SIBGRAPI '17
30th Conference on Graphics, Patterns and Images

Interval methods
for computer graphics and geometric modeling

Luiz Henrique de Figueiredo

impa
Instituto de Matemática Pura e Aplicada

Motivation

Basic problems in computer graphics and geometric modeling typically reduce to solving systems of nonlinear equations:

$$\begin{aligned} f_1(x_1, \ldots, x_n) &= 0 \\ &\cdots \\ f_m(x_1, \ldots, x_n) &= 0 \end{aligned}$$

Motivation

Basic problems in computer graphics and geometric modeling typically reduce to solving systems of nonlinear equations:

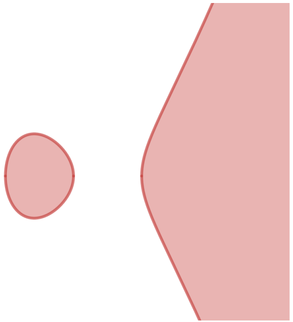
$$f_1(x_1, \ldots, x_n) = 0$$
$$\ldots$$
$$f_m(x_1, \ldots, x_n) = 0$$

$$y^2 = x^3 - x$$

Motivation

Basic problems in computer graphics and geometric modeling typically reduce to solving systems of nonlinear inequalities:

$$f_1(x_1, \ldots, x_n) \geq 0$$
$$\ldots$$
$$f_m(x_1, \ldots, x_n) \geq 0$$



$y^2 \leq x^3 - x$

Motivation – rendering an implicit surface with ray casting

Implicit surface

$$h(x, y, z) = 0, \qquad h\colon \mathbf{R}^3 \to \mathbf{R}$$

Ray

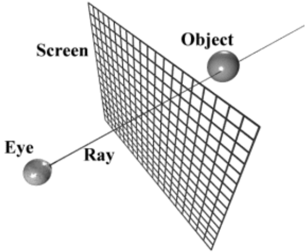
$$r(t) = e + t \cdot v = (x(t), y(t), z(t)), \quad t \in [0, \infty)$$

Ray intersects surface when

$$f(t) = h(r(t)) = 0$$

First intersection occurs at smallest zero of f in $[0, \infty)$

Need all zeros for rendering CSG models



Motivation – rendering an implicit surface with ray casting

Implicit surface

$$h(x, y, z) = 0, \qquad h\colon \mathbf{R}^3 \to \mathbf{R}$$

Ray

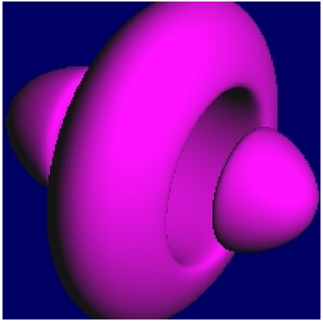
$$r(t) = e + t \cdot v = (x(t), y(t), z(t)), \quad t \in [0, \infty)$$

Ray intersects surface when

$$f(t) = h(r(t)) = 0$$

First intersection occurs at smallest zero of f in $[0, \infty)$

Need all zeros for rendering CSG models



$$4(x^4 + (y^2 + z^2)^2) + 17x^2(y^2 + z^2) - 20(x^2 + y^2 + z^2) + 17 = 0$$

Motivation – plotting an implicit curve

Implicit curve

$$f(x, y) = 0, \qquad f\colon \mathbf{R}^2 \to \mathbf{R}$$

Motivation – plotting an implicit curve

Implicit curve

$$f(x,y) = 0, \qquad f\colon \mathbf{R}^2 \to \mathbf{R}$$

$$\begin{gathered}0.004 + 0.110x - 0.177y - 0.174x^2 + 0.224xy - 0.303y^2 \\ -0.168x^3 + 0.327x^2y - 0.087xy^2 - 0.013y^3 + 0.235x^4 \\ -0.667x^3y + 0.745x^2y^2 - 0.029xy^3 + 0.072y^4 = 0\end{gathered}$$

Motivation – plotting an implicit curve

Implicit curve

$$f(x, y) = 0, \qquad f\colon \mathbf{R}^2 \to \mathbf{R}$$

$$\begin{gathered} 0.004 + 0.110x - 0.177y - 0.174x^2 + 0.224xy - 0.303y^2 \\ -0.168x^3 + 0.327x^2y - 0.087xy^2 - 0.013y^3 + 0.235x^4 \\ -0.667x^3y + 0.745x^2y^2 - 0.029xy^3 + 0.072y^4 = 0 \end{gathered}$$

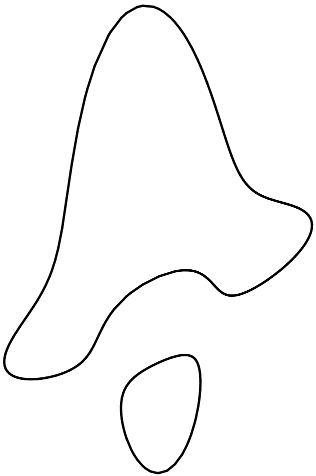
Descartes

Motivation – plotting an implicit curve

Implicit curve

$$f(x,y)=0, \qquad f\colon \mathbf{R}^2 \to \mathbf{R}$$

$$\begin{gathered}0.004+0.110x-0.177y-0.174x^2+0.224xy-0.303y^2\\ -0.168x^3+0.327x^2y-0.087xy^2-0.013y^3+0.235x^4\\ -0.667x^3y+0.745x^2y^2-0.029xy^3+0.072y^4=0\end{gathered}$$



Motivation – intersecting two parametric surfaces

Parametric surfaces

$$g_1\colon \Omega_1 \subset \mathbf{R}^2 \to \mathbf{R}^3$$

$$g_2\colon \Omega_2 \subset \mathbf{R}^2 \to \mathbf{R}^3$$

Intersection

$$g_1(u_1, v_1) - g_2(u_2, v_2) = 0$$

Motivation – intersecting two parametric surfaces

Parametric surfaces

$$g_1 \colon \Omega_1 \subset \mathbf{R}^2 \to \mathbf{R}^3$$

$$g_2 \colon \Omega_2 \subset \mathbf{R}^2 \to \mathbf{R}^3$$

Intersection

$$g_1(u_1, v_1) - g_2(u_2, v_2) = 0$$

$$x_1(u_1, v_1) - x_2(u_2, v_2) = 0$$

$$y_1(u_1, v_1) - y_2(u_2, v_2) = 0$$

$$z_1(u_1, v_1) - z_2(u_2, v_2) = 0$$

Motivation – intersecting two parametric surfaces

Parametric surfaces

$$g_1 : \Omega_1 \subset \mathbf{R}^2 \rightarrow \mathbf{R}^3$$

$$g_2 : \Omega_2 \subset \mathbf{R}^2 \rightarrow \mathbf{R}^3$$

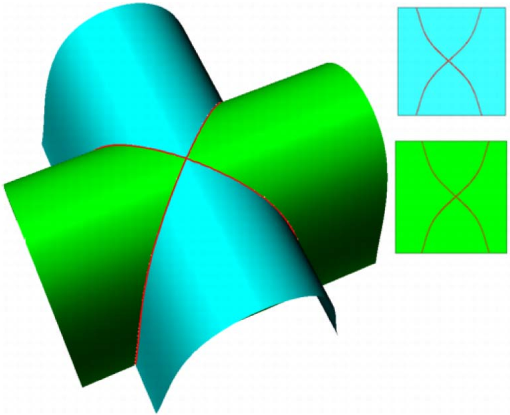
Intersection

$$g_1(u_1, v_1) - g_2(u_2, v_2) = 0$$

$$x_1(u_1, v_1) - x_2(u_2, v_2) = 0$$

$$y_1(u_1, v_1) - y_2(u_2, v_2) = 0$$

$$z_1(u_1, v_1) - z_2(u_2, v_2) = 0$$



Hongwei Lin, Yang Qin, Hongwei Liao, Yunyang Xiong
"Affine Arithmetic-Based B-Spline Surface Intersection with GPU Acceleration"
IEEE TVCG, 2014

Motivation

Basic problems in computer graphics and geometric modeling typically reduce to solving systems of nonlinear equations:

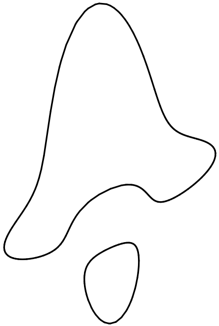
$$
\begin{aligned}
f_1(x_1, \ldots, x_n) &= 0 \\
&\ldots \\
f_m(x_1, \ldots, x_n) &= 0
\end{aligned}
$$

Motivation

Basic problems in computer graphics and geometric modeling typically reduce to solving systems of nonlinear equations:

$$f_1(x_1, \ldots, x_n) = 0$$
$$\ldots$$
$$f_m(x_1, \ldots, x_n) = 0$$

Low-dimensional solutions



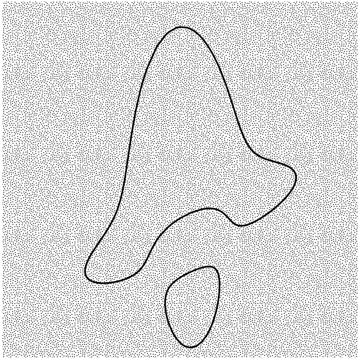
Motivation

Basic problems in computer graphics and geometric modeling typically reduce to solving systems of nonlinear equations:

$$f_1(x_1, \ldots, x_n) = 0$$
$$\ldots$$
$$f_m(x_1, \ldots, x_n) = 0$$

Low-dimensional solutions

$\implies$ sampling costly and unreliable



Motivation

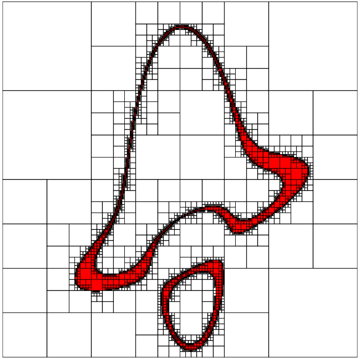
Basic problems in computer graphics and geometric modeling typically reduce to solving systems of nonlinear equations:

$$f_1(x_1, \ldots, x_n) = 0$$
$$\ldots$$
$$f_m(x_1, \ldots, x_n) = 0$$

Low-dimensional solutions

$\Longrightarrow$ sampling costly and unreliable

Interval methods provide robust adaptive solutions

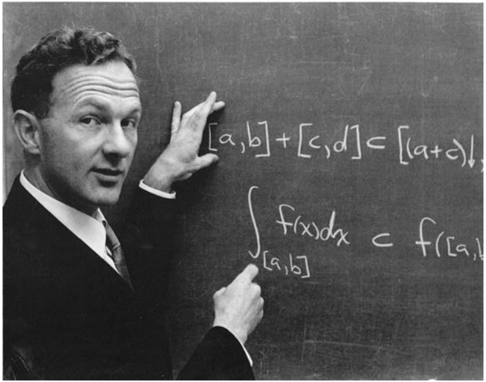


interval arithmetic

Interval arithmetic

Moore (1960)

