INTERVAL MATHEMATICS AND INTERVAL CONSTRAINT PROGRAMMING
ICCM-2004 WORKSHOPS P. 289–302

# A reliable curve tracing method

Dominique Michelucci, Dominique Faudot*

## Introduction

Curve tracing is a problem encountered in CAD during the calculation of intersection between surfaces or the resolution of systems of equations by homotopy or in linear programming with the interior method. The curve is defined by $n-1$ independent equations of $n$ real variables $f(x)=0$ with $f=(f_1,\dots,f_n)$ and $x=(x_1,\dots,x_n)$. Here only algebraic equations are considered. The curve is thus smooth almost everywhere. Tracing is classically done by two methods:

**M1:** By piecewise linear approximation. For any simplex of $\boldsymbol{R}^n$ there is only one affine function $l_i=(x_1,\dots,x_n)$ which takes the same values as the function $f_i=(x_1,\dots,x_n)$ at the $n+1$ vertices (the hypersurface in $\boldsymbol{R}^n$ which equation is $f_i=(x_1,\dots,x_n)=0$) the zero level set of which is a hyperplane inside the simplex. The intersection between the $n-1$ hyperplanes $l_1(x_1,\dots,x_n)=\dots=l_{n-1}(x_1,\dots,x_n)=0$ defines a line, which approaches the curve inside the simplex when this line cuts the simplex. Knowing a cut simplex, one gradually follows then the curve, from one simplex to another.

**M2:** By prediction and correction. Let $a$ be a known regular point of the curve and $t$ a vector tangent to the curve in $a$. The method predicts at first that a point $p_k=a+kt/\|t\|$ is close to the curve and corrects this point by some variant of the Newton method. In practice, the step of progression $k$ is given empirically. Possible heuristics are: one gives $k$ a priori. If the Newton's method converges in few stages and if the tangent at the new point is not too distant from $t$, then $k$ was probably correct; otherwise one starts again with smaller $k$. We can however jump from a branch of $f(x)=0$ to another one. One secures oneself then by using small values for $k$. This article proposes a reliable method to choose the step $k$.

An alternative of the second method considers the osculatory circle of the curve at the point $a$ to control the step $k$; this allows a better prediction but does not guarantee absolutely against the risk to jump from a branch to another as opposed to [7, 3].

**Example.** Let us consider the equation in $\boldsymbol{R}^2$ $c_1(x,y)c_2(x,y)=0$ where $c_1$ and $c_2$ are two circle equations. For a point $a$ on $c_1$, the osculatory circle will be $c_1$ and one will deduce from them a step $k$ and a point $p_k=a+kt/\|t\|$ independently of $c_2$. It is thus enough to place $c_2$ judiciously (for example $p\in c_2$, $c_2$ tangent with the line $ap$, through points $a$ and $p$, and $c_1\cap c_2=\emptyset$, so that the correction step converges towards a point of $c_2$ rather than of $c_1$.

*LE2I, Informatique, Université de Bourgogne, BP 47870, 21078 Dijon Cedex, FRANCE. Emails: michelucci@u-bourgogne.fr; faudot@u-bourgogne.fr

# 1. Notations and conventions

**1.1. Compatible norms, upperbounds.** One uses the sup norm for the vectors $\|x\| = \max\bigl(|x_1|, \ldots, |x_n|\bigr)$, $x^t$ is the transpose of $x$. The norm of the matrices $\|M\|$ is sup $\|xM\|$ for $\|x\| = 1$ and is here equal to $\max_{j=1}^{n} \sum_{i=1}^{n} |M_{i,j}|$ where $M_{i,j}$ is the element of line $i$ column $j$. In other words, the norm of $M$ is the greatest value obtained by making in each column the sum of the absolute values. One will need to compute an upper bound of the norm of a product of two matrices. One can certainly carry out the product of two matrices but the following upper bound $(U_1)$ requires only the product of a vector by a matrix

$$\|AB\| \le \|a\hat{B}\|, \quad \text{where } a_j = \sum_{i=1}^{n} |A_{ij}| \quad \text{and } \hat{B}_{ij} = |B_{ij}| \tag{1}$$

e.g.

$$\left\| \begin{pmatrix} -1 & 2 \\ 3 & -4 \end{pmatrix} \begin{pmatrix} -5 & -6 \\ 7 & -8 \end{pmatrix} \right\| \le \left\| \bigl(|-1| + |3| \quad |2| + |-4|\bigr) \begin{pmatrix} |-5| & |-6| \\ |7| & |-8| \end{pmatrix} \right\|.$$

An upper bound $(U_2)$ of $\|AB\|$ even faster and cruder is

$$\|AB\| \le \|A\| \cdot \|B\|. \tag{2}$$

One will need an upper bound of the sum of two matrices. One can carry out the calculation of the sum but another upper bound is possible $\|A + B\| \le \|A\| + \|B\|$.

The ball$(a, R)$ is the set of points $x$ such that $\|x - a\| \le r$. Such a ball is in fact an hypercube of center $a$ and half side $r$. The Jacobian of $f(x) = 0$ with $x = (x_1, \ldots, x_n)$ and $f = (f_1, \ldots, f_m)$ is the matrix $\dfrac{\partial f}{\partial x}$. Let $\alpha$ be a multi index, $|\alpha|$ is a notation for $\alpha_1 + \ldots + \alpha_n$, $\left(\dfrac{k}{\alpha}\right)$ for $\dfrac{k!}{\alpha_1! \ldots \alpha_n!}$ with $k \in \boldsymbol{N}$, $x^\alpha$ for $x_1^{\alpha_1} \ldots x_n^{\alpha_n}$, $\dfrac{\partial^k f}{\partial x^\alpha}$ for $\dfrac{\partial^k f}{\partial x_1^{\alpha_1} \ldots \partial x_n^{\alpha_n}}$ with $k = |\alpha|$.

Other authors use a dual convention and write $Mx$ for the product of a vector $x$ by a matrix $M$, i.e. this dual convention regards $x$ as a vector column. It thus uses the dual norm of ours $\|M\| = \sup_{\|x\|=1} \|Mx\| = \max_{i=1}^{n} \sum_{j=1}^{n} |M_{ij}|$. In the same way, this convention considers a jacobian, which is the transpose of ours.

