$\mathbf{P}_{\mathbb{R}} \neq \mathbf{NC}_{\mathbb{R}}$

Felipe Cucker

Report LSI–91–44

$\mathbf{P}_{\mathbb{R}} \neq \mathbf{NC}_{\mathbb{R}}$

Felipe Cucker †
Dept. Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Barcelona 08028
SPAIN
e-mail: cucker@lsi.upc.es

Abstract. In this note, we show the existence of sets of real numbers that can be decided in polynomial time for the Blum, Shub and Smale model of computation but cannot be decided in polylogarithmic parallel time using a polynomial number of processors.

Introduction

In a recent paper ([2]) a theory of computation over an arbitrary ring was devised that introduced ideas and methods from classical Complexity and Computability theories into the area of Algebraic Complexity (see [6] for a survey of this latter subject). A special emphasis was placed on the case in which the ring is $\mathbb{R}$ the field of real numbers. The theory then reflects the kind of computations made in Numerical Analysis or Computational Geometry. For that special case, many basic results have been shown such as the existence of natural NP-complete problems or universal machines.

In a subsequent work ([3]), the existence of natural P-complete problems was also proved but, just as the existence of NP-complete problems left open the question of whether P=NP, the existence of the P-complete ones focuses interest on the question of whether polynomial time equals parallel polylogarithmic time (with a polynomial number of processors). We will prove in this paper that the answer to that last question is NO.

1. Parallel models

There is no unified theory of parallel machines for the real numbers. However, research done in algebraic complexity extensively used circuits as models of computations either as a non–uniform model for getting lower bounds, either combined with some uniformity condition when possible. A recent survey of that subject can be found in [5].

We introduce now a class of circuits togheter with a class of sets defined through them. They are equivalent to the arithmetical networks of [5].

† Partially supported by the ESPRIT BRA Program of the EC under contract no. 3075, project ALCOM, DGICyT PB 89/0379 and UPC PR9014.

Definition. An *algebraic circuit over the reals* with inputs in $\mathbb{R}^n$ is a finite directed graph C, whose nodes have labels from $\mathbb{N} \times \mathbb{N} - \{0\}$, that satisfies the following conditions:

- there are exactly n nodes $v_{01}, \ldots, v_{0n}$ with first index 0, and they have no incoming edges
- all the other nodes v_{ij} are of one of the following types
  1) *arithmetic nodes:* they have an associated arithmetic operation $\{+, -, *, /\}$ and there exist l, k, r, m with $l, k < i$ such that their incoming edges are (v_{lr}, v_{ij}) and (v_{km}, v_{ij}).
  2) *constant nodes:* they have an associated real number γ and no incoming edges.
  3) *sign nodes:* they have a unique incoming edge (v_{km}, v_{ij}) with $k < i$.

To each node we inductively associate a function of the input variables in the usual way. We only note that a sign node with input x returns 1 if $x \geq 0$ and 0 otherwise. Also, we shall call *depth* of the circuit the largest n such that we have nodes v_{nj}, and *size* of the circuit the total number of nodes.

Also, a circuit of depth d is *decisional* if there is only one node v_{d1} at level d, and it is a sign node. We finally define the *accepted set* of a decisional circuit to be the set $S \subset \mathbb{R}^n$ of the points whose image by the associated function is 1.

In order to get a uniform model of parallel computation we should endow families of circuits with some uniformity condition, but the one used in the Boolean case to define the class NC —the generation of the circuits by a machine working in logarithmic space— is meaningless now. One of the remarkable features of the theory of computation over the reals is given by the fact that to obtain complexity classes bounding the used space is irrelevant. Thus, in [7] it is shown that any recursive subset of $\mathbb{R}^\infty$ can be decided by a real Turing machine within linear space.

One possible way, then, of defining uniformity is given by imposing the existence of a real Turing machine which, given input n, generates the n^{th} circuit in time which is polynomic in n. The complexity class defined by such families of circuits having polylogarithmic depth and polynomial size is an analog of the class PUNC defined in the boolean case, which contains NC (see [1]). Another possibility is to require both polynomial time and logarithmic space to the above mentioned real Turing machine. The later requirement allows to define a class more similar to NC.

One can also define models like PRAM's or PRTM's with a polynomial number of processors (real RAM's (see [8]) or real Turing machines) that work within polylogarithmic time and that communicate directly between them or via a shared memory. In the Boolean case, there is a large amount of work done showing the equivalence of those models and the cost of the simulations among them. For computations with real numbers, this is something waiting to be done.

