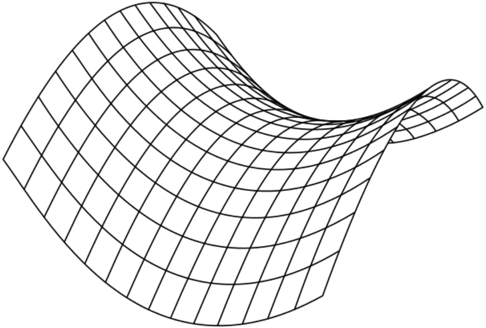
Fast surface visualization

Nuwan Herath
Supervisors: Sylvain Lazard, Marc Pouget, Guillaume Moroz
GAMBLE

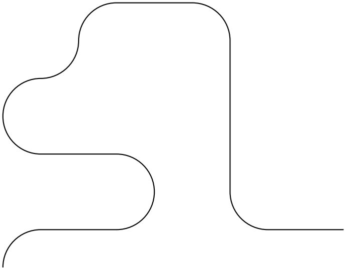
16/06/2021

Context

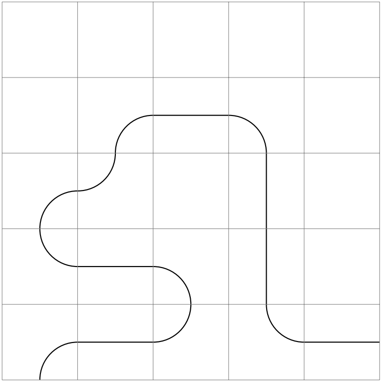
Discrete representation of a surface



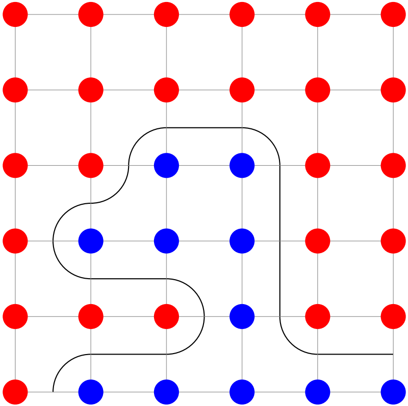
Implicit function: $f(x, y) = 0$



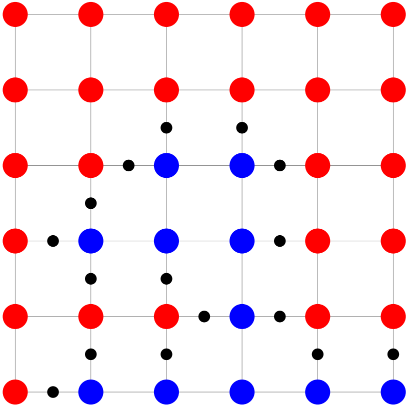
Implicit function: $f(x, y) = 0$



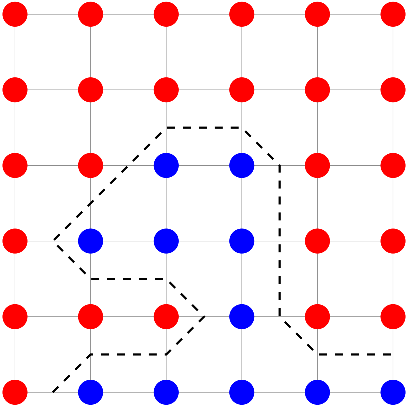
Implicit function: $f(x, y) = 0$



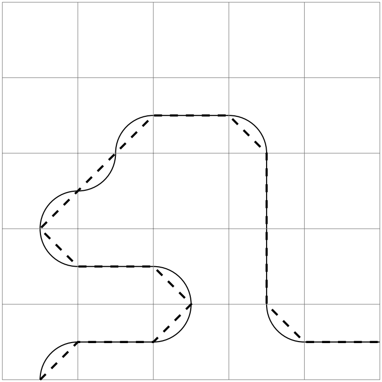
Implicit function: $f(x, y) = 0$



Implicit function: $f(x, y) = 0$

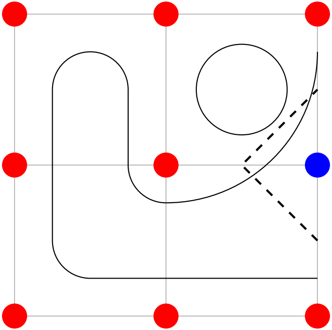


Implicit function: $f(x, y) = 0$



Marching cubes

An issue



Direct evaluation of the nodes

Complexity (number of elementary operations):

$$\Theta(Nd^2) + \Theta(N \cdot \log_2(N) \cdot d)$$

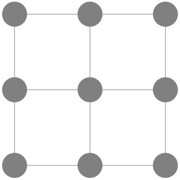
N resolution of the grid
d partial degree

If $\log_2(N) < d$: $\Theta(Nd^2)$
For instance $\log_2(10000) \simeq 13 < 100$.