Introduced to improve reliability of numerical computations through automated a posteriori error analysis of both rounding errors in floating-point arithmetic and measurement errors in input data

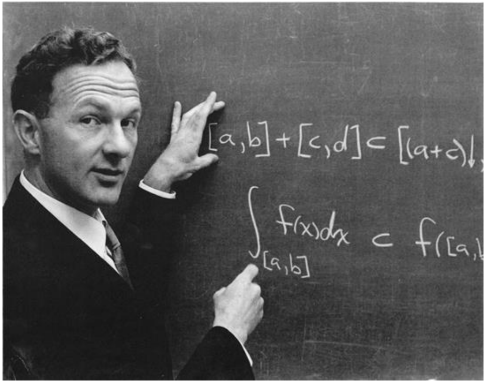


Ramon E. Moore (1929–2015)

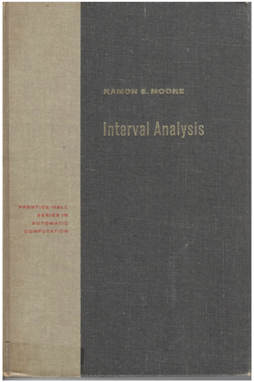
Interval arithmetic

Moore (1960)

Introduced to improve reliability of numerical computations through automated a posteriori error analysis of both rounding errors in floating-point arithmetic and measurement errors in input data



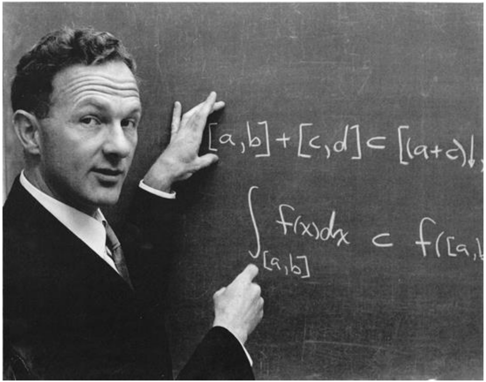
Ramon E. Moore (1929–2015)



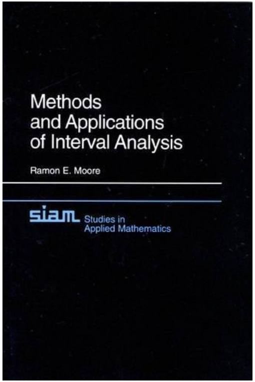
1966

Interval arithmetic

Introduced to improve reliability of numerical computations through automated a posteriori error analysis of both rounding errors in floating-point arithmetic and measurement errors in input data



Ramon E. Moore (1929–2015)

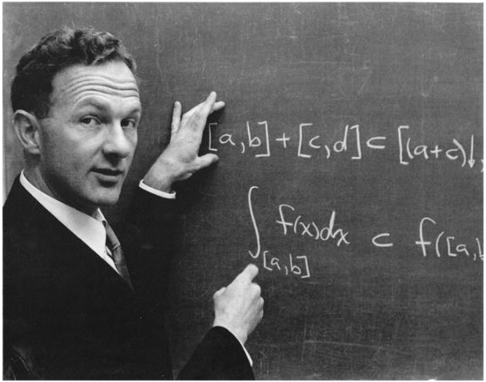


1979

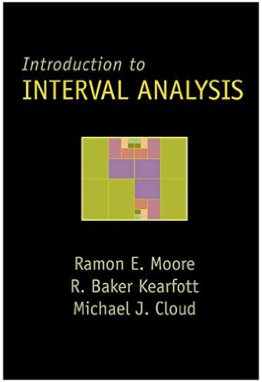
Interval arithmetic

Moore (1960)

Introduced to improve reliability of numerical computations through automated a posteriori error analysis of both rounding errors in floating-point arithmetic and measurement errors in input data



Ramon E. Moore (1929–2015)



2009

Interval arithmetic in computer graphics and geometric modeling

Can probe the global behavior of mathematical functions

Provides reliable bounds for the values of a function over whole regions of its domain

Avoids costly and unreliable point sampling

Leads naturally to adaptive algorithms

Both micro and macro scales

Interval arithmetic in computer graphics and geometric modeling

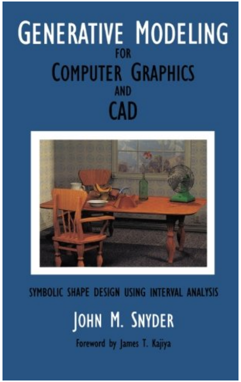
Can probe the global behavior of mathematical functions

Provides reliable bounds for the values of a function over whole regions of its domain

Avoids costly and unreliable point sampling

Leads naturally to adaptive algorithms

Both micro and macro scales



1992

Interval arithmetic in computer graphics and geometric modeling

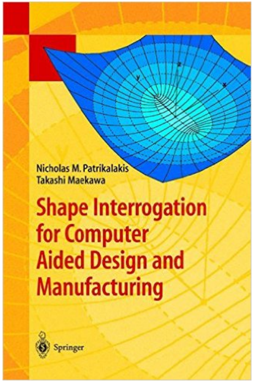
Can probe the global behavior of mathematical functions

Provides reliable bounds for the values of a function over whole regions of its domain

Avoids costly and unreliable point sampling

Leads naturally to adaptive algorithms

Both micro and macro scales



2002

Interval arithmetic in computer graphics and geometric modeling

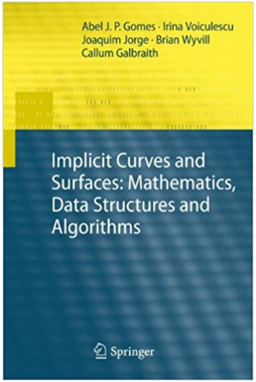
Can probe the global behavior of mathematical functions

Provides reliable bounds for the values of a function over whole regions of its domain

Avoids costly and unreliable point sampling

Leads naturally to adaptive algorithms

Both micro and macro scales



2009

Interval arithmetic

Represent quantities as intervals

$$x \sim [a, b] \implies x \in [a, b]$$

Operate with intervals generating other intervals

$$\begin{aligned}
[a, b] + [c, d] &= [a + c, b + d] \\
[a, b] \times [c, d] &= [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)] \\
[a, b] \,/\, [c, d] &= [a, b] \times [1/d, 1/c] \\
[a, b]^2 &= [\min(a^2, b^2), \max(a^2, b^2)] \quad \text{if } 0 \notin [a, b] \\
[a, b]^2 &= [0, \max(a^2, b^2)] \quad \text{if } 0 \in [a, b] \\
\exp\,[a, b] &= [\exp(a), \exp(b)]
\end{aligned}$$

Automatic extensions for complicated expressions with operator overloading

Interval arithmetic

Every expression f has an interval extension F :

$$x_i \in X_i \implies f(x_1, \ldots, x_n) \in F(X_1, \ldots, X_n)$$

Reliable range estimates without point sampling

$$F(X) \supseteq f(X) = \{f(x) : x \in X\}$$

In particular:

$$\begin{aligned} 0 \notin F(X) \quad &\implies \quad 0 \notin f(X) \\ &\implies \quad f(x) = 0 \text{ has no solution in } X \end{aligned}$$

Interval arithmetic

Every expression f has an interval extension F :

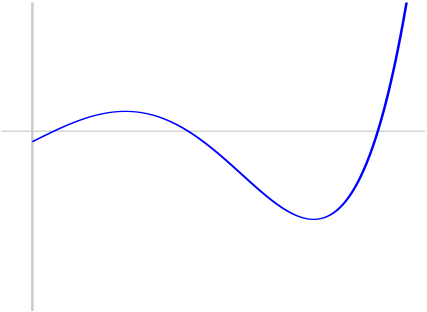
$$x_i \in X_i \implies f(x_1, \ldots, x_n) \in F(X_1, \ldots, X_n)$$

Reliable range estimates without point sampling

$$F(X) \supseteq f(X) = \{f(x) : x \in X\}$$

In particular:

$$\begin{aligned} 0 \notin F(X) \quad &\implies \quad 0 \notin f(X) \\ &\implies \quad f(x) = 0 \text{ has no solution in } X \end{aligned}$$



f

Interval arithmetic

Every expression f has an interval extension F :

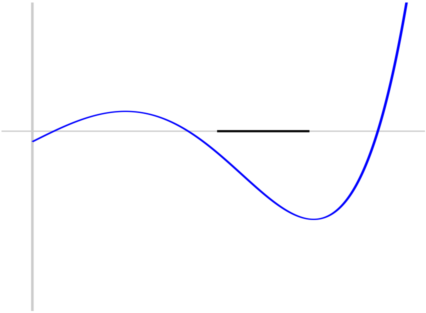
$$x_i \in X_i \implies f(x_1, \ldots, x_n) \in F(X_1, \ldots, X_n)$$

Reliable range estimates without point sampling

$$F(X) \supseteq f(X) = \{f(x) : x \in X\}$$

In particular:

$$\begin{aligned} 0 \notin F(X) \quad &\implies \quad 0 \notin f(X) \\ &\implies \quad f(x) = 0 \text{ has no solution in } X \end{aligned}$$



X

Interval arithmetic

Every expression f has an interval extension F :