Concerning the result we want to prove, there is no need, however, of using a particular model. The only feature we shall use is that, for all inputs of a given size n and at each moment of the computation, the actual configuration consists of a finite number of non-zero coordinates in the space state, and that at each computing step a new configuration is obtained from the present one modifying some of the coordinates of the space state (at must as many as the number of processors) replacing them by the result of operating (via one of $(+, -, *, /)$) on two other coordinates. These modifications may depend on a set

of Boolean conditions (again at must as many as the number of processors) of the form $x \geq 0$, where x is the value of one of these coordinates. We observe that this is exactly what happens with the circuits above introduced (independently of any uniformity condition) or with any PRAM or PBSS.

2. The theorem

We briefly recall some basic notions in algebraic geometry that will be useful to us in the sequel.

A set $V \subset \mathbb{C}^k$ is called an *algebraic set* when V is the set of all points in $\mathbb{C}^k$ satisfying a system of polynomial equations

$$\begin{aligned} f_1(X_1, \ldots, X_k) &= 0 \\ f_2(X_1, \ldots, X_k) &= 0 \\ &\vdots \\ f_r(X_1, \ldots, X_k) &= 0 \end{aligned}$$

Of course, all the polynomials belonging to the ideal generated by $f_1, \ldots, f_r$ also vanish on V. On the other hand, this ideal is called *definition ideal* of V when all the polynomials vanishing on V belong to it. Hilbert's Nullstellensatz characterizes the ideals of $\mathbb{C}[X_1, \ldots, X_n]$ that are definition ideals of some algebraic set, which turn out to be the radical ones (see [4]).

Also, an algebraic set V is said to be *reducible* when there exist two algebraic sets V_1 and V_2, both different from V, such that $V = V_1 \cup V_2$. It is a basic fact that a set is irreducible iff its definition ideal is prime.

In the sequel we shall be concerned with plane algebraic curves, i.e. curves in $\mathbb{C}^2$ given by a single polynomial in $\mathbb{C}[X, Y]$. More concretely, we shall deal with some Fermat curves which are given by polynomials of the form $X^d + Y^d - 1$, and we shall denote by $\mathcal{F}_d$ the set of its complex points and by $\mathcal{F}_d^{\mathbb{R}}$ its intersection with $\mathbb{R}^2$. We recall that such polynomials are irreducible (since they define non–singular curves in the projective plane) and thus generate prime ideals in $\mathbb{C}[X, Y]$.

Let us now introduce the problem

$$\mathrm{FER} = \{x \in \mathbb{R}^\infty \mid |x| = n \text{ then } (x_1, x_2) \in \mathcal{F}_{2^n}^{\mathbb{R}}\}$$

whose first property is given in the next result.

Proposition 2.1. *The problem* FER *belongs to* P.

Proof: The following algorithm

```
begin
  n := |x|;
  a := x_1;
  b := x_2;
  for i = 1 to n do
    a := a * a;
    b := b * b
  od
  if a + b = 1 then ACCEPT
    else REJECT
  fi
end
```

recognizes FER in linear time. ∎

We can now show our main result.

Theorem 2.2. *For all* $k \in \mathbb{N}$ *and all function* $f : \mathbb{N} \to \mathbb{N}$ *there is no parallel machine accepting* FER *within time* $\log^k n$ *using* $f(n)$ *processors.*

Proof: Let us assume that there is a parallel machine M, as in the statement that solves FER. For any n and any input $(x_1, \ldots, x_n)$ of size n we shall consider the tree of all possible configurations of the machine. For the sake of simplicity, we shall suppose that $x_3 = \cdots = x_n = 1$ without loss of generality. Each configuration can be described by a point in the state space $\mathbb{R}^N$ where now N is a fixed bound that only depends on n.

At each step of the computation we modify some of the coordinates replacing them by the result of operating (via one of $(+, -, *, /)$) on two other coordinates. Those modifications can depend on Boolean conditions of the form

$$Q_i(x_1, x_2) \geq 0$$

—where $Q_i(x_1, x_2)$ is the content of cell i and is a rational function in x_1 and x_2. Those Boolean conditions will produce a branching in our tree of configurations. Moreover, since the number of processors is bounded by $f(n)$, the fan-out in our branchings is bounded by $2^{f(n)}$. After $\log^k n$ steps, we shall have a large (but finite) number of leaves that are accepting or rejecting leaves, and FER is the union of the sets of inputs for which the computation leads to an accepting leaf.