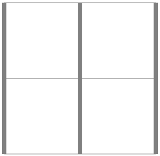
Slow for high degree polynomials...

Our approach

~~Evaluation on intersections of the grid~~

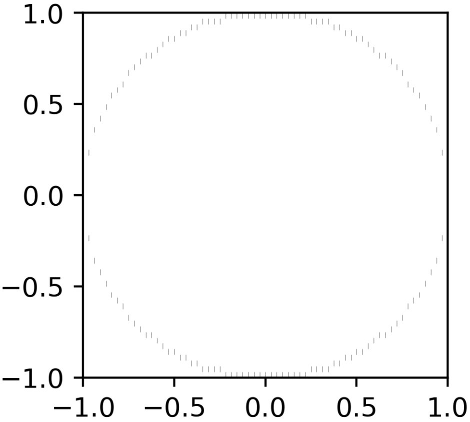


Evaluation along fibers



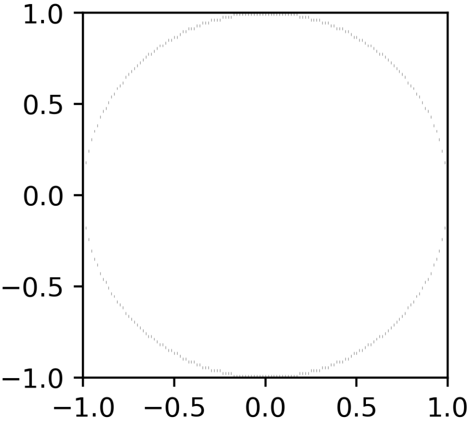
$\Rightarrow$ Making it fast and “certified”

Examples



Resolution $N = 64$

Examples



Resolution $N = 128$

Definition

The Chebyshev polynomials T_n verify

$$T_n(\cos\theta) = \cos(n\theta).$$

The first three Chebyshev polynomials

$$\cos(0 \cdot \theta) = 1 \qquad T_0 = 1$$

$$\cos(1 \cdot \theta) = \cos(\theta) \qquad T_1 = X$$

$$\cos(2 \cdot \theta) = 2\cos(\theta)^2 - 1 \qquad T_2 = 2X^2 - 1$$

Definition

The Chebyshev polynomials T_n verify

$$T_n(\cos\theta) = \cos(n\theta).$$

Lemma

An arbitrary polynomial p of degree N can be written in terms of the Chebyshev polynomials:

$$p(x) = \sum_{i=0}^{N} a_i T_n(x).$$

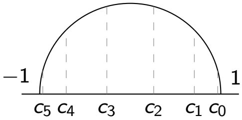
Definition

For $N \in \mathbb{N}$, the Chebyshev nodes are

$$c_k = \cos\left(\frac{2k+1}{2N}\pi\right),\ k = 0, \ldots, N-1.$$

They are the roots of T_N.

For $N = 6$



DFT / DCT

Discrete Fourier Tranform (DFT): $x_n \to X_k$

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk}$$

Discrete Cosine Transform (DCT-II): $x_n \to X_k$

$$X_k = \sum_{n=0}^{N-1} x_n \ \cos\left[\frac{\pi(2n+1)k}{2N}\right]$$

$\Rightarrow$ Fast thanks to the FFT algorithm ~~$\Theta(N^2)$~~ $\Theta(N \log_2 N)$

Multipoint evaluation with the IDCT

Inverse Discrete Cosine Transform (IDCT): $X_k \to x_n$

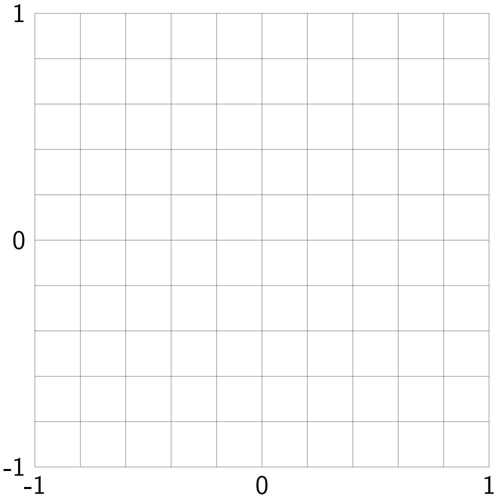
$$x_n = -\frac{1}{2}X_0 + \sum_{k=0}^{N-1} X_k \ \cos\left[\frac{\pi k(2n+1)}{2N}\right]$$