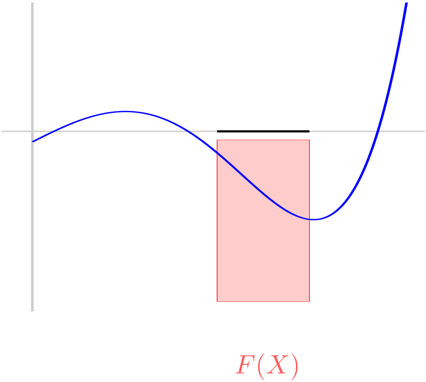
$$x_i \in X_i \implies f(x_1, \ldots, x_n) \in F(X_1, \ldots, X_n)$$

Reliable range estimates without point sampling

$$F(X) \supseteq f(X) = \{f(x) : x \in X\}$$

In particular:

$$0 \notin F(X) \implies 0 \notin f(X)$$
$$\implies f(x) = 0 \text{ has no solution in } X$$



Interval arithmetic

Every expression f has an interval extension F :

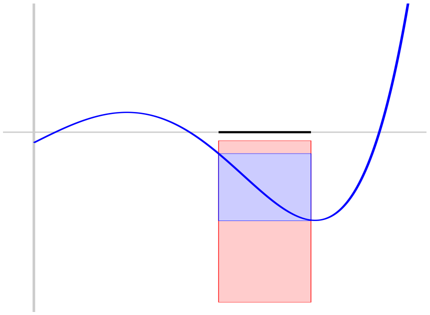
$$x_i \in X_i \implies f(x_1, \ldots, x_n) \in F(X_1, \ldots, X_n)$$

Reliable range estimates without point sampling

$$F(X) \supseteq f(X) = \{f(x) : x \in X\}$$

In particular, even if $F(X) \supsetneq f(X)$:

$$\begin{aligned} 0 \notin F(X) \quad &\implies \quad 0 \notin f(X) \\ &\implies \quad f(x) = 0 \text{ has no solution in } X \end{aligned}$$



$F(X) \supsetneq f(X)$

Interval arithmetic

Every expression f has an interval extension F :

$$x_i \in X_i \implies f(x_1, \ldots, x_n) \in F(X_1, \ldots, X_n)$$

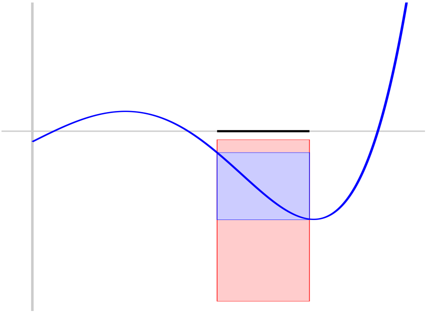
Reliable range estimates without point sampling

$$F(X) \supseteq f(X) = \{f(x) : x \in X\}$$

In particular, even if $F(X) \supsetneq f(X)$:

$$\begin{aligned} 0 \notin F(X) \quad &\implies \quad 0 \notin f(X) \\ &\implies \quad f(x) = 0 \text{ has no solution in } X \end{aligned}$$

This is a computational proof



$F(X) \supsetneq f(X)$

Interval arithmetic

Given a system of nonlinear equations

$$f_1(x_1, \ldots, x_n) = 0$$
$$\ldots$$
$$f_m(x_1, \ldots, x_n) = 0$$

and interval extensions

$$F_1, \ldots, F_m$$

there are no solutions in a box $X = X_1 \times \cdots \times X_n \subseteq \mathbf{R}^n$ if

$$0 \notin F_k(X) \quad \text{for some } k$$

There may be solutions in X if

$$0 \in F_k(X) \quad \text{for all } k$$

Interval arithmetic

Given a system of nonlinear inequalities

$$\begin{gathered} f_1(x_1, \ldots, x_n) \geq 0 \\ \cdots \\ f_m(x_1, \ldots, x_n) \geq 0 \end{gathered}$$

and interval extensions

$$F_1, \ldots, F_m$$

there are no solutions in a box $X = X_1 \times \cdots \times X_n \subseteq \mathbf{R}^n$ if

$$\max F_k(X) < 0 \quad \text{for some } k$$

There may be solutions in X if

$$\max F_k(X) \geq 0 \quad \text{for all } k$$

Interval probing of implicit curve

$$y^2 - x^3 + x = 0$$

$$\begin{aligned} X &= [-2, -1] \\ Y &= [1, 2] \end{aligned}$$

Interval probing of implicit curve

$$y^2 - x^3 + x = 0$$

$$\begin{aligned} X &= [-2, -1] \\ Y &= [1, 2] \\ X^3 &= [-8, -1] \\ -X^3 &= [1, 8] \\ -X^3 + X &= [-1, 7] \end{aligned}$$

Interval probing of implicit curve

$$y^2 - x^3 + x = 0$$

$$\begin{aligned} X &= [-2, -1] \\ Y &= [1, 2] \\ X^3 &= [-8, -1] \\ -X^3 &= [1, 8] \\ -X^3 + X &= [-1, 7] \\ Y^2 &= [1, 4] \\ Y^2 - X^3 + X &= [0, 11] \end{aligned}$$

Interval probing of implicit curve

$$y^2 - x^3 + x = 0$$

$$\begin{aligned}
X &= [-2, -1] \\
Y &= [1, 2] \\
X^3 &= [-8, -1] \\
-X^3 &= [1, 8] \\
-X^3 + X &= [-1, 7] && \text{exact} = [0, 6] \\
Y^2 &= [1, 4] \\
Y^2 - X^3 + X &= [0, 11] && \text{exact} = [1, 10]
\end{aligned}$$

Interval estimates not tight, but improve as intervals shrink

Interval probing of implicit curve

$$y^2 - x^3 + x = 0$$

$$\begin{aligned}
X &= [-2, -1] \\
Y &= [1, 2] \\
X^3 &= [-8, -1] \\
-X^3 &= [1, 8] \\
-X^3 + X &= [-1, 7] \qquad \text{exact} = [0, 6] \\
Y^2 &= [1, 4] \\
Y^2 - X^3 + X &= [0, 11] \qquad \text{exact} = [1, 10]
\end{aligned}$$

Interval estimates not tight, but improve as intervals shrink $\Longrightarrow$ divide-and-conquer