**1.2. Naive Arithmetic of Intervals (IA).** Naive IA [6] calculates on intervals $[v_0, v_1]$ where $v_0 \le v_1$ are two standard floating-point numbers. We are using the usual rules of IA. The IA has for principal interest to give perfectly reliable results, contrarily to floating-point arithmetic (FPA). It does not modify the theoretical effectiveness of the methods: the elementary arithmetic operations on intervals are carried out in constant time.

But IA over-estimates the width of the intervals and does not permit the comparison of two values with overlapping intervals. The IA is today performed by software and is approximately 4 times slower than the standard FPA. One can however hope that it will be soon available on all the arithmetic processors.

IA is used in two very distinct ways:

- the intervals are sharp, at least initially. The intervals are then used only to control the rounding errors of floating point arithmetic.
- the intervals are broad, even initially. The IA is then a tool for numerical analysis by intervals. The amplitude of the intervals grows quickly during calculations. We will use IA according to this second mode in section 6 and elsewhere according to the first mode. An alternative was also proposed by [2] using affine IA.

## 2. The principle of our method

The curve is defined by $f(x) = 0$ with $x = (x_1, \ldots, x_n)$ and

$$f(x) = \big(f_1(x), \ldots, f_{n-1}(x)\big) = (0, \ldots, 0).$$

The known point is $a = (a_1, \ldots, a_n)$, $a$ is a regular point: the rank $(n-1)$ of $\dfrac{\partial f}{\partial x}(a)$ is maximal. The tangent in $a$ is $t = (t_1 \ldots t_n)$ in other words $t\dfrac{\partial f}{\partial x}(a) = 0$. One supposes moreover that $\|t\| = \max\big(|t_1| \ldots |t_n|\big) = 1$. To use the fixed-point theory, it is first assumed that each point of the curve $f(x) = 0$ is solution of a system of $n$ unknown equations and $n$ unknown variables.

Let $s_k(x) = \big(f(x), f_n(x,k)\big)$, where $f_n(x,k) = 0$ is an additional equation, parameterized by the real $k$, $k$ being such that $f_n(a,0) = 0$ and measures the progression along the curve. A natural choice for the additional equation $f_n(x,k) = 0$ is $f_n(x,k) = (x - a - kt) \cdot t^t = t_1(x_1 - a_1 - kt_1) + \ldots + t_n(x_n - a_n - kt_n) = 0$.

In other words, $a$ is seen as the point of intersection between the curve $f(x) = 0$ and the hyperplane passing through $a$ and normal with $t$; this hyperplane will be translated by a vector $kt$ to traverse the curve, close to $a$. The jacobian of the system $s_k(x) = 0$, independent of $k$, is $s' = s'_k = \dfrac{\partial s_k}{\partial x} = \Big(\dfrac{\partial f}{\partial x}, t'\Big)$.

For a given value of $k$, the corresponding point of the curve, solution of $s_k(x) = 0$ will be calculated by the quasi Newton iteration $x^{(1)} = a + kt$, $x^{(n+1)} = QN_k x^{(n)}$ where the quasi Newton function $QN(x)$ is defined by $QN_k(x) = x - s_k(x)\big[s'(a)\big]^{-1}$ (note $QN_k(a) = a + kt$). One could thus start from $x^{(1)} = a$ instead of $x^{(1)} = a + kt$.

The larger $k$ the better, for advancing quickly along the curve; but it is also needed that the convergence of the correction step is guaranteed and fast. According to the fixed-point theory, the iteration $x^{(n+1)} = QN(x^n)$ starting from the initial point $x^{(1)}$ converges if one can find a neighborhood (actually a ball) $B$ containing $x^{(1)}$ and satisfying the conditions $\mathrm{C}_1$ and $\mathrm{C}_2$ defined below.

**$\mathbf{C_1}$.** The contractivity condition: $QN$ is contracting in the ball $B$, i.e. for any couple of items $x$, $y$ in $B$, $\|QN(x) - QN(y)\| \le c\|x - y\|$ where $c < 1$ is the factor of contraction. One will even impose $c \le 1/2$ to ensure that $QN$ converges quickly: with each iteration, the distance to the solution is divided by at least 2. Since $\|QN(x) - QN(y)\| = \|QN(x) - QN(x + (y - x))\| \le \|y - x\| \max_{z \in B}(QN'(z))$ one will impose that

$$\max_{x\in B}\big(\|QN'(z)\|\big) = \big\|QN'(B)\big\| \le 1/2 \tag{3}$$

(contractivity condition CC).

**C$_2$.** Stability condition: the image of the ball $B$ by $QN$ $QN(B)$ is such that $QN(B) \subset B$. We will use in fact a condition stronger than C$_2$, and thus sufficient to guarantee C$_2$, and which is more easily computable. Let us suppose that $r$ is known such that the ball $B(a,r)$ satisfies the condition of contractivity C$_1$. Then a sufficient condition to satisfy C$_2$ is $\|QN_k(a) - a\| \le r/2$.

**Sufficient condition (SC).** Indeed by assumption $x \in B = \text{ball}(a,R) \Rightarrow \|x-a\| \le R$ and $\|QN'(B)\| \le 1/2$ and $\|QN_k(a) - a\| \le R/2$. Then

$$\|QN_k(x)-a\| \le \|QN_k(x)-QN_k(a)\|+\|QN_k(a)-a\| \le \frac{1}{2}\|x-a\|+\frac{R}{2} \le \frac{R}{2}+\frac{R}{2} = R$$

that implies $x \in B$. However, the SC is equivalent to $k \le R/2$.

