The exclusion algorithm

Jean-Claude Yakoubsohn
Université Paul Sabatier, Toulouse

March 8, 1993

[summary by François Morain]

1. Introduction

Most numerical algorithms that look for the roots of a polynomial P over a field $\mathbf{K}$ first try to find subsets of $\mathbf{K}$ that contain just one root of P. Then, some sort of refining algorithm is used to get a more accurate value of the roots of P (e.g., Newton's algorithm). For this, we refer to [3].

The exclusion algorithm, on the contrary, eliminates large regions of K that do not contain any root of P. After this, the refining algorithms are used in the subsets that were found to have roots of P.

First of all, we describe the exclusion algorithm for computing real roots of polynomials. Then, we briefly describe the changes to be made when trying to localize hypersurfaces.

2. The exclusion algorithm in the 1-dimensional case

Let $P(x) = \sum_{k=0}^{d} a_k x^k$ be a polynomial in $\mathbb{R}[x]$, with $a_d \neq 0$. Let Z denote the (finite) set of real zeroes of $P(x)$. We suppose that we are given a positive real number ρ such that $Z \subset [-\rho, \rho]$. (Such a number can be computed using Cauchy's bound, see [4].) Let $\varepsilon > 0$ be any real number (the precision of the exclusion). The goal of the algorithm is to find a set F_ε such that

$$Z \subset F_\varepsilon \subset Z + [-K\varepsilon, +K\varepsilon]$$

where K is an absolute constant independent of ε.

2.1. The exclusion function. For $x \in \mathbb{R}$, let $M(x, r)$ be the polynomial

$$M(x,t) = |P(x)| - \sum_{k=1}^{d} \frac{|P^{(k)}(x)|}{k!} t^k.$$

It is easy to see that $M(x,t)$ is decreasing and concave, has a positive value in $t = 0$, and tends to $-\infty$ when t tends to $+\infty$, and therefore $M(x,t)$ has a unique positive root noted $m(x)$. We call $m(x)$ the *exclusion function* associated to $P(x)$. Let $d(x, Z)$ denote the distance from x to Z.

Proposition 1. *The function m has the following properties:*

(1) *$m(x) = 0$ if and only if $P(x) = 0$;*
(2) *if $P(x) \neq 0$, then $]x - m(x), x + m(x)[\cap Z = \emptyset$;*
(3) *for all x, y in $\mathbb{R}$, $|m(x) - m(y)| \leq |x - y|$;*
(4) *if $Z \neq \emptyset$, then there is a constant $\alpha > 0$ such that for all x, $\alpha d(x, Z) \leq m(x) \leq d(x, Z)$.*

2.2. A very simple exclusion algorithm. We start from $Z \subset [-\rho, \rho]$. The algorithm runs as follows:

```
function Exclusion(P, x, r, eps)

  if r < eps then return(]x-r, x+r[)
  else
    compute M(x, t)
    if M(x, r) < 0 then
      return(Exclusion(P,x-r/2,r/2,eps) union Exclusion(P,x+r/2,r/2,eps));

end.
```

We start with `Exclusion`$(P, 0, \rho, \varepsilon)$ and at each iteration, we determine whether $Z \subset]x-r, x+r[$ by testing whether $M(x, r) < 0$ or not. This trick is due to X. Gourdon who simplified the method given in [1, 2]. Note that this algorithm always stops as soon as we enter intervals of length less than ε.

A Maple implementation of this would simply be:

```
Exclusion := proc(P, X, x, r, eps) local mxt, d, k, t;
   if r < eps then RETURN((x-r)..(x+r)) fi;
   d:=degree(P, X);
   mxt:=0;
   for k to d do mxt:=mxt+abs(subs(X=x,diff(P,X$k)))*t^k/k! od;
   mxt:=abs(subs(X=x,P))-mxt;
   if subs(t=r, mxt) < 0 then
      RETURN(Exclusion(P,X,x-r/2,r/2,eps), Exclusion(P,X,x+r/2,r/2,eps))
   fi;
end:
```

If we try it on $P(X) = X^3 + X + 1$ and $\rho = 3$, we get:

```
> Exclusion(X^3+X+1,X,0.,3.,0.001);
           -.6826171876 .. -.6811523438, -.6811523438 .. -.6796875000
```

whereas

```
> fsolve(X^3+X+1);
                                              -.6823278038
```

An iterative exclusion algorithm is given in [2], together with an analysis of the complexity of the algorithm. In particular, it is shown that the iterative version enables one to get

$$Z \subset F_\varepsilon \subset Z + [-K\varepsilon, +K\varepsilon]$$

with $K = 2/\alpha$ with α defined above. The complexity of the algorithm is then shown to be

$$O(d^2|\log\varepsilon| + d|\log\varepsilon|\log|\log\varepsilon|).$$

3. Localization of hypersurfaces

Let $P(x)$ be a polynomial in $\mathbf{K}[x]$, where $x = (x_1, \dots, x_n)$ with degree d. This time the set of zeroes of P need no longer be finite. We suppose first that Z has a point at infinity (which means we treat the affine case).