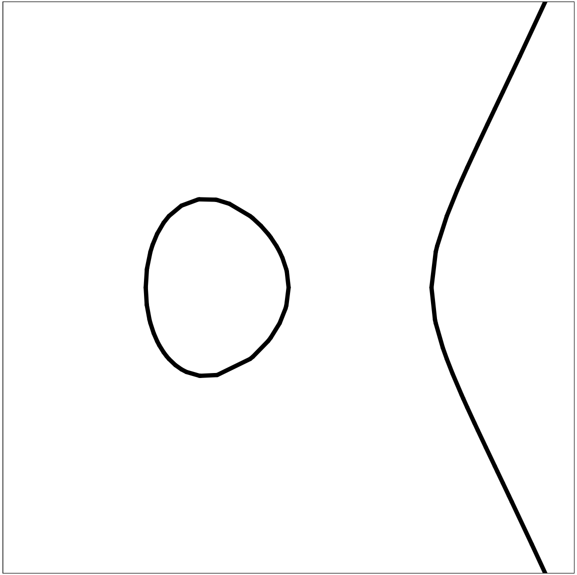
Interval probing of implicit curve

$y^2 - x^3 + x = 0$

$X \times Y = [-2, -1] \times [1, 2]$

$F(X, Y) = [0, 11]$ maybe

$f(X, Y) = [1, 10]$ no



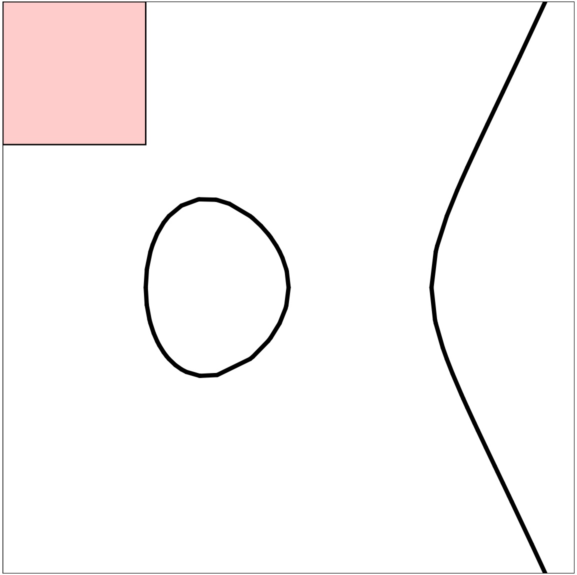
Interval probing of implicit curve

$y^2 - x^3 + x = 0$

$X \times Y = [-2, -1] \times [1, 2]$

$F(X, Y) = [0, 11]$ maybe

$f(X, Y) = [1, 10]$ no



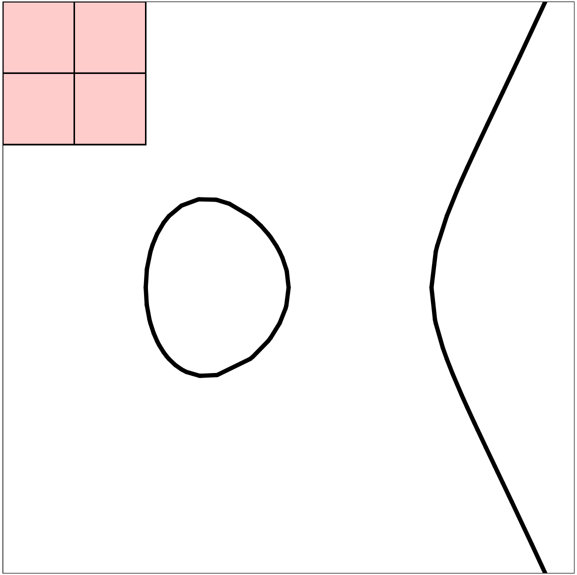
Interval probing of implicit curve

$y^2 - x^3 + x = 0$

$X \times Y = [-2, -1] \times [1, 2]$

$F(X, Y) = [0, 11]$ maybe

$f(X, Y) = [1, 10]$ no

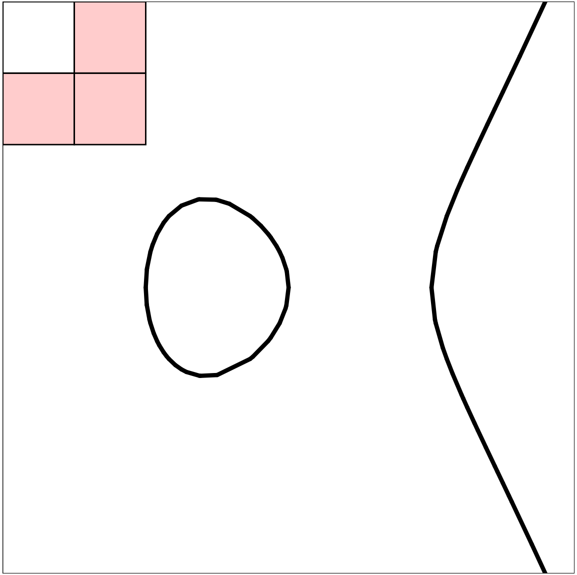


Interval probing of implicit curve

$y^2 - x^3 + x = 0$

$X \times Y = [-2, -1.5] \times [1.5, 2]$

$F(X, Y) = [3.625, 10.5]$ no

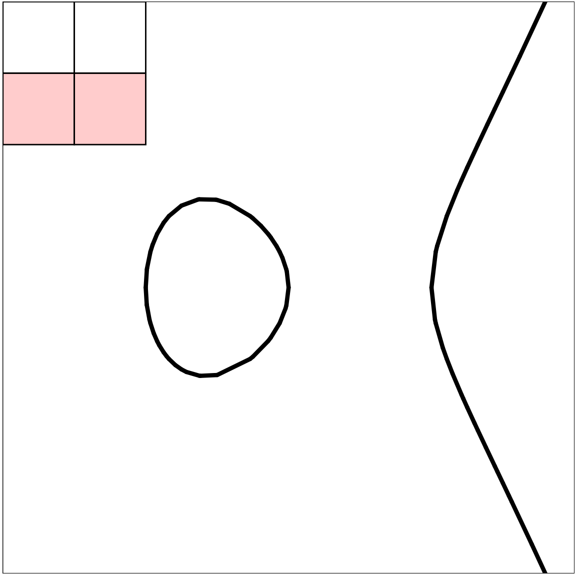


Interval probing of implicit curve

$y^2 - x^3 + x = 0$

$X \times Y = [-1.5, -1] \times [1.5, 2]$

$F(X, Y) = [1.75, 6.375]$ no

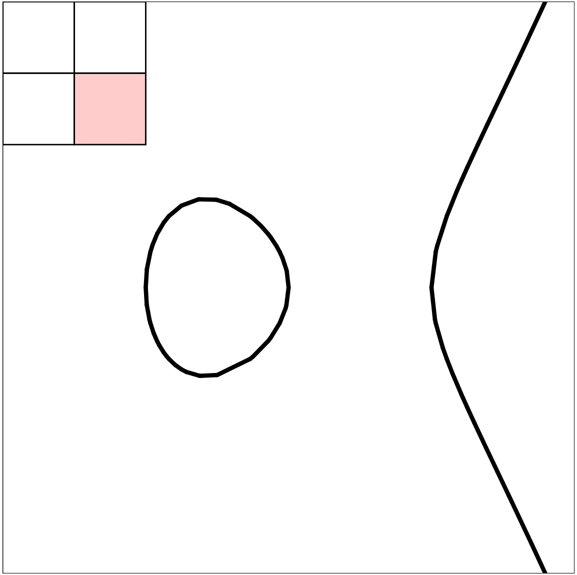


Interval probing of implicit curve

$y^2 - x^3 + x = 0$

$X \times Y = [-2, -1.5] \times [1, 1.5]$

$F(X, Y) = [2.375, 8.75]$ no

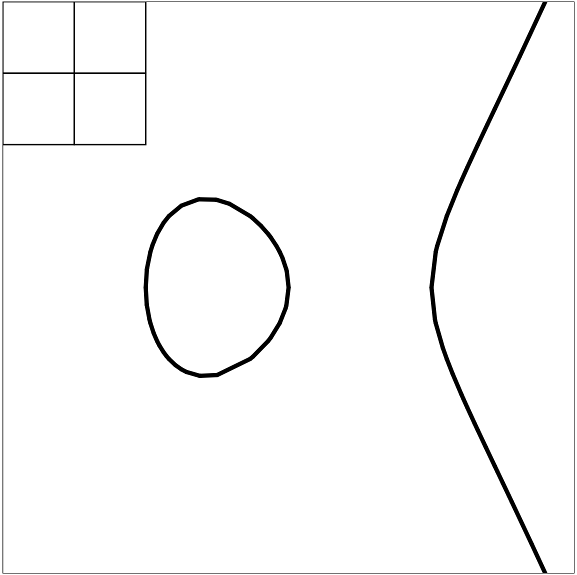


Interval probing of implicit curve

$y^2 - x^3 + x = 0$

$X \times Y = [-1.5, -1] \times [1, 1.5]$

$F(X, Y) = [0.5, 4.625]$ no



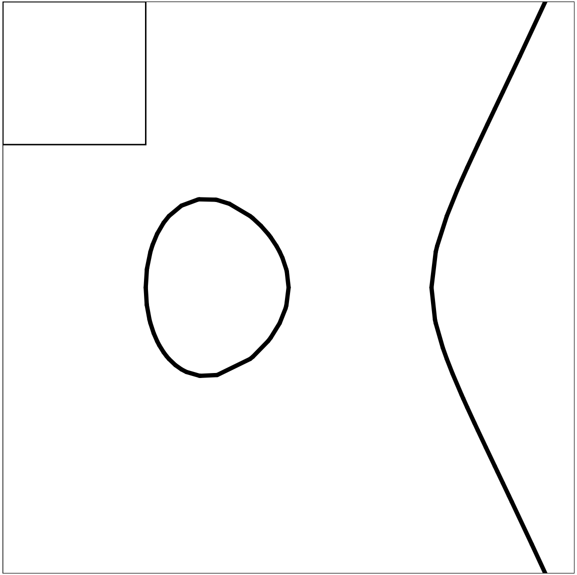
Interval probing of implicit curve

$y^2 - x^3 + x = 0$

$X \times Y = [-2, -1] \times [1, 2]$

$F(X, Y) = [0.5, 10.5]$ no

$f(X, Y) = [1, 10]$ no



Adaptive domain subdivision

To solve $f(x) = 0$ in $\Omega \subseteq \mathbf{R}^n$
call explore(Ω)

procedure explore(X)
  if $0 \notin F(X)$ **then**
    discard X
  elseif small(X) **then**
    output X
  else
    $X_1, \ldots, X_k \leftarrow$ subdivide(X)
    for each i **do** explore(X_i)
  end
end

Suffern–Fackerell (1991), Snyder (1992)

Adaptive domain subdivision

To solve $f(x) = 0$ in $\Omega \subseteq \mathbf{R}^n$
call explore(Ω)

procedure explore(X)
  if $0 \notin F(X)$ **then**
    discard X
  elseif small(X) **then**
    output X
  else
    $X_1, \ldots, X_k \leftarrow$ subdivide(X)
    for each i **do** explore(X_i)
  end
end

Suffern–Fackerell (1991), Snyder (1992)

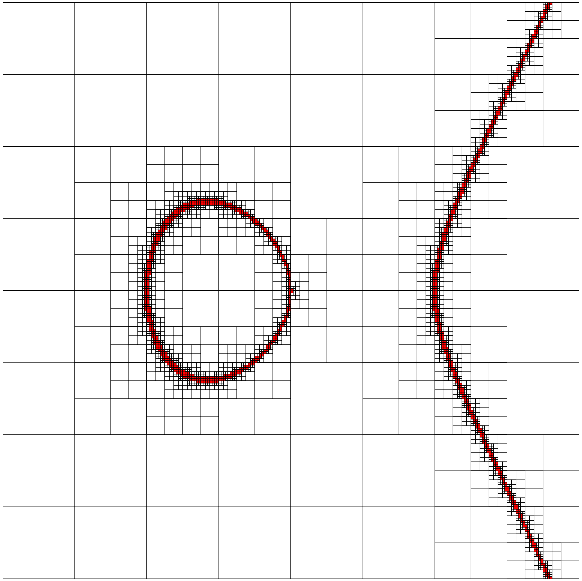
"When you have eliminated the impossible,
whatever remains, however improbable,
must be the truth."
—Sherlock Holmes in *The Sign of Four*

Adaptive domain subdivision

To solve $f(x) = 0$ in $\Omega \subseteq \mathbf{R}^n$
call explore(Ω)

procedure explore(X)
  if $0 \notin F(X)$ **then**
    discard X
  elseif small(X) **then**
    output X
  else
    $X_1, \ldots, X_k \leftarrow$ subdivide(X)
    for each i **do** explore(X_i)
  end
end

Suffern–Fackerell (1991), Snyder (1992)

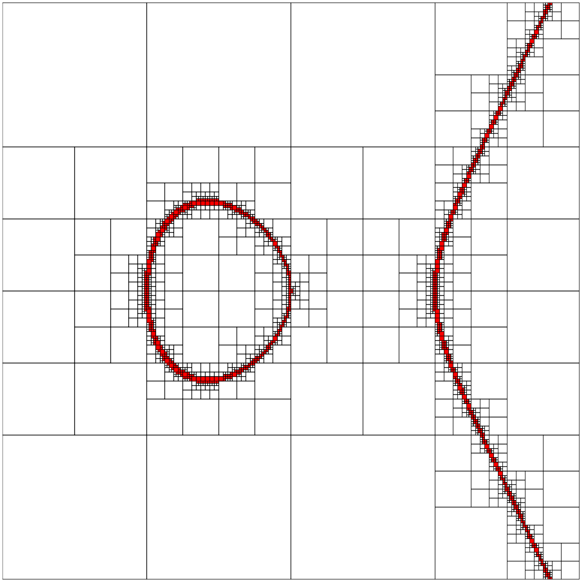


Adaptive domain subdivision

To solve $f(x) = 0$ in $\Omega \subseteq \mathbf{R}^n$
call explore(Ω)

procedure explore(X)
    if $0 \notin F(X)$ **then**
        discard X
    elseif small(X) **then**
        output X
    else
        $X_1, \ldots, X_k \leftarrow$ subdivide(X)
        for each i **do** explore(X_i)
    end
end