Really, by assumption $\|QN_k(a) - a\| \le \dfrac{R}{2}$. But $QN_k(a) = a + kt$. Then

$$\|QN_k(a) - a\| = \|kt\| = |k| \cdot \|t\| = |k| \le \frac{R}{2}.$$

It is thus very simple to find $k$ knowing $R$. It is enough to take $k = R/2$. This choice is probably not optimal but this very simple value is guaranteed. The entire problem is thus reduced to find $R$, such that in the ball $B(a,R)$ the CC (3) is guaranteed. $QN'(x)$ is the jacobian of $QN(x)$: $QN'_k = \dfrac{\partial QN}{\partial x} = Id - s'(x)(s'(a)))^{-1}$ is independent of $k$ and will be denoted by $QN'$. Defining $\sigma = x - a = (\sigma_1, \dots, \sigma_n)$ the jacobian of $QN_k$ is

$$\begin{aligned}
QN'(a+\sigma) &= Id - s'(a+\sigma)\big(s'(a)\big)^{-1} = s'(a)\big(s'(a)\big)^{-1} - s'(a+\sigma)\big(s'(a)\big)^{-1}\\
&= -\big(s'(a+\sigma) - s'(a)\big)\big(s'(a)\big)^{-1}
\end{aligned}$$

$$= -\begin{pmatrix}
\frac{\partial f_1}{\partial x_1}(a{+}\sigma) - \frac{\partial f_1}{\partial x_1}(a) & \dots & \frac{\partial f_{n-1}}{\partial x_1}(a{+}\sigma) - \frac{\partial f_{n-1}}{\partial x_1}(a) & 0\\
\frac{\partial f_1}{\partial x_2}(a{+}\sigma) - \frac{\partial f_1}{\partial x_2}(a) & \dots & \frac{\partial f_{n-1}}{\partial x_2}(a{+}\sigma) - \frac{\partial f_{n-1}}{\partial x_2}(a) & 0\\
\dots & \dots & \dots & 0\\
\frac{\partial f_1}{\partial x_n}(a{+}\sigma) - \frac{\partial f_1}{\partial x_n}(a) & \dots & \frac{\partial f_{n-1}}{\partial x_n}(a{+}\sigma) - \frac{\partial f_{n-1}}{\partial x_n}(a) & 0
\end{pmatrix}(s'(a))^{-1} = -KM.$$

The CC also ensures that in the ball $B(a,R)$ for given $k$, the system $s_k(x) = 0$ has at most one solution.

## 3. The naive method

The naive method calculates $QN'(a+\sigma)$ symbolically either by using the Taylor's formula, or by developing $\dfrac{\partial f_i}{\partial x_j}(a+\sigma) - \dfrac{\partial f_i}{\partial x_j}(a)$ naively. Then it multiplies $K$

by $M$, then it deduces for each column a polynomial in $r \geq \|\sigma\|$ which is an upper bound of the sum of the absolute values of the elements in this column. The largest upper bound is an upper bound of the norm of the matrix $QN'(a+\sigma)$. These $n$ polynomials $p_i(r)$ are null at 0, and increase with $r$: it is easy by dichotomy to find $r_i$ such that $p_i(r_i) = \dfrac{1}{2}$: $R = \min_{i=1}^{n}(r_i)$. The course of the naive method is illustrated on the following example.

### 3.1. Example:

$$n = 3, \quad f(x) = (x_1^3 - x_2^3 + 3x_1x_2^2 - 3x_1^2x_2 - x_3, x_2^2 + x_3^2 - 1) = (0,0), \tag{4}$$
$$a = (1,1,0); \quad t = (1,0,0).$$

The auxiliary equation is $f_3(x,k) = x_1 - 1 - k$. The jacobian of $s_k$ is

$$s'(k) = \begin{pmatrix} 3x_1^2 + 3x_2^2 - 6x_1x_2 & 0 & 1 \\ -3x_1^2 - 3x_2^2 + 6x_1x_2 & 2x_2 & 0 \\ -1 & 2x_3 & 0 \end{pmatrix}. \tag{5}$$

The inverse matrix of

$$s'(a) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ -1 & 0 & 0 \end{pmatrix} \text{ is } s'(a)^{-1} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0.5 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

### 3.2. Naive method on Example 3.1:

$$QN'(a+\sigma) = -\begin{pmatrix} 3\sigma_1^2 + 3\sigma_2^2 - 6\sigma_1\sigma_2 & 0 & 1 \\ -3\sigma_1^2 - 3\sigma_2^2 + 6\sigma_1\sigma_2 & 2\sigma_2 & 0 \\ 0 & 2\sigma_3 & 0 \end{pmatrix} M$$
$$= -\begin{pmatrix} 0 & 0 & -3\sigma_1^2 - 3\sigma_2^2 + 6\sigma_1\sigma_2 \\ 0 & \sigma_2 & 3\sigma_1^2 + 3\sigma_2^2 - 6\sigma_1\sigma_2 \\ 0 & \sigma_3 & 0 \end{pmatrix},$$
$$\big\|QN'(a+\sigma)\big\| \leq \left\| \begin{pmatrix} 0 & 0 & |-3|r^2 + |-3|r^2 + |6|r^2 \\ 0 & r & |3|r^2 + |3|r^2 + |-6|r^2 \\ 0 & r & 0 \end{pmatrix} \right\|,$$

where $\|\sigma\| = \max\big(|\sigma_1|, |\sigma_2|, |\sigma_3|\big) \leq r$ and $\big\|QN'(a+\sigma)\big\| \leq \big\|(0, 2r, 24r^2)\big\|$.

In practice, $a$ is not exactly on the curve, and $M$ is not exactly the inverse of $s'(a)$. If floating intervals are used, calculations yield, very naturally, upper bounding polynomials. The constant term will not be exactly null, but a small number such as $10^{-9}$ or $10^{-6}$. This is not awkward, and allows on the contrary to take into account and to control very naturally the numerical inaccuracy. The most important is that the upper bounding polynomials are almost null at 0.

The main disadvantage of the naive method is its cost: its use, implicit or explicit, of the Taylor development requires the calculation of all the terms $\dfrac{\partial^{|\alpha|} f_i(a)}{\partial x^\alpha}$ They are numerous, even when the terms $f_i$ are sparse. This method is inspired by [1].