Put

$$M(x,t) = |P(x)| - \sum_{k=1}^{d} b_k t^k$$

with

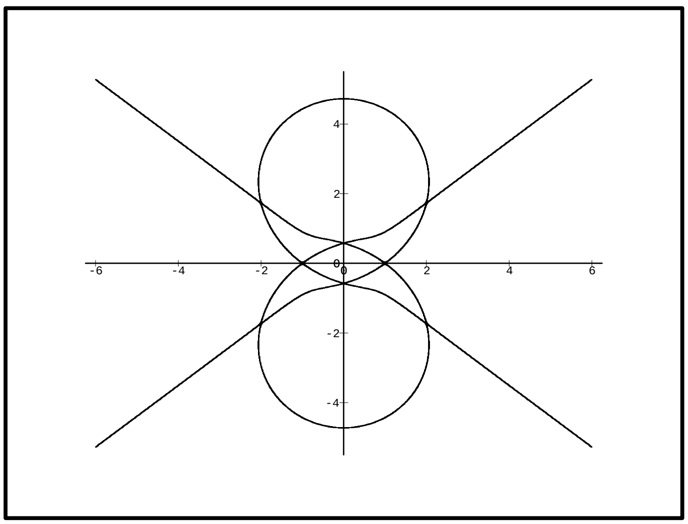
$$b_k = \frac{1}{k!} \sum_{1 \le i_1 \le \dots \le i_k \le n} \left| \frac{\partial^k P(x)}{\partial x_{i_1} \cdots \partial x_{i_k}} \right|.$$

As in the 1-dimensionnal case, $M(x,t)$ is concave and decreasing, implying it has a unique positive root $m(x)$. We have:

PROPOSITION 2. (1) *$m(x) = 0$ if and only if $P(x) = 0$. Moreover x is a singular point of Z if and only if $m(x)$ is a root of $M(x,t)$ of multiplicity greater than 2;*
(2) *if $P(x) \neq 0$, then $B_o(x, m(x)) \cap Z = \emptyset$;*
(3) *$m(x)$ is continuous and semi-algebraic.*
(4) *let F be a semi-algebraic compact subset of $\mathbf{K}^n$. There is a constant $\alpha > 0$ and an integer $n_1 \neq 0$ such that for all $x \in F$, one has*

$$\alpha d(x, Z)^{n_1} \leq m(x) \leq d(x, Z).$$

Using this, we can write a very naïve program that localizes Z in a compact F. It is just the same algorithm as in the 1-dimensionnal case, where we replace the interval $]x-r, x+r[$ with the open ball $B_o(x,r)$ and we replace dichotomy in two subintervals by dichotomy in four regions of the plane (in the case $\mathbf{K} = \mathbb{R}^2$).



The resulting algorithm has been programmed by Bruno Salvy in MAPLE and gives very good results. For example, the curve of Gergueb, corresponding to

$$\begin{aligned} P(x,y) &= -7 + 9y^8 - 204y^6 + 70y^4 - 7x^8 + 28x^6 - 42x^4 + 28x^2 - 52x^2y^2 \\ &\quad +68x^2y^4 + 20x^2y^6 + 44x^4y^2 + 6x^4y^4 - 12x^6y^2 + 20y^2 \end{aligned}$$

was drawn using MAPLE (see figure).

The reader interested in an iterative version of this algorithm, together with an analysis of its complexity is referred to [1].

Bibliography

[1] Dedieu (Jean-Pierre) and Yakoubsohn (Jean-Claude). – Localization of an algebraic hypersurface by the exclusion algorithm. *Applicable Algebra in Engineering, Communication and Computing*, vol. 2, 1992, pp. 239–256.

[2] Dedieu (Jean-Pierre) and Yakoubsohn (Jean-Claude). – Computing the real roots of a polynomial by the exclusion algorithm. *Numerical Algorithms*, vol. 4, 1993, pp. 1–24.

[3] Gourdon (Xavier). – *Algorithmique du théorème fondamental de l'algèbre.* – Technical Report n° 1852, Institut National de Recherche en Informatique et en Automatique, February 1993.
[4] Marden (M.). – Geometry of polynomials. In *AMS Surveys.* – AMS, second edition, 1966.