $\tilde{F}$ from $x_1,\dots,x_n$, g_1 and K in one less operation. By induction hypothesis $\tilde{F}$ satisfies **(I)**.

Taking the derivative of **(II)** we get

$$\text{(III)} \qquad \frac{\partial F}{\partial x_h} = \frac{\partial \tilde{F}}{\partial x_h} + \frac{\partial \tilde{F}}{\partial y}\cdot\frac{\partial g_1}{\partial x_h}, \qquad h = 1,2,\dots,n.$$

Let us examine 6 cases :

<u>1) –</u> $g_1 = c \cdot x_i$ where $c \in K$, $i \in (1,n)$

(III) gives
$$\frac{\partial F}{\partial x_i} = \frac{\partial \tilde{F}}{\partial x_i} + c\cdot\frac{\partial \tilde{F}}{\partial x_i} \quad \text{for } h=i$$
$$\frac{\partial F}{\partial x_h} = \frac{\partial \tilde{F}}{\partial x_n} \quad \text{for } h\neq i$$

$s(F,F') = s(\tilde{F},\tilde{F}')$; $m(F,F') \leqslant m(\tilde{F},\tilde{F}')+1+1$ (1 is for g_1 the other for (III)).

$$T(F,F') \leqslant T(\tilde{F},\tilde{F}')+1$$
$$\theta(F,F') \leqslant \theta(\tilde{F},\tilde{F}')+2.$$

2) - $g_I = x_i \cdot x_j \quad i,j \in (1,n)$

(III) gives $\frac{\partial F}{\partial x_i} = \frac{\partial \widetilde{F}}{\partial x_i} + \frac{\partial \widetilde{F}}{\partial y} x_j$; $\frac{\partial F}{\partial x_j} = \frac{\partial \widetilde{F}}{\partial x_j} + \frac{\partial \widetilde{F}}{\partial y} x_i$

$\frac{\partial F}{\partial x_h} = \frac{\partial \widetilde{F}}{\partial x_h}$ for $h \neq i$ or j.

$s(F,F') \leq s(\widetilde{F},\widetilde{F}')+2+1$ $\qquad T(F,F') \leq T(\widetilde{F},\widetilde{F}')+5$

$m(F,F') \leq m(\widetilde{F},\widetilde{F}')+2+1$ $\qquad \theta(F,F') \leq \theta(\widetilde{F},\widetilde{F}')+5$

3) - $g_1 = c/x_i$

(III) gives $\frac{\partial F}{\partial x_i} = \frac{\partial \widetilde{F}}{\partial x_i} - \frac{\partial F}{\partial y} \left(\frac{c}{x_i}\right) \frac{1}{x_i}$

$\frac{\partial F}{\partial x_h} = \frac{\partial \widetilde{F}}{\partial x_h}$ for $h \neq i$

$s(F,F') \leq s(\widetilde{F},\widetilde{F}')+3$ $\qquad T(F,F' \leq T(\widetilde{F},\widetilde{F}')+4$

$m(F,F') \leq m(\widetilde{F},\widetilde{F}')+3$ $\qquad \theta(F,F') \leq \theta(\widetilde{F},\widetilde{F}')+4$

4) - $g_1 = x_i/x_j$

(III) gives $\frac{\partial F}{\partial x_i} = \frac{\partial \widetilde{F}}{\partial x_i} + \frac{\partial \widetilde{F}}{\partial y} \cdot \frac{1}{x_j}$

$\frac{\partial F}{\partial x_j} = \frac{\partial \widetilde{F}}{\partial x_j} - \frac{\partial F}{\partial y} \frac{x_i}{x_j} \frac{1}{x_j}$

$\frac{\partial F}{\partial x_h} = \frac{\partial \widetilde{F}}{\partial x_h}$ for $h \neq i$ or j

$S(F,F') \leq s(\widetilde{F},\widetilde{F}')+3$ $\qquad T(F,F') \leq T(\widetilde{F},\widetilde{F}')+5$

$m(F,F') \leq m(\widetilde{F},\widetilde{F}')+3$ $\qquad \theta(F,F') \leq \theta(\widetilde{F},\widetilde{F}')+5$

5) - $g_1 = x_i+d \quad d \in K\ , \quad i \in (1,n)$

(III) gives $\frac{\partial F}{\partial x_i} = \frac{\partial \widetilde{F}}{\partial x_i} + \frac{\partial \widetilde{F}}{\partial y}$