## 4. Solution using Interval Arithmetic

Let $p(\sigma)$ be one of the $n(n-1)$ polynomials $\dfrac{\partial f_i}{\partial x_j}(a+\sigma) - \dfrac{\partial f_i}{\partial x_j}(a)$ of the matrix $K$. We want to find an interval around $p(\sigma)$ using $[p_1^-, p_1^+]\sigma_1 + \ldots + [p_n^-, p_n^+]\sigma_n$ where $p_i^-$ and $p_i^+$ are lower and upper bounds of $\dfrac{\partial p}{\partial x_i}\big(x \in \mathrm{ball}(a,r)\big)$. We can use other range methods but it is necessary that $\|\sigma\| = 0 \Rightarrow p(\sigma) = 0$. The intervals $[p_i^-, p_i^+]$ for $\dfrac{\partial p}{\partial x_i}(x \in \mathrm{ball}(a,r))$ may be evaluated by a classical IA, for instance the naive one or an affine one if the $r$ value is known. The computation of $r$ gives a value of $R$, called $v(r)$ because it is depending on $r$.

Suppose in Example 3.1 that $r = 1$, i. e.

$$x \in B = \mathrm{ball}(a,1) = ([0,2],[0,2],[-1,1]).$$

Intervals around $\dfrac{\partial f_1}{\partial x_1}(B)$:

$$\frac{\partial f_1}{\partial x_1}(x) = 3x_1^2 + 3x_2^2 - 6x_1x_2 \quad \text{and} \quad \frac{\partial f_1}{\partial x_1}(a) = 0,$$

$$\frac{\partial^2 f_1}{\partial x_1^2}(x) = 6x_1 + 6x_2 \Rightarrow \frac{\partial^2 f_1}{\partial x_1^2}(B) \in [-12,12];$$

$$\frac{\partial^2 f_1}{\partial x_1 x_2}(x) = -6x_1 + 6x_2 \Rightarrow \frac{\partial^2 f_1}{\partial x_1 x_2}(B) \in [-12,12];$$

$$\frac{\partial^2 f_1}{\partial x_1 x_3}(x) = 0 \Rightarrow \frac{\partial^2 f_1}{\partial x_1 x_3}(B) \in [0,0].$$

Thus $\dfrac{\partial f_1}{\partial x_1}(B) \in 0 + [-12,12]\sigma_1 + [-12,12]\sigma_2$. Similarly for other derivatives. Finally, we obtain

$$\begin{aligned} QN'(a+\sigma) &= -\begin{pmatrix} [-12,12]\sigma_1 + [-12,12]\sigma_2 & 0 & 0 \\ [-12,12]\sigma_1 + [-12,12]\sigma_2 & [2,2]\sigma_2 & 0 \\ 0 & [2,2]\sigma_3 & 0 \end{pmatrix} M \\ &= \begin{pmatrix} 0 & 0 & [-12,12]\sigma_1 + [-12,12]\sigma_2 \\ 0 & [1,1]\sigma_2 & [-12,12]\sigma_1 + [-12,12]\sigma_2 \\ 0 & [1,1]\sigma_3 & 0 \end{pmatrix}, \end{aligned} \tag{6}$$

$$\big\|QN'(a+\sigma)\big\| \leq \left\| \begin{pmatrix} 0 & 0 & 24r \\ 0 & r & 24r \\ 0 & r & 0 \end{pmatrix} \right\| = \big\|(0, 2r, 48r)\big\|. \tag{7}$$

The $n$ computed polynomials in $r$ have degree 1 and their constant term is ideally null. Taking into account the numerical inaccuracy can make them non null, but they remain of low magnitude, say $10^{-6}$. To find the value of $r$ where such a polynomial is equal to $1/2$ is trivial.

By studying the ball $(a, r = 1)$ we find $v(1) = 1/96 = 0.0104166$. As $v(1)$ is much smaller than 1, the value of $r$, one can think that one may find it beneficial to reduce the radius $r$ of the studied ball; this will reduce the width of the intervals of $\dfrac{\partial^2 f_i}{\partial x_j \partial x_k}(B)$ thus will decrease the (upper bound of the) norm of $QN'$ and will make it possible to upper bound $v(r)$.

Thus, if one starts again calculation with $r = 0.25$, one finds $v(0.25) = \dfrac{1}{24} = 0.04166$. While trying with $r = 0.1$, one would find $v(0.1) = \dfrac{1}{2 \cdot 4.8} = 0.104166$. Since this value of $v(r)$ goes out of the studied ball, of radius $r = 0.1$, this value is not safe, but however we can deduce that the value of $R = r = 0.1$ is indeed correct.

The bounds of $\dfrac{\partial f_i}{\partial x_1}$ for each function $f_i$ require the evaluation by the IA of $\dfrac{1}{2}n(n+1)$ functions, that is to say $O(n^3)$ functions to evaluate to bound the elements of $K$. Note that a derivative cannot have more monomials than the initial function. When the intervals of the elements of $K$ are calculated, several ways of upper bounding $\|KM\|$ and thus of calculating $R$, are possible.

We can calculate products $KM$. It costs $O(n^3)$ products between elements of the matrices $K$ and M, each product costing $O(n)$ floating point operations, which makes $O(n^4)$ floating point operations for the product by $M$ (this estimate is pessimistic since $K$ is often sparse).

One can speed up this part, if a more pessimistic value $v(r)$ is accepted, by using $U_1$ (6), or even $U_2$ (7). This is detailed in the following section. Calculating $v(1)$ with $U_1$ (6):

$$\begin{aligned}
\big\|QN'(a+\sigma)\big\| &= \left\| \begin{pmatrix} [-12,12]\sigma_1 + [-12,12]\sigma_2 & 0 & 0 \\ [-12,12]\sigma_1 + [-12,12]\sigma_2 & [2,2]\sigma_2 & 0 \\ 0 & [2,2]\sigma_3 & 0 \end{pmatrix} M \right\| \qquad (8) \\
&\le \left\| (24\sigma_1 + 24\sigma_2, 2\sigma_2 + 2\sigma_3, 0) \begin{pmatrix} |0| & |0| & |-1| \\ |0| & |0.5| & |0| \\ |1| & |0| & |0| \end{pmatrix} \right\| \\
&= \big\|(0, 2r, 48r)\big\| = 48r. \qquad (9)
\end{aligned}$$

Finally, we get $R = 1/96$; nothing is lost. The cost is $O(n^3)$ functions to evaluate. The cost to go from equation (8) to (9) is $O(n^2)$ operations, the product with $M$ costs $O(n^3)$ operations.