Suffern–Fackerell (1991), Snyder (1992)

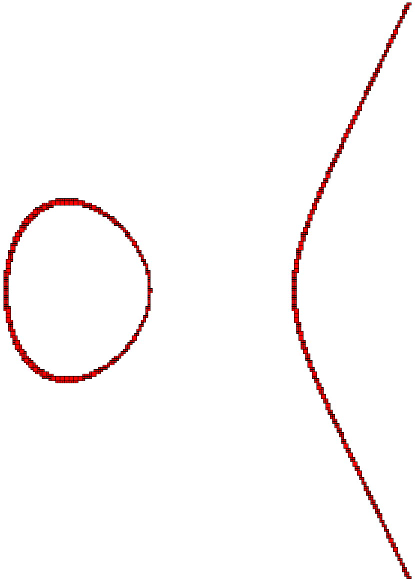


Adaptive domain subdivision

To solve $f(x) = 0$ in $\Omega \subseteq \mathbf{R}^n$
call explore(Ω)

procedure explore(X)
  if $0 \notin F(X)$ **then**
    discard X
  elseif small(X) **then**
    output X
  else
    $X_1, \ldots, X_k \leftarrow$ subdivide(X)
    for each i **do** explore(X_i)
  end
end

Suffern–Fackerell (1991), Snyder (1992)

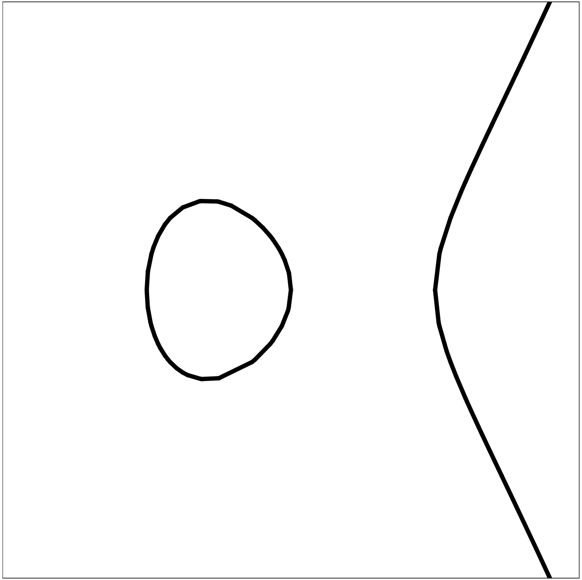


Adaptive domain subdivision

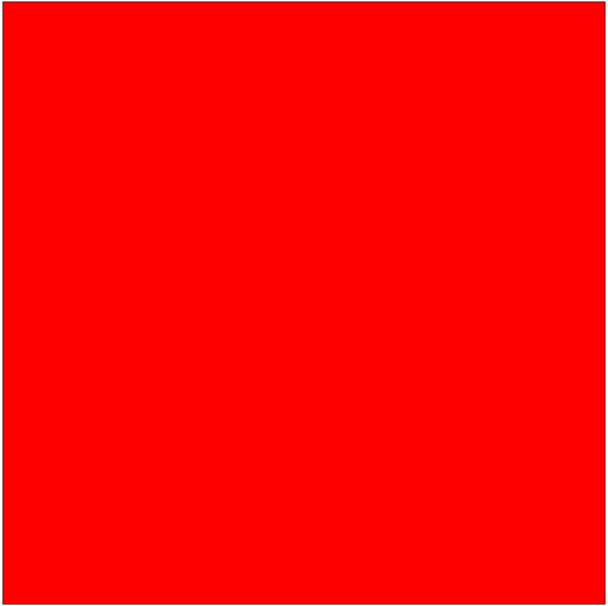
To solve $f(x) = 0$ in $\Omega \subseteq \mathbf{R}^n$
call explore(Ω)

procedure explore(X)
  if $0 \notin F(X)$ **then**
    discard X
  elseif small(X) **then**
    output X
  else
    $X_1, \ldots, X_k \leftarrow$ subdivide(X)
    for each i **do** explore(X_i)
  end
end

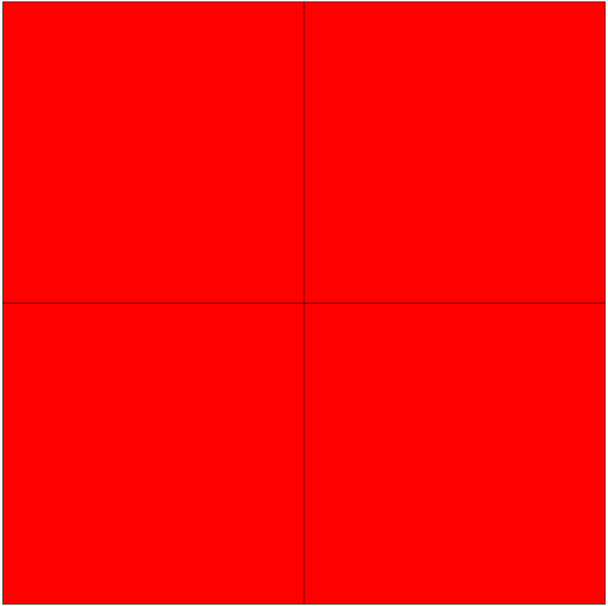
Suffern–Fackerell (1991), Snyder (1992)