For each one of those accepting leaves, the final configuration will consist of at most N rational functions in x_1 and x_2 whose numerator and denominator have a degree which is bounded by $2^{\log^k n}$, since the depth of the tree is $\log^k n$. Thus, all the rational functions $Q_i(x_1, x_2)$ appearing in the Boolean conditions above mentioned have the same bounds for the degrees. We conclude that the set of inputs that are led to a given leaf can be characterized by a finite system of inequalities of the form

$$\bigwedge_{i=1}^{s} Q_i(X_1, X_2) \leq 0 \wedge \bigwedge_{i=1}^{t} R_j(X_1, X_2) < 0$$

where s and t are bounded by $f(n)\log^k n$. By clearing denominators we can replace the rational functions by polynomials with the same (actually twice the) bound for the degrees that one had for the rational functions. Also, expressing an inequality like

$$F(X_1, X_2) \geq 0$$

as the disjunction

$$F(X_1, X_2) = 0 \vee F(X_1, X_2) > 0$$

and then distributing, we can describe FER as a union of sets given by systems of polynomial inequalities of the form

$$\bigwedge_{i=1}^{s} F_i(X_1, X_2) = 0 \wedge \bigwedge_{i=1}^{t} G_j(X_1, X_2) > 0$$

Now, since the curve $\mathcal{F}_{2^n}^{\mathbb{R}}$ is infinite, one of those sets must contain an infinite number of points of the curve. Since the set described by the G_j's is open, it must be non–empty, and then it defines an open subset of $\mathbb{R}^2$. But $\mathcal{F}_{2^n}^{\mathbb{R}}$ is a curve, and therefore we must have $s > 0$.

Finally, all the polynomials F_i, $i = 1, \ldots, s$, vanish on that infinite subset of the curve and, thus, in a 1-dimensional component of the curve. But, since the curve is an irreducible one, this implies that every F_i must vanish on the whole curve. Using the fact that the ideal $(X_1^{2^n} + X_2^{2^n} - 1)$ is prime (and, a fortiori, radical), we conclude that all the F_i are multiples of $X_1^{2^n} + X_2^{2^n} - 1$ which is impossible since their degree is bounded by $2^{\log^k n}$, which is strictly smaller than 2^n. ∎

Remarks 2.3.

i) As one can expect, the preceding argument cannot be applied as it stands in the discrete case. The main obstruction, if we work in $(\mathbb{Z}/p)^2$ is the lack of infinite points in the algebraic curves in that plane.

ii) On the other hand, the same proof applies for machines over $\mathbb{C}$ or over $\mathbb{Q}_p$.

iii) We finally remark that one can prove in the same way, for any function bound B, that sequential time 2^B is different to polylogarithmic time B using 2^B processors.

Acknowledgement. The results in this paper arose from a talk with Mike Shub at Berkeley and subsequently an e-mail dialog. The author is greatly indebted to him.

References

[1] E. Allender; "Characterizations of PUNC and precomputation", *13th ICALP*, Springer LNCS **226**, pp. 1-10, 1986.

[2] L. Blum, M. Shub and S. Smale; "On a theory of computation and complexity over the real numbers: *NP*-completeness, recursive functions and universal machines". *Bulletin of the Amer. Math. Soc.*, vol.**21**, n.**1**, pp.1-46, 1989.

[3] F. Cucker and A. Torrecillas; "Two P-complete problems in the theory of the reals", *18th ICALP*, Springer LNCS **510**, pp. 556-565, 1991.

[4] W. Fulton; *Algebraic Curves.* W. A. Benjamin Inc., New York, 1969.

[5] J. von zur Gathen; "Parallel arithmetic computations: a survey", *Proc. 12th Int. Symp. Math. Found. Comp. Sc.*, LNCS 233, pp.93-112, Springer Verlag, 1986.

[6] J. von zur Gathen; "Algebraic complexity theory", *Ann. Rev. Comput. Sci.*, **3**, pp.317-347, 1988.

[7] C. Michaux; "Une remarque à propos des machines sur Rintroduites par Blum, Shub et Smale", *C. R. Acad. Sc. Paris*, t. **309**, Série I, pp. 435-437, 1989.

[8] F.P. Preparata and M.I. Shamos; *Computational Geometry: an introduction.* Texts and Monographs in Computer Science, Springer Verlag, 1985.