Choosing $r$: $v(r)$ is a decreasing function of $x \in \boldsymbol{R}^+$: $v(0) = \infty$ and $v(r)$ is decreasing with $r$. We want $r$ such that $R = \min(r, v(r))$ is maximum. In other words, one wants $r$ such that $r = v(r)$: it is inevitably the sought optimum $R$. The simplest is to use the dichotomy. We know $r_0$ such that $v(r_0) > r_0$: $r_0 = 0$ is correct. One searches then $r_1$ such that $v(r_1) \le r_1$:

$$r_1 := 1; \quad \text{while} \ \ (v(r_1) < r_1) \ \ \text{do} \ \{r_0 := r_1;\ r_1 := 2r_1\}.$$

To determine the interval containing $R$, rather than to start from $[0,1]$, one could also start from intervals found in the preceding prediction. Then we use dichotomy inside $[r_0, r_1]$ until knowing $r_0$ with a sufficient relative precision, for example $r_1^{(n)} - r_0^{(n)} < r_1^{(n)}/10$, we choose $R = r_0^{(n)}$. A logarithmic number of evaluations of $v(r)$ is necessary to determine $R$.

## 5. Quadratic systems

**5.1. The interest.** $f$ is quadratic when the total degree of all the monomials of $f_i$ is at most 2. This case is interesting because:

1. Any algebraic system can be reduced to a quadratic system, with the help of the addition of a logarithmic number (using repeated squaring) of variables and equations. For example, a monomial $x_1^3 x_2$ will be replaced by the monomial $y_1 y_2$ where $y_1$ and $y_2$ are two auxiliary variables, and by the two quadratic equations $y_1 - x_1^2 = 0$ and $y_2 - x_1 x_2 = 0$.

2. In a quadratic system $f(x) = 0$ all $\dfrac{\partial f_i}{\partial x_j}(x)$ are constant polynomials, possibly null, and thus independent of the point $a$: they are calculable once and for all. The matrix $K$ is calculable once and for all, just as an upper bound of $\|K\|$. More precisely

$$K_{ij} = \frac{\partial f_j}{\partial x_i}(a+\sigma) - \frac{\partial f_j}{\partial x_i}(a) = \sum_{k=1}^{n} \frac{\partial^2 f_j}{\partial x_i \partial x_k}\sigma_k \Rightarrow \|K\| \leq r\|(l_1 \dots l_n)\| = r \times l(f)$$

where $r \geq \|\sigma\|$ and

$$l_i = \sum_{j=1}^{n}\sum_{k=1}^{n}\left|\frac{\partial^2 f_j}{\partial x_j \partial x_k}\right| = \sum_{j=1}^{n}\left|\frac{\partial^2 f_i}{\partial x_j^2}\right| + 2\sum_{1\leq j<k\leq n}\left|\frac{\partial^2 f_i}{\partial x_j x_k}\right|.$$

$l_i$ is the "complexity" of $f_i$ and is 0 when $f_i(x)$ is first degree. $l(f) = \max_{i=1}^{n}(l_i)$ is the complexity of $f$. The smallest is $l$, the largest is $R$. When $l$ is null (in the case of a linear system for example), $R$ is infinite.

**5.2. Computation of *R*.** At least three ways of calculating $R$ are possible:

1. If one applies the crudest $U_2$, $\|QN'\| \leq \|K\| \cdot \|M\|$ with $\|K\| \leq l(f) \times r$ then $\|QN'\| \leq \dfrac{1}{2} \Leftarrow l(f)r\|M\| \leq \dfrac{1}{2} \Leftrightarrow r \leq R = \dfrac{1}{2\|M\|l(f)}$ and one obtains $R$ very quickly. The computing time of $\|M\|$ is $O(n^2)$ and is negligible in front of the calculation of $M$ itself, which is $O(n^3)$, $l(f)$ is a constant, calculated once and for all, in $O(n^3)$[1].

2. If upper bound (6) is applied, it is necessary to multiply the vector

$$\begin{pmatrix} l_1 r & l_1 r & \dots & l_{n-1} r & 0 \end{pmatrix}$$

by $\hat{M}$ which is equivalent to multiply

$$\begin{pmatrix} l_1 & l_1 & \dots & l_{n-1} & 0 \end{pmatrix}$$

by $\hat{M}$. $O(n^2)$ floating-point operations are required. The obtained value of $R$ is at least as good as with upper bound (7) and is often better.

3. Lastly, $KM$ can explicitly be calculated. In the worst case, each term of $K$ (except the last column, which is null) is a linear combination of all $\sigma_i$ and the product $KM$ requires $O(n^4)$ floating point operations. In practice, $K$ is often sparse. This method gives a value of $R$ at least as good as with $U_1$ (6), generally much better. This method is the slowest of the three. These three ways of calculating $R$ also apply when IA is used. The costs are identical.

[1] when $l(f)$ is zero, $R$ is infinite.

### 5.3. Example:

$$n = 2; \qquad f(x) = (x_1^2 + x_2^2 + x_3^2 - 3,\ x_1^2 + x_2^2 + x_3^2 - 2x_1 - 2x_2),$$

$$f'(x) = \begin{pmatrix} 2x_1 & 2x_1 - 2 \\ 2x_2 & 2x_2 - 2 \\ 2x_3 & 2x_3 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 2\sigma_1 & 2\sigma_1 & 0 \\ 2\sigma_2 & 2\sigma_2 & 0 \\ 2\sigma_3 & 2\sigma_3 & 0 \end{pmatrix}.$$

Then

$$\|K\| = \big\|\big(|2\sigma_1| + |2\sigma_2| + |2\sigma_3|,\ |2\sigma_1| + |2\sigma_2| + |2\sigma_3|,\ 0\big)\big\| \Rightarrow$$
$$\|K\| \le \big\|\big(6r, 6r, 0\big)\big\| = 6r \quad \text{where } (6,6,0) = (l_1, l_2, 0).$$

Suppose that $a = (1,1,1)$ then $t = (1,-1,0)$,

$$s'(a) = \begin{pmatrix} 2 & 0 & 1 \\ 2 & 0 & -1 \\ 2 & 2 & 0 \end{pmatrix}, \qquad M = \begin{pmatrix} 0.25 & 0.25 & 0 \\ -0.25 & -0.25 & 0.5 \\ 0.5 & -0.5 & 0 \end{pmatrix}.$$