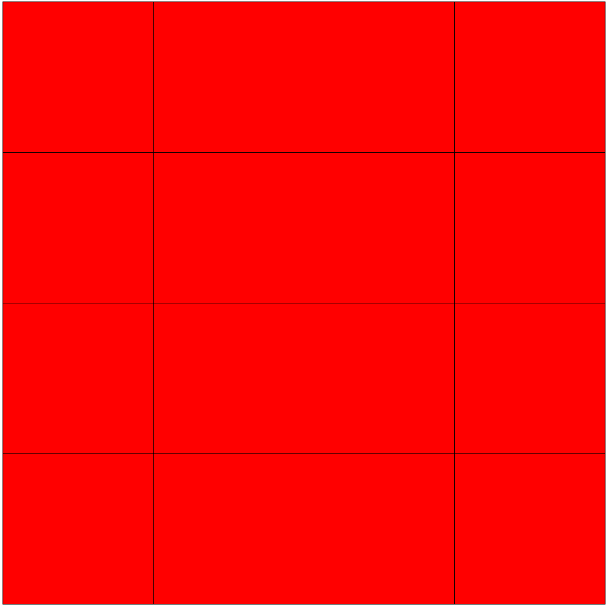
Implicit curves



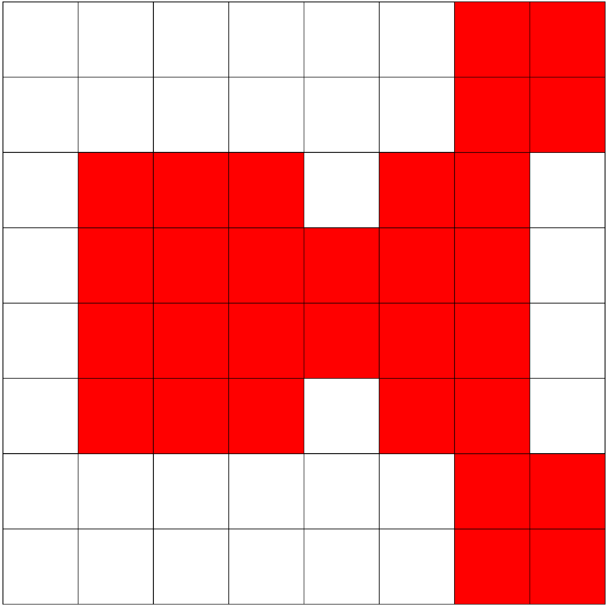
Implicit curves



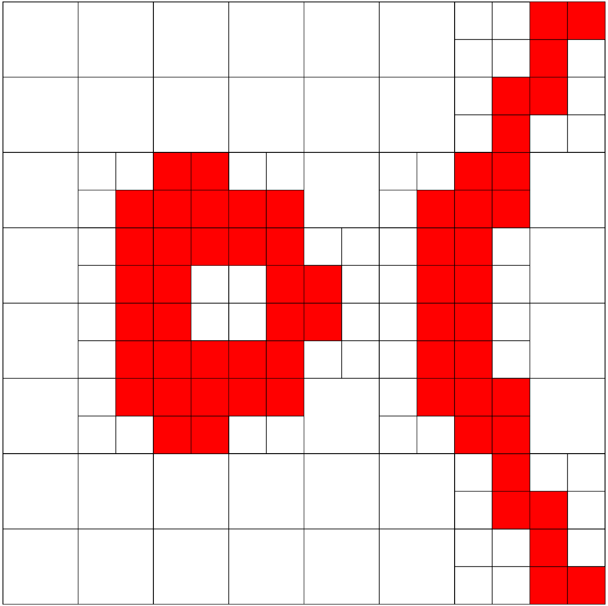
Implicit curves



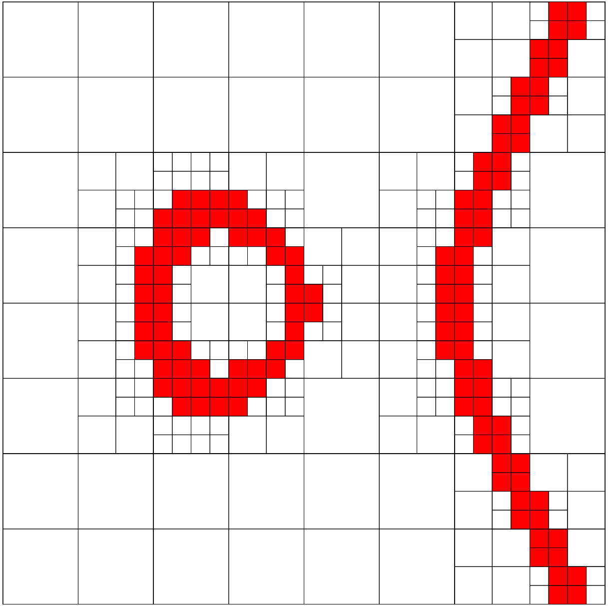
Implicit curves



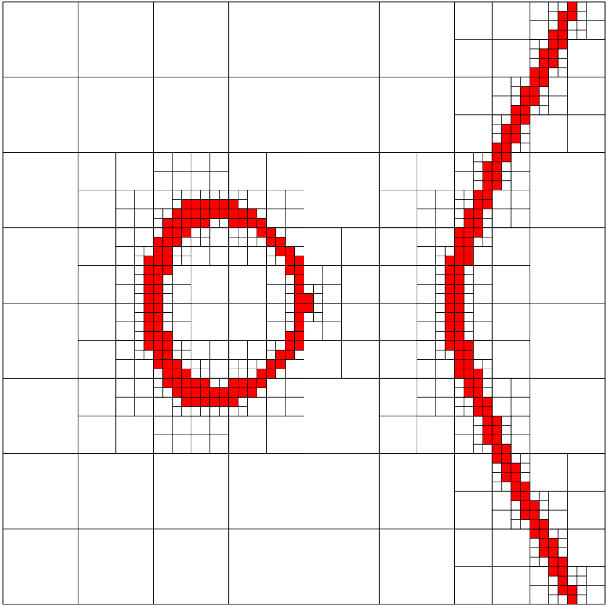
Implicit curves



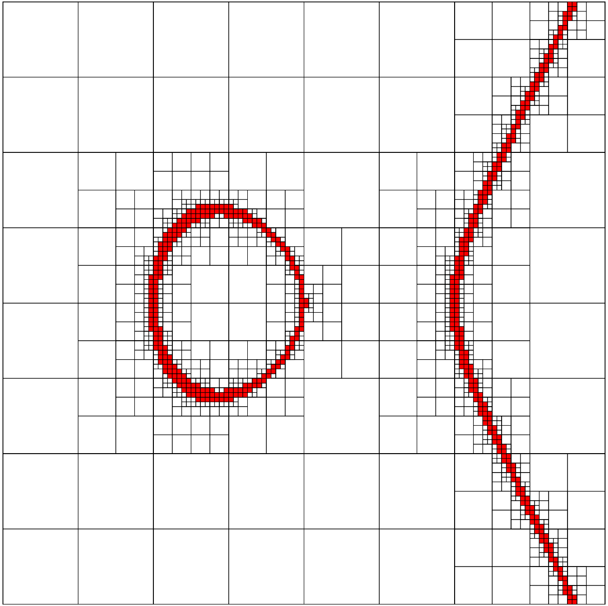
Implicit curves



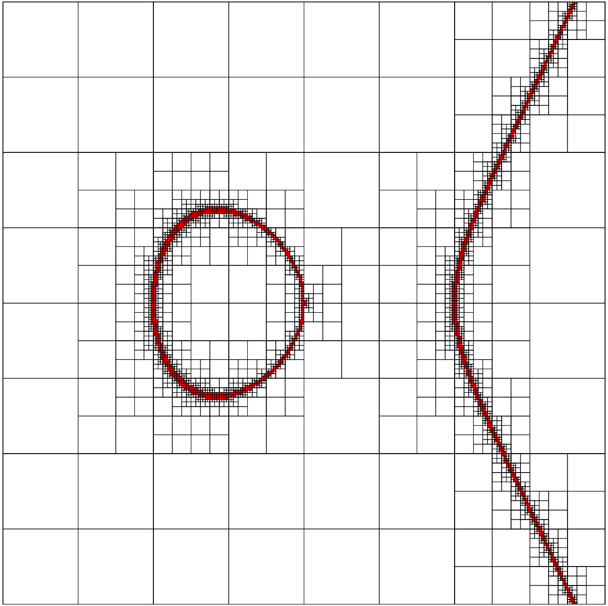
Implicit curves



Implicit curves



Implicit curves



Implicit curves

F inclusion function for f

procedure explore(X)
  if $0 \notin F(X)$ **then**
    discard X
  elseif small(X) **then**
    output X
  else
    $X_1, \ldots, X_k \leftarrow$ subdivide(X)
    for each i **do** explore(X_i)
  end
end

spatial adaption

Suffern–Fackerell (1991), Snyder (1992)

Implicit curves

F inclusion function for f

```
procedure explore(X)
   if 0 ∉ F(X) then
         discard X
   elseif small(X) then
         output X
   else
         X_1, ..., X_k ← subdivide(X)
         for each i do explore(X_i)
   end
end
```

spatial adaption

Suffern–Fackerell (1991), Snyder (1992)

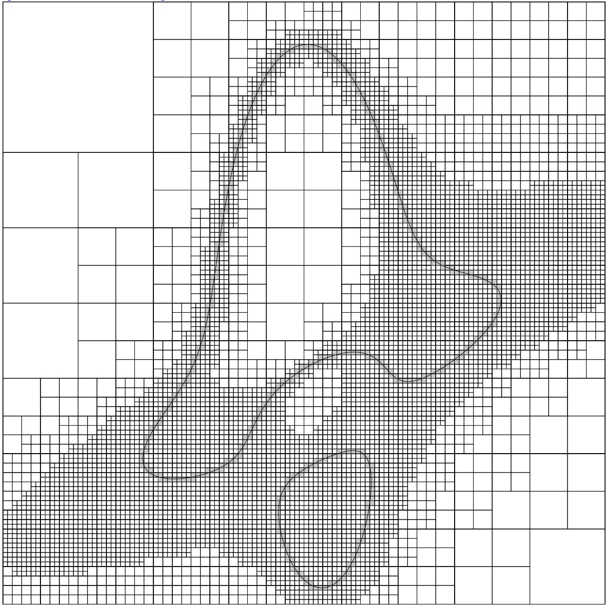
G inclusion function for $\mathrm{grad}\, f$

```
procedure explore(X)
   if 0 ∉ F(X) then
         discard X
   elseif small(X) or small(G(X)) then
         approx(X)
   else
         X_1, ..., X_k ← subdivide(X)
         for each i do explore(X_i)
   end
end
```

geometric adaption

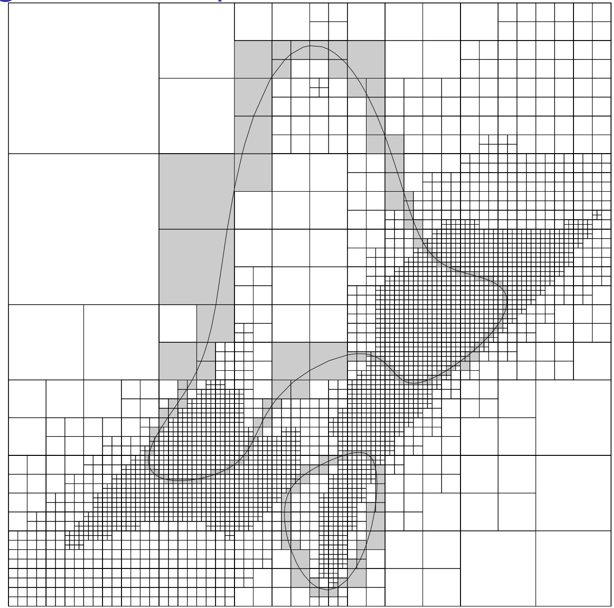
Lopes–Oliveira–Figueiredo (2002)

Implicit curves – spatial adaption



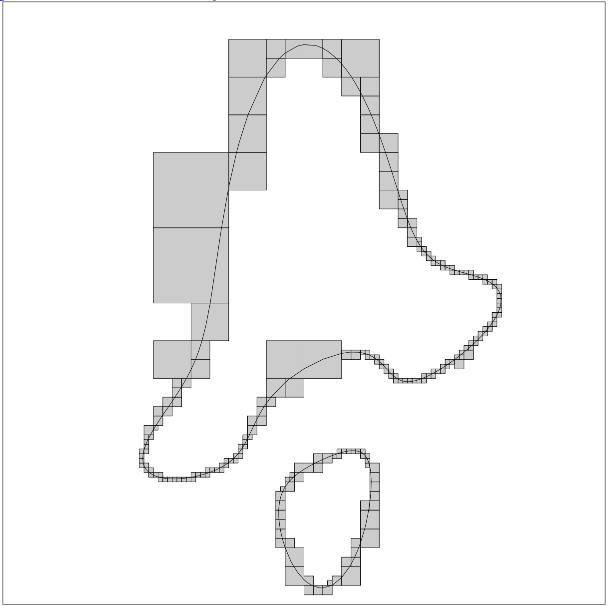
Implicit curves – geometric adaption

Lopes–Oliveira–Figueiredo (2002)



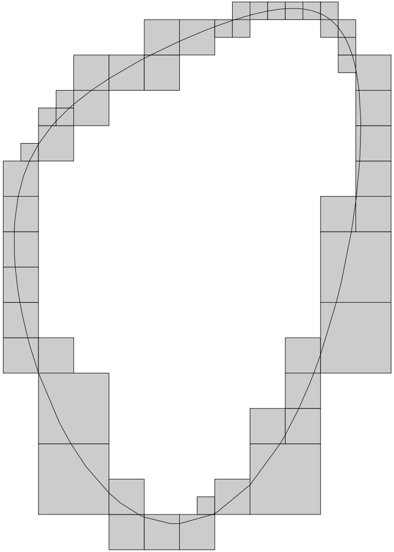
Implicit curves – geometric adaption

Lopes–Oliveira–Figueiredo (2002)



Implicit curves – geometric adaption

Lopes–Oliveira–Figueiredo (2002)



more applications

Implicit regions

Given by systems of nonlinear inequalities

$$\begin{aligned} f_1(x, y) &\geq 0 \\ &\dots \\ f_m(x, y) &\geq 0 \end{aligned}$$

Implicit regions

Given by systems of nonlinear inequalities

$$f_1(x,y) \geq 0$$
$$\cdots$$
$$f_m(x,y) \geq 0$$

procedure explore(X)
  if $\max F(X) < 0$ **then**
    discard X
  elseif small(X) **then**
    output X
  else
    $X_1, \ldots, X_k \leftarrow$ subdivide(X)
    for each i **do** explore(X_i)
  end
end