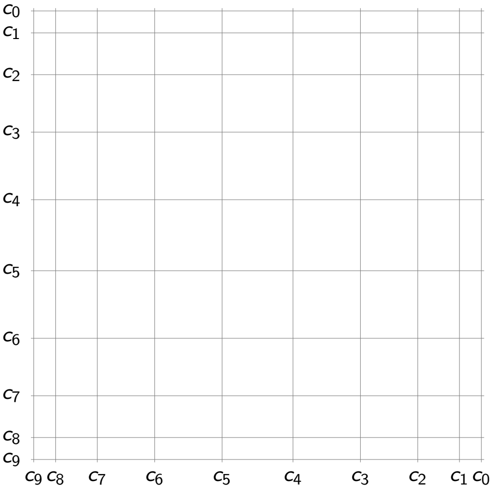
$$p(c_n) = \sum_{k=0}^{N-1} a_k \cos\left[\frac{\pi k(2n+1)}{2N}\right]$$

$$(p(c_n))_n = idct((a_k)_k) + \frac{1}{2}(a_0, \ldots, a_0)$$

Algorithm: full multipoint

Illustration

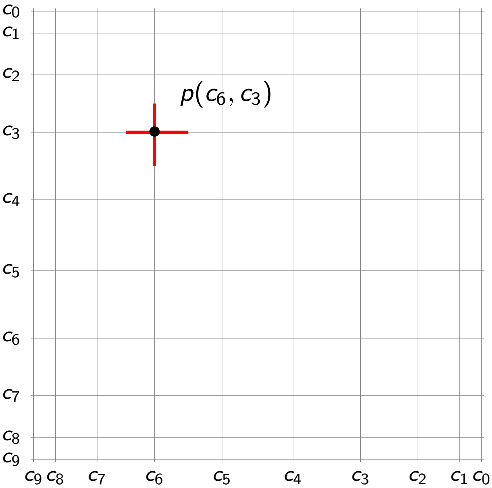




c_0
c_1
c_2
c_3
c_4
c_5
c_6
c_7
c_8
c_9
c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0

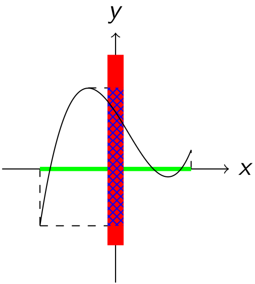
Algorithm: full multipoint

Illustration



F is an interval extension of f and verifies

$$F(I) \supseteq f(I).$$



Taylor approximation

For $r \in [-R, R]$ and $m \in \mathbb{N}$,

$$p^c_{approx}(r) = f(c) + rf'(c) + \cdots + \frac{r^m}{m!} f^{(m)}(c)$$

$$|p(c + r) - p^c_{approx}(r)| \leq B$$

where

$$m \ll degree(p)$$

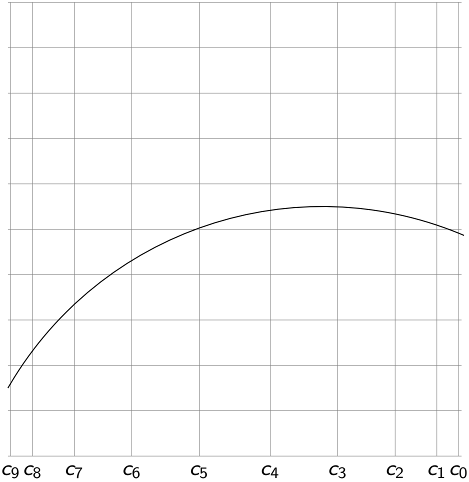
$\Rightarrow$ Speed and certification

Without multipoint evaluation: $\Theta(Nd^2)$
With multipoint evaluation:

$$\Theta(N^2 m \log_2 d) \text{ if } \frac{d}{\log_2 d} < m \frac{N}{\log_2 N}$$

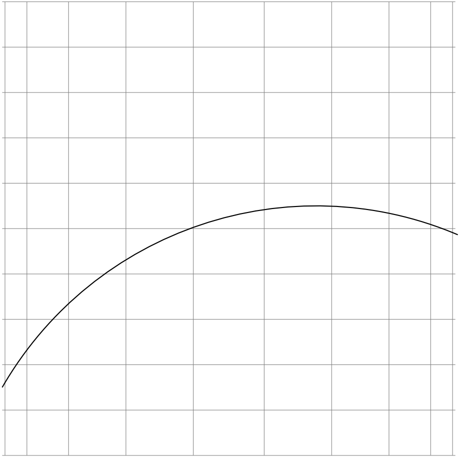
Algorithm: partial multipoint

Illustration



Algorithm: partial multipoint

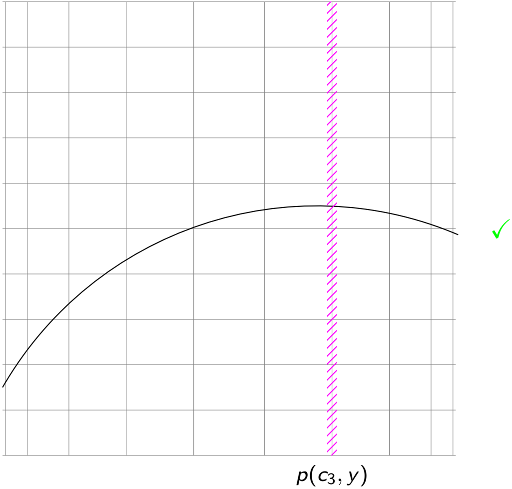
Illustration



$p(c_3, y)$

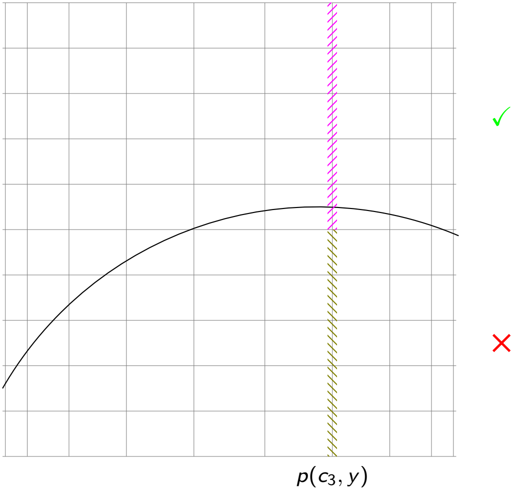
Algorithm: partial multipoint

Illustration



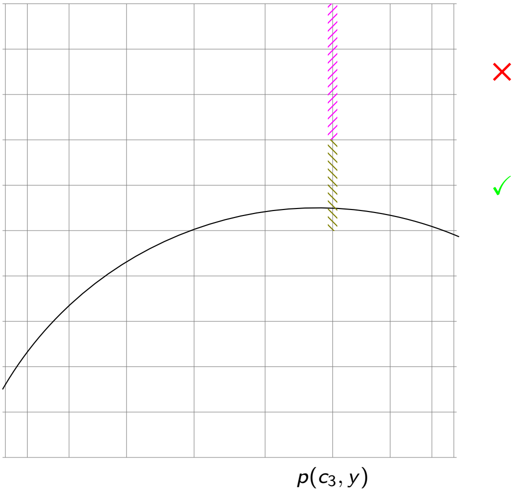
Algorithm: partial multipoint

Illustration



Algorithm: partial multipoint

Illustration



Algorithm: partial multipoint

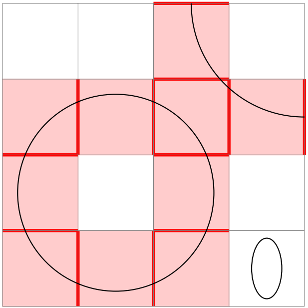
Complexity

With multipoint evaluation: $\Theta(N^2 m \log_2 d)$

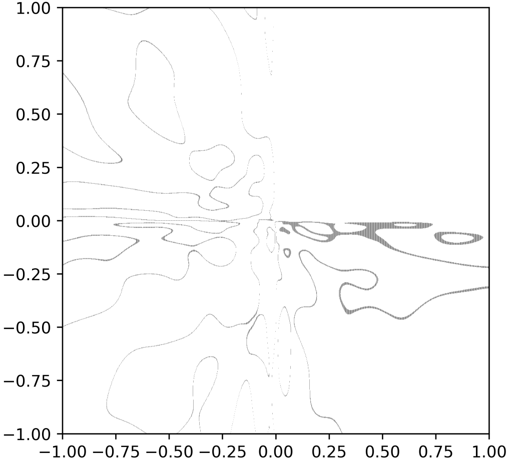
With pruning: $O(dN\underset{\sim}{\log_2 d})$ expected

Current guarantees

The algorithms return an enclosure of the curves along the fibers.



Example



Resolution $N = 1024$

Conclusion

- It is not possible for now to say which method is better.
- The computational time is promising (a few seconds for a polynomial of partial degree 100 and $N = 1000$)
- 2D $\rightarrow$ 3D