First upper bound:

$$r \le R = \frac{1}{2\|M\|l(f)} = \frac{1}{2 \cdot 1 \cdot 6} = 0.083333.$$

Second upper bound:

$$\|QN'\| \le r\big\|(l_1, l_2, 0)\hat{M}\big\| = \big\|(6r, 6r, 0)\hat{M}\big\|$$
$$= \|(3r, 3r, 3r)\| = 3r; \quad 3r \le \frac{1}{2} \Leftrightarrow r \le R = \frac{1}{2 \cdot 3} = 0.166667.$$

Third upper bound:

$$\|QN'\| \le \|KM\| = \begin{pmatrix} 2\sigma_1 & 2\sigma_1 & 0 \\ 2\sigma_2 & 2\sigma_2 & 0 \\ 2\sigma_3 & 2\sigma_3 & 0 \end{pmatrix} \le \left\| \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right\| = 0r,$$
$$0r \le \frac{1}{2} \Leftrightarrow r \le R = \infty.$$

Ideally (without rounding errors due to floating point arithmetic), the method detects that the curve is a straight line, i.e. $R$ is infinite. In practice, with the use of IA to control the rounding errors, $\|KM\|$ will be found not null but very small, for instance $[0, 10^{-6}]r$ which will give finally $R = [5 \cdot 10^{-5}, \infty]$ and $R = 5 \cdot 10^{-5}$.

## 6. Singular or quasi-singular points

The preceding method no longer applies in the presence of a singular point, or of a quasi singular point, i. e. when the value found for $R$ causes an underflow of the floating-point arithmetic. How to solve, or avoid, this kind of problem?

The simplest solution is to consider the piecewise linear approximation (PLA) of the curve for given parameter $\mu$ (i.e $\mu$ is the length of the sides of simplices partitioning $\boldsymbol{R}^n$ as the real curve). The algebraic formulation is just a shortcut, a convenience to describe, in fact, its piecewise linear approximation. This principle may be extended from a system of equations to a C.S.G. tree (a binary tree, which nodes are boolean operations). The PLA (of a CSG tree or of a curve) has many good properties:

1. The PLA is mathematically well defined.
2. The topology of the PLA is simple: no singular points.
3. PLA is calculable, and even quickly calculable.
4. The topology of the PLA may differ from that of the algebraic curve, but it has the topology of an infinitesimal perturbation- and desingularized – of this one.
5. Geometrically the PLA is close to the algebraic curve: it is close with a precision of $\mu$.
6. The PLA is piecewise linear: one can resort to exact rational arithmetic, to ensure reliability and consistency of the built boundary representation (for example lazy rational arithmetic).
7. The PLA is a priori enough for the needs of CAD/CAM.
8. The calculation of the PLA can be made perfectly reliable.

Conceptually, the M1 method is thus sufficient to advance in a reliable way along the curve. However, if $\mu$ is chosen small to approach closely the curve, M1 becomes terribly slow, even in areas where the curve is very close to a line. Thus, the presented method M2 remains interesting: it makes possible to advance quickly in the calm areas. It is only in presence of a singularity or of quasi singularity, when $R$ is not significantly larger than $\mu$ that one resorts to the M1 method.

## 7. Taking into account rounding errors

Up to now it was supposed that $a$ was exactly on the curve, i. e. $f(a) = 0$, $M = s'(a)^{-1}$. In fact with the numerical inaccuracy inherent to floating –point arithmetic, $a$ is only very close to the curve, and $M$ is only close to $s'(a)^{-1}$. Moreover the floating values calculated for $f(a)$, $s(a)$, $s'(a)$ and so on, are approximations. There are three possible strategies (S1, S2, or S3).

**S1:** This casual approach is unaware of the problem completely and uses only the usual floating-point arithmetic, hoping that all will occur well. It is probable because the contractivity condition imposed for $QN$ is strong. In this strategy, the terms ideally null either are not calculated (which avoids a possible contradiction between practice and theory), or are calculated but regarded as null because they are too small, in short are ignored. In this strategy, the singularities or quasi singularities on the curve are detected only when $R$ is not sufficiently tall in front of $\mu$.

**S2:** The paranoiac approach considers that $a$, $t$, $M$, $s(a)$ and $s(x)$, $s'(a)$, $s'(x)$ are intervals, which contain the exact value. The theory of the reliable preceding curve tracing is not modified. This approach is simple and systematic; calculation by intervals guarantees the results to 100%; certainly the terms ideally null generally cease being it, and there is no problem when they remain sufficiently small, and make it possible to control the effect of the rounding errors. If ever they become too large (for example $0 = r = \|\sigma\|$ that makes it possible to detect simply a singularity or quasi singularity on the curve. Disadvantage: to ensure that the interval of $a$ contains the good value, one needs an additional test, which ensures us that an interval for $a$ contains one zero (simple) of an algebraic system: there already exist such tests, which are provided by Krawckwicz–Moore operators or Segupta–Hansel operators. Another disadvantage is that the IA over-estimates their width largely, so that the convergent iterative methods converge much more slowly, or even diverge. It is possible to find functions $f$ such that $f(x)$ converges for any $x$ in some domain $D$, but $f(D)$ diverges when it is calculated by intervals (subdividing $D$ is a solution). It may be the same when calculating $QN_k(x^{(i)})$. One can think that this problem will not arise, but a rigorous argument would be preferable.

**S3:** To consider that $a$, $M$, $t$ are specific values (i. e. non-intervals), and are approximations of the correct values, and to modify consequently the theory of the curve tracing: knowing a point $a$ near to the curve, to calculate how much one can advance in a reliable way along the curve. This approach is certainly most painful, the least generalizable, the least systematic, but it has the advantage of employing iteration $QN$ on points and not on intervals, and of avoiding the suspicion. The following approach is developed:

The point $a$ is not exactly on the curve. $M$ is a matrix not exactly equal to $s'(a)^{-1}$ is not exactly such that $tf'(a) = 0$. $\|t\| = 1$ remains true nevertheless ($\|t\|$ is the max norm). CC (3) is written now $\|QN'(a+\sigma)\| \leq \frac{1}{2}$. $\|QN'(a+\sigma)\| = Id - s'(a+\sigma)M$ (where $M \approx s'(a)^{-1}$) $= M^{-1}M - s'(a+\sigma)M = (s'(a+\sigma)+E)M - s'(a+\sigma)M$ (where $M^{-1} = s'(a+\sigma)+E$) $= (s'(a)-s'(x)+E)M = KM+EM$ where $K$ is the usual matrix and $EM = Id - s'(a)M$. $\|QN'(a+\sigma)\| = \|KM+EM\| \leq \|KM\| + \|EM\| = \|KM\| + \|Id - s'(a)M\|$.