Implicit manifolds

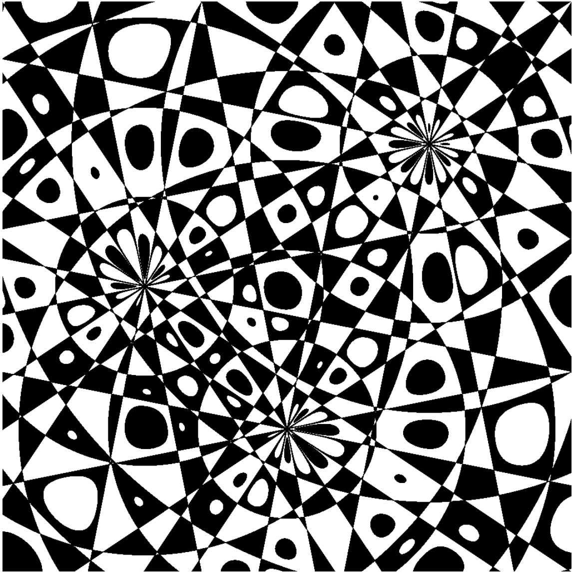
Given by systems of nonlinear equations

$$f_1(x,y) = 0$$
$$\cdots$$
$$f_m(x,y) = 0$$

procedure explore(X)
    if $0 \notin F(X)$ **then**
        discard X
    elseif small(X) **then**
        output X
    else
        $X_1, \ldots, X_k \leftarrow$ subdivide(X)
        for each i **do** explore(X_i)
    end
end

Implicit regions

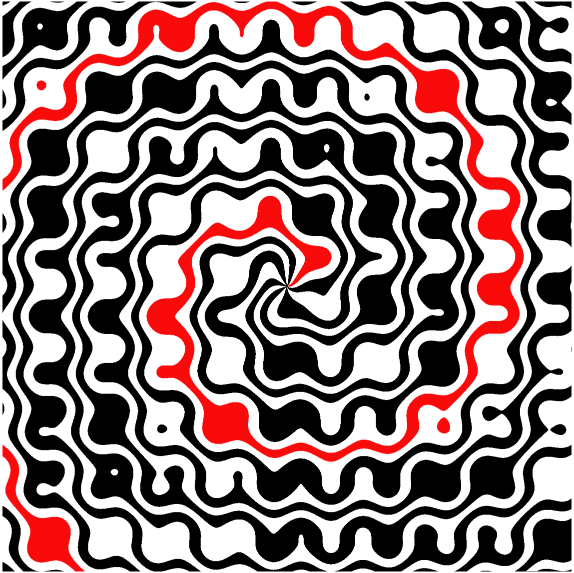
Tupper (2001)



GrafEq

Implicit regions

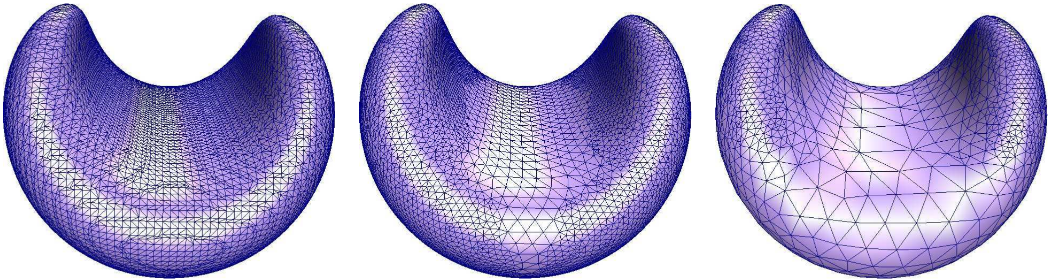
Tupper (2001)



GrafEq

Implicit surfaces

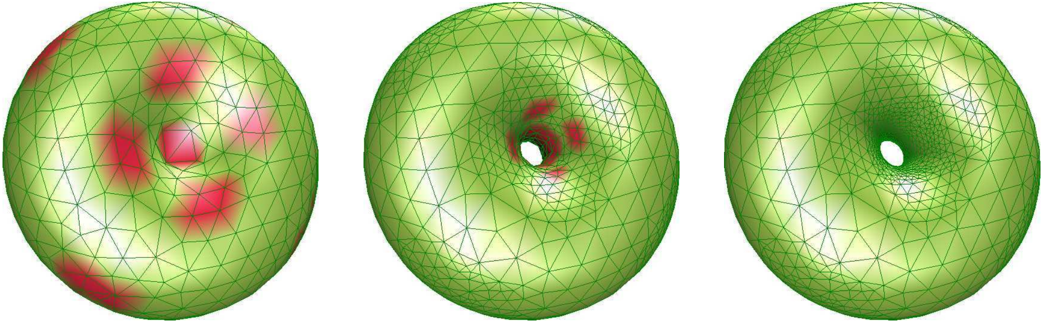
Paiva–Lopes–Lewiner–Figueiredo (2006)



track regions of high curvature

Implicit surfaces

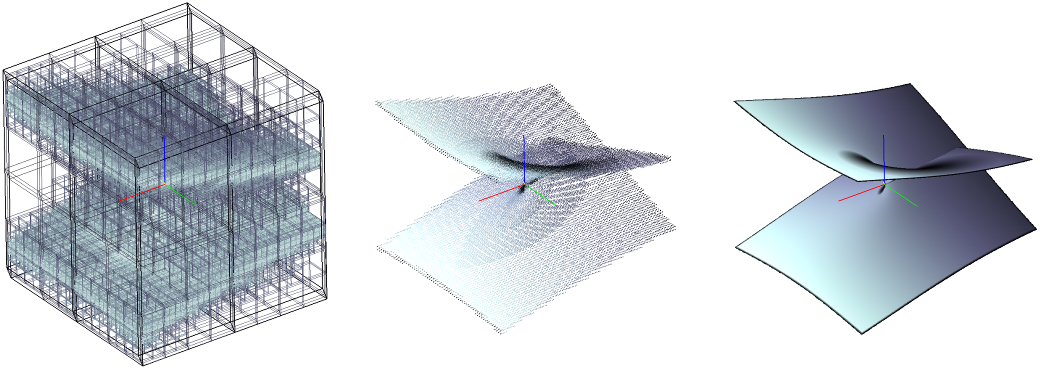
Paiva–Lopes–Lewiner–Figueiredo (2006)



flag regions of possible topological ambiguity

Implicit surfaces in 4D

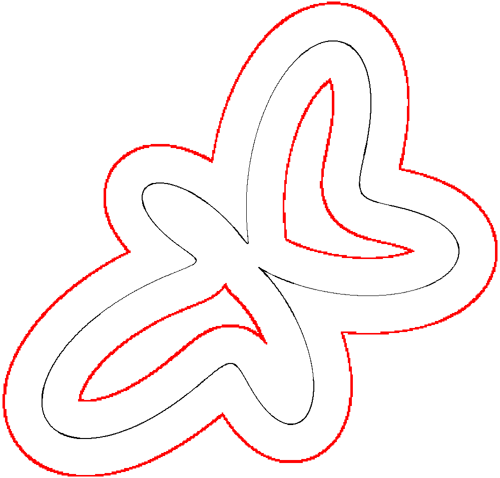
Bordignon–Sá–Lopes–Pesco–Figueiredo (2013)



seed points for point-based rendering

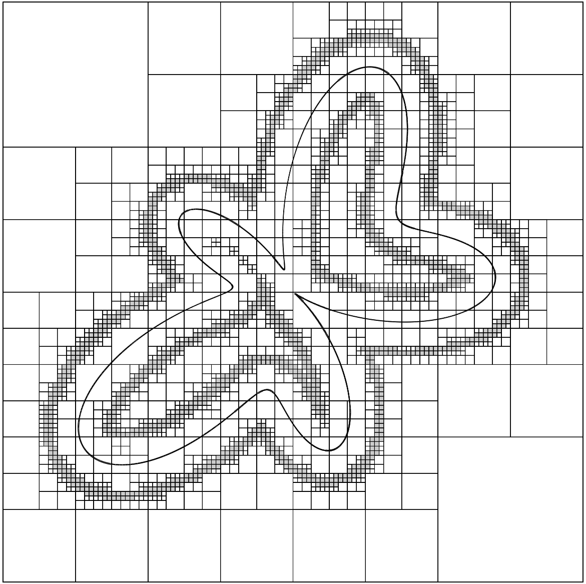
Offsets of parametric curves

Oliveira–Figueiredo (2003)



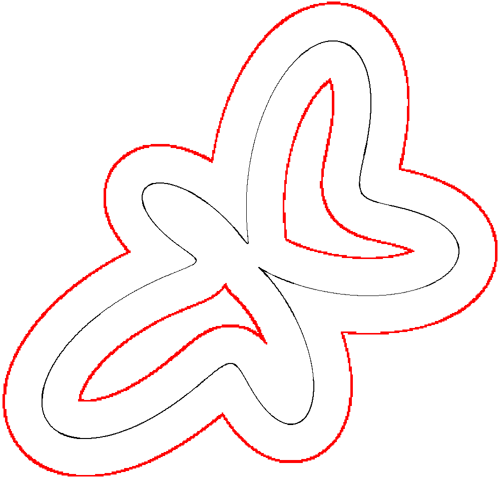
Offsets of parametric curves

Oliveira–Figueiredo (2003)



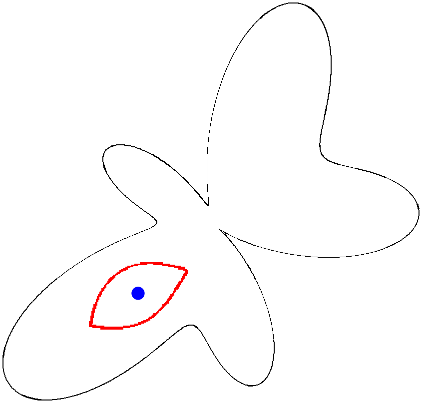
Offsets of parametric curves

Oliveira–Figueiredo (2003)