$\frac{\partial F}{\partial x_h} = \frac{\partial \widetilde{F}}{\partial x_h}$ for $h \neq i$

s,m are unchanged, $T(\widetilde{F},\widetilde{F}') \leq T(\widetilde{F},\widetilde{F}')+1$ $\qquad \theta(F,F') \leq \theta(\widetilde{F},\widetilde{F}')+1$

6) - $g_I = x_i+x_j$

(III) gives $\frac{\partial F}{\partial x_i} = \frac{\partial \widetilde{F}}{\partial x_i} + \frac{\partial \widetilde{F}}{\partial y}$

$\frac{\partial F}{\partial x_j} = \frac{\partial \widetilde{F}}{\partial x_j} + \frac{\partial \widetilde{F}}{\partial y}$

$\frac{\partial F}{\partial x_h} = \frac{\partial \widetilde{F}}{\partial x_h}$ for $h \neq i$ or f

s,m are unchanged ; $T(F,F') \leq T(\widetilde{F},\widetilde{F}')+2+1$; $\qquad \theta(F,F') = \theta(\widetilde{F},\widetilde{F}')+2+1$.

Putting everything together, using **(I)** for F we see that we get at most the inequalities :

$$s(F,F') \leq s(\widetilde{F},\widetilde{F}')+3 \leq 3s(\widetilde{F}))+3 = 3s(F)$$
$$m(F,F') \leq m(\widetilde{F},\widetilde{F}')+3 \leq 3m(\widetilde{F})+3 = 3m(F)$$
$$T(F,F') \leq T(\widetilde{F},\widetilde{F}')+5 \leq 5T(\widetilde{F})+5 = 5T(F)$$
$$\theta(F,F') \leq \theta(\widetilde{F},\widetilde{F}')+5 \leq 5\theta(\widetilde{F})+5 = 5\theta(F)$$

which proves the theorem since the beginning of the induction is trivial.

The inequalities of the theorem were proved to be independent on n since in each case we added a bounded number of operations.

But if we had taken second derivatives of **(II)** the number of operations added would depend on n; this is unfortunate since from any algorithm A" computing second derivations of F we could deduce an algorithm to perform matrix product [5]:

Let $u=(u_1,u_2,\ldots,u_n)$ and $v(v_1,v_2,\ldots,v_n)$ be two vectors of K^n. Let X and Y be two n by n matrices and let $Z=Z \cdot Y$ be their product. Let $F(u_i,x_{nk},y_{rs})={}^t u\, X \cdot Y v$ be a function, bilinear in u_i, v_j and of total degree 4 with respect to the $2n^2+2n$ variables.

We have
$$\frac{\partial^2 F}{\partial u_i\, \partial v_j} = Z_{ij} \ .$$

And F can be computed in $2n^2+n$ essential m/d steps and $4n^2-1$ total arithmetics.

If we had a theorem to bound the complexity the computation of the second derivatives in term of the complexity of F, we would have an $O(n^2)$ upper bound for matrix product.

Conclusion :

a) - **Practical**

From the proof of the theorem we can deduce a constructive, recursive but backwards way to build an algorithm for the computation of all the partial derivatives of F at a given point from any algorithm computing F at the same point.

b) - **Theoretical**

From the theorem one can deduce lower bounds for the computation of the determinant on the computation of simple single functions [1].

REFERENCES

[1] **Baur,W.**, and **V.Strassen**, The complexity of partial derivatives, preprint

[2] **Strassen,V.**, Vermeidung von Divisionen, Crelle J. für die Reine und Angewandte Mathematik, Math. 264, 184-202, 1973

[3] **Morgenstern,J.** Algorithmes linéaires tangents et complexité, C.R. de l'Académie des Sciences, t.277, p.367 (3 sept.1973)

[4] **Morgenstern,J.**, Linear tangent algorithms and lower bound for the complexity of computation, IFIP (1974)

[5] **Stoss, H.J.**(private communication).

Addendum: Those results could be generalized to other operations like extracting roots or exponentiation since taking derivatives leads to results that are rationaly expressi--ble in term of the data:

$((X)^r)' = (r-1).X^r/X$ etc... (simple differential equation)

The next step is to generalize them to other algebras and also to mechanize the process and write a translator of algorithms to compile the given one and output the algorithm for first partial derivatives.

J.M.

05.15.1984