However, $\|Id - s'(a)M\|$ is indeed upper bounded by a floating value up($\|Id - s'(a)M\|$) (where "up" is the upper bound) calculating $Id - s'(a)M$ with naive IA. To guarantee $\|QN'(a+\sigma)\| \leq 1/2$, $\|KM+EM\| \leq \frac{1}{2}$ is thus enough: If $R$ is such that $\|\sigma\| \leq R \Rightarrow \|KM\| \leq \frac{1}{2}$ then to take account of the rounding errors, it is enough to take $\|\sigma\| \leq R' = R - \text{up}(\|Id - s'(a)M\|)$. Of course $\|Id - s'(a)M\|$ is generally negligible.

**Note:** it would be also possible to get an upper bound of $\|KM + Id - s'(a)M\|$ directly (with IA) and without using an upper bound of $\|KM + Id - s'(a)M\| \leq \|KM\| + \|Id - s'(a)M\|$.

Let us treat now the SC. By being unaware of the rounding errors, one saw that SC is enough to ensure $QN(B) \subset B$ and is equivalent to $k \leq \frac{R}{2}$. One takes again the reasoning in the no ideal case. It is shown at first that $QN(B) \subset B$ always ensures SC by replacing $R$ by $R'$.

Suppose that $x \in B = \text{ball}(a, R')$ and $\|QN'(B)\| \leq \dfrac{1}{2}$ and $\|QN_k(a) - a\| \leq \dfrac{R'}{2}$. Then

$$\big\|QN(x) - a\big\| \leq \big\|QN_k(x) - QN_k(a)\big\| + \big\|QN_k(a) - a\big\| \leq \frac{1}{2}\|x - a\| + \frac{R'}{2};$$

$$\big\|QN_k(x) - a\big\| \leq \frac{R'}{2} + \frac{R'}{2} = R' \Rightarrow x \in B.$$

We want $\|QN_k(a) - a\| \leq \dfrac{R'}{2}$. Let $H$ be the first $n-1$ rows of $M$, $V$ the last row. Then

$$QN_k(a) = a - s(a)M = a - (f(a),\ -ktt^t)\begin{pmatrix} H \\ V \end{pmatrix} = a - f(a)H + ktt^tV;$$

$$\big\|QN_k(a) - a\big\| = \big\|-f(a)H + (ktt^t)V\big\| \leq \big\|f(a)H\big\| + k\|tt^tV\|,$$

$$\big\|QN_k(a) - a\big\| \leq \frac{R'}{2} \Leftarrow k \leq \frac{R'/2 - \|f(a)H\|}{\|tt^tV\|}.$$

We can effectively upper bound $\|tt^tV\|$ with $\text{up}(\|tt^tV\|)$ and $\|f(a)H\|$ with $\text{up}(\|f(a)H\|)$ by calculating these expressions using IA. We obtain finally the following reliable value for $k$: $k \leq \dfrac{R'/2 - \text{up}(\|f(a)H\|)}{\text{up}(\|tt^tV\|)}$. In the ideal case ($f(a) = 0$, $ts'(a) = 0$, $M = s'(a)^{-1}$) the term $\|f(a)H\|$ is zero and $\|tt^tV\|$ is equal to 1.

Really,

$$Id = s'(a)M = (f'(a),\ t^t)\begin{pmatrix} H \\ V \end{pmatrix} = f'(a)H + t^tV.$$

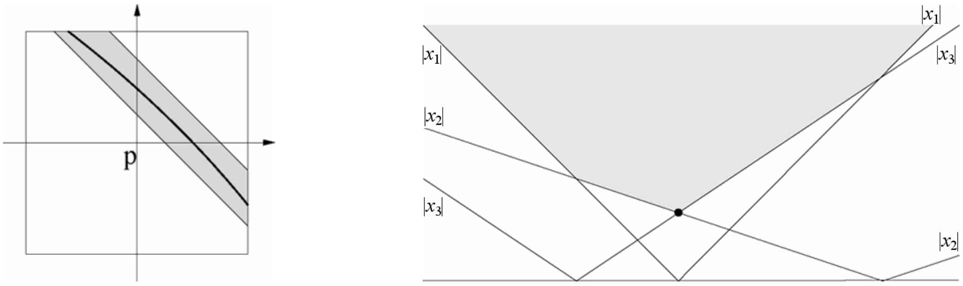
Otherwise $tf'(a) = 0$. Then $tId = tf'(a)H + tt^tV$ and $\|t\| = 1$. Then $\|tt^tV\| = 1$[2].

Obviously, it is interesting that the floating values for $a, t$ and $M$ are as close as possible to the ideal values. If the approximations are too bad, one obtains very small values of $k$, or even negative values, which means that we cannot advance in a reliable way any more, and that it is necessary to resort to the M1 method.

## 8. Distance from a point to the curve

A related question is as follows. Let $p$ be a point. We want to evaluate the distance between $p$ and the curve. A solution resorts to affine arithmetic of intervals. Let us suppose that we know a radius $r$ of a hypercube around $p$ (a good value for $r$ can be found by dichotomy as we saw before). We calculate $f(p + r\varepsilon)$ where $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)$ by affine arithmetic intervals and we obtain $n-1$ linear equations $0 = f(p + r\varepsilon) = \varepsilon A + \Delta$ where $A$ is a $n \times (n+1)$ matrix whose elements are sharp intervals and $\Delta$ a vector of $n$ thick intervals. $\varepsilon A + \Delta = 0$ is the equation of a thick line, the thickness being given by the width of the intervals in the vector $\Delta$. The vector $\varepsilon$ with smallest Euclidean norm satisfying $\varepsilon A + \Delta = 0$ is $-\Delta(A^tA)^{-1}A^t$. The interval point on the thick line and closest to $p$ is thus $p = -\Delta(A^tA)^{-1}A^t$.