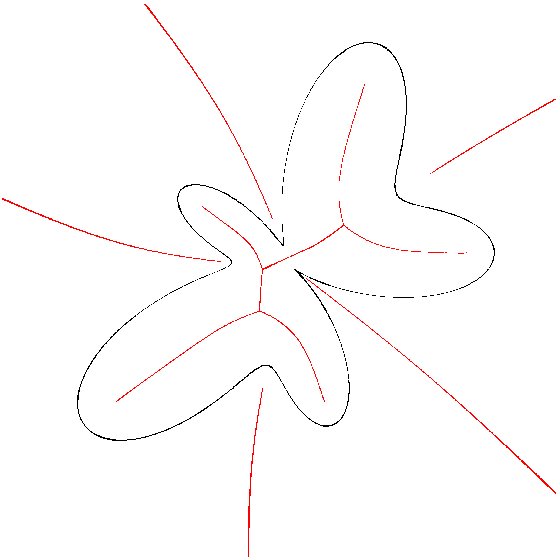
Bisectors of parametric curves

Oliveira–Figueiredo (2003)



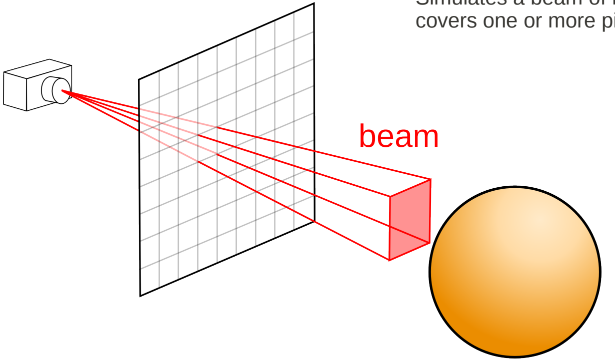
Medial axis of parametric curves

Oliveira–Figueiredo (2003)



Beam casting implicit surfaces

Ganacim–Figueiredo–Nehab (2011)



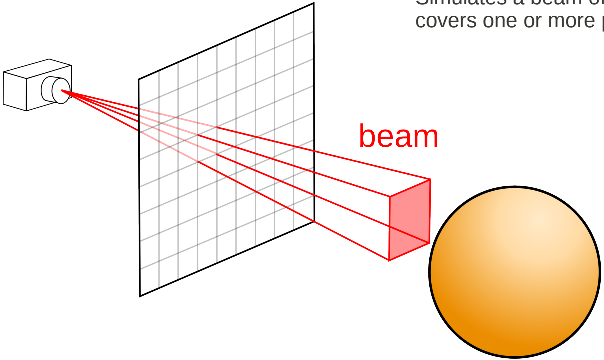
- Simulates a beam of rays that covers one or more pixels

Avoids sampling errors

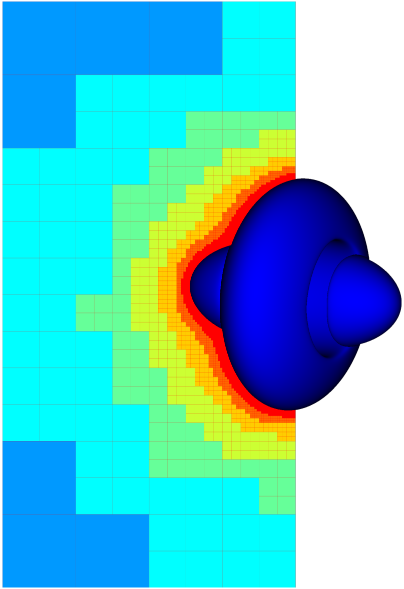
also Flórez et al. (2008)

Beam casting implicit surfaces

Ganacim–Figueiredo–Nehab (2011)



- Simulates a beam of rays that covers one or more pixels

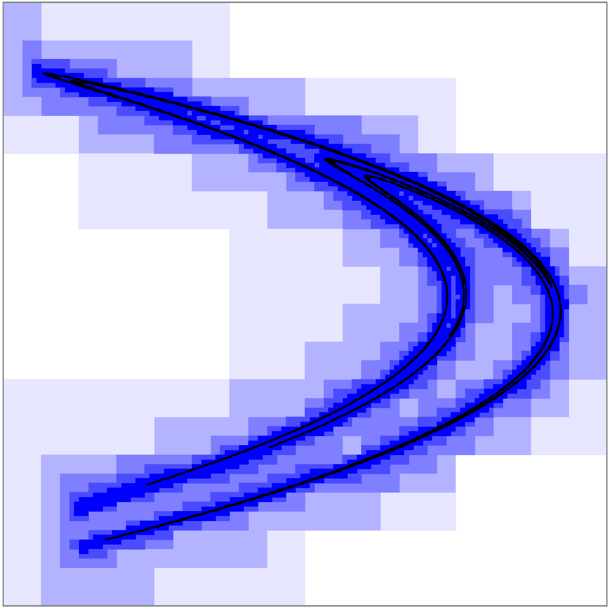


Avoids sampling errors

also Flórez et al. (2008)

Fractals

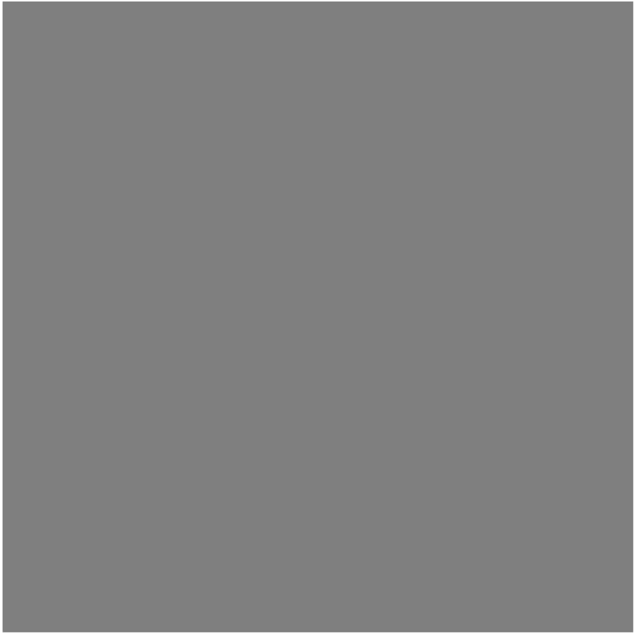
Paiva–Figueiredo–Stolfi (2006)



Hénon attractor

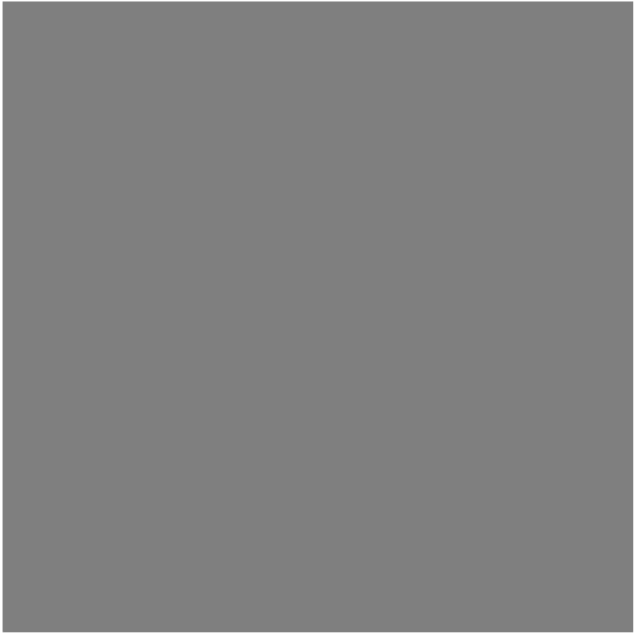
Julia sets

Fig–Nehab–Oliveira–Stolfi (2016)



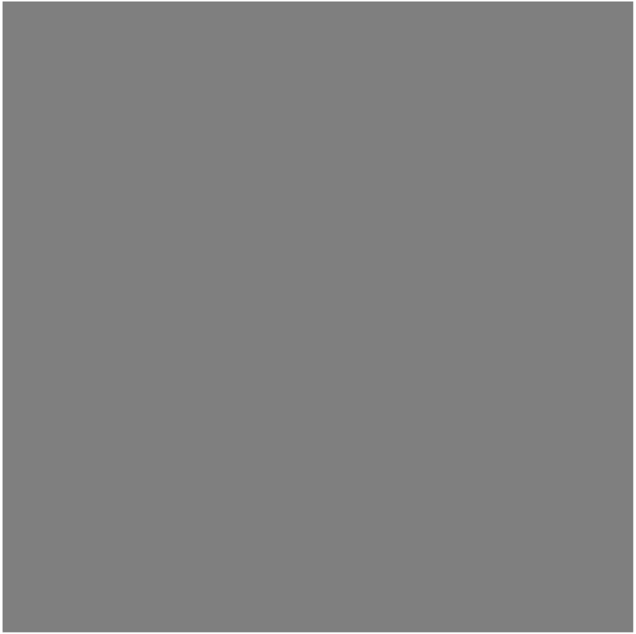
Julia sets

Fig–Nehab–Oliveira–Stolfi (2016)



Julia sets

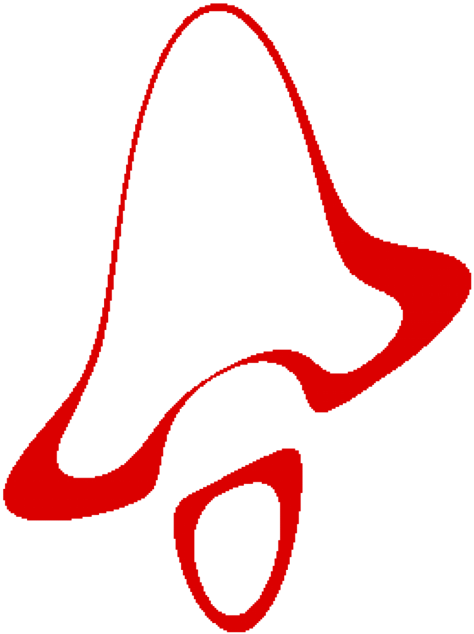
Fig–Nehab–Oliveira–Stolfi (2016)



but . . .

Overestimation