[2] Note: $V = 1/(tt^t)$.



Left: An arc of curve, and a containing thick line. The center of the square is $p$ and the half side $r$. The affine IA makes it possible to calculate the thick line. Right: Finding the point on the line $x = (q_1 + \lambda\mu_1, q_2 + \lambda\mu_2, q_3 + \lambda\mu_3)$ which is of minimal norm. The lines represent $|x_1|$, $|x_2|$, $|x_3|$

**8.1. Example:** $n = 3$, $f(x, y, z) = (x^2 + y^2 - 4,\ z)$, $p = (1.4, 1.4, 0)$, $r = 0.1$. Then

$$f(p + r\varepsilon) = (\varepsilon_1, \varepsilon_2, \varepsilon_3) \begin{pmatrix} 0.28 & 0 \\ 0.28 & 0 \\ 0 & 0.1 \end{pmatrix} + \big([-0.08, -0.06],\ [0, 0]\big) = (0, 0). \quad (10)$$

The point $p + \varepsilon$ closest to $p$ (with Euclidean distance) is given by $(\varepsilon_1, \varepsilon_2, \varepsilon_3) = -\Delta(A^tA)^{-1}A^t = ([-0.08, -0.06],\ [0, 0])(AA^t)^{-1}A^t$.

Skipping tedious computations, the point of the curve nearest to $p$ (in Euclidean distance) is in the box $([1.4071, 1.4071],\ [1.4071, 1.4071],\ [0, 0])$ which indeed contains the exact solution $(\sqrt{2}, \sqrt{2}, 0)$. These intervals prove that $p$ is distant by more than 0.01515 from the curve but less than 0.02021. Here the method could obtain one lower bound (not null) distance to the curve. In the general case, the obtained lower bound can be null. It suffices that $f(p)$, calculated by intervals, contains zero, or in an equivalent way that $p$ is inside the thick line. The preceding method extends naturally to calculation of the distance between one point and a surface, a hypersurface, and so on, and of the distance between a point $p$ and another simple point zero of an algebraic system.

Which is the distance between $p$ and the curve, for the max norm? The thick line is expressed in the form $x = q + \lambda\mu$ where $q$ is a thick point, $\lambda \in \boldsymbol{R}$ is the parameter, along the line, and $\mu$ the directing thick line vector. The sought distance is then $d = \min \|x\| = \min(\max(|q_1 + \lambda\mu_1|, \ldots, |q_n + \lambda\mu_n|))$ where $x = q + \lambda\mu$ is the only unknown. Each $|q_i + \lambda\mu_i|$ gives two lines $y - (q_i + \lambda\mu_i) = 0$ and $y + (q_i + \lambda\mu_i) = 0$ in the plane $(\lambda, y)$. The sought value of $d$ is the coordinate $y$ of the lowest point (i. e. of minimal $y$ coordinate) of the convex described by $y - (q_i + \lambda\mu_i) \geq 0$ and $y + (q_i + \lambda\mu_i) \geq 0$. It is a linear programming problem in 2d. There is a traditional algorithm in $O(n)$ to find this point. The thickness of the point $q$ is easily taken into account: one considers initially $\min(|q_i|)$, then $\max(|q_i|)$ to find one lower and upper bounds of $d$.

This method is generalizable to the distance (always according to the norm max) between one point and one surface (it is still a linear programming problem), or a hypersurface.

## Conclusion

This paper has proposed a reliable prediction correction method for curve tracing. Several ways of computing a safe step parameter were presented, and compared on simple examples. A related question, computing a range of the distance between a point and a curve/surface/hyper surface, was also treated.

Finally, the methods presented here are compatible with all kinds of IA: the naive one, the centered one (to compute a range for $f\left(x_c \pm \frac{\omega}{2}\right)$, the latter computes $f(x_c)$ and $f'\left(x_c \pm \frac{\omega}{2}\right)$with the naive IA), the interval affine arithmetic of Figueiredo and Stolfi [2], or the "Bernstein IA" used by Hu et al. [4], Garlof [5] (and others). The presented method detects when the current points approaches a singularity (or a almost quasi singularity, i.e. a singularity up to the finite accuracy of the computer). In such a case, one may resort to another method, for instance PLA.

## References

[1] Dedieu J.P., Yakoubsohn J.C. Two seminumerical algorithms for solving polynomial systems / Technical report. – Labo "Approximation et Optimisation", Univ. Paul Sabatier, Toulouse, France, 1994.

[2] de Figueiredo L.H., Stolfi J. Adaptive enumeration of implicit surfaces with affine arithmetic // Computer Graphics Forum. – 1996. – Vol. 15(5). – P. 287–296.

[3] Faux I.D., Pratt J.M. Computational geometry for design and manufacture. – Chichester: Ellis Horwood, 1979.

[4] Chun-Yi Hu, Takashi Maekawa, Patrikalakis N.M., Xiuzi Ye. Robust interval algorithm for surface intersections // Computer-aided Design. – 1997. – Vol. 29(9). – P. 617–627.

[5] Garloff J., Graf B. Solving strict polynomial inequalities by Bernstein expansion // The Use of Symbolic Methods in Control System Analysis and Design / N. Munro, Ed. – London: The Institution of Electrical Engineers (IEE), 1999. – P. 339–352.

[6] Baker R. Kearfott Rigorous Global Search: Continuous Problems. – Dordrecht: Kluwer Academic Publisher, 1996.

[7] Luo R.C., Ma Y., Mac D.F. Allister Tracing tangential surface-surface intersections // Symposium on Solid Modeling Foundations and CAD/CAM Applications. – 1995. – P. 255–262.

[8] Neumaier A. Interval Methods for Systems of Equations Encyclopedia of Mathematics and its Applications. 37. – Cambridge: Cambridge Univ. Press, 1990.

[9] Taubin G. An accurate algorithm for rasterizing algebraic curves // Proc. of the Second Symposium on Solid Modeling and Applications (SMA'93). – 1993. – P. 221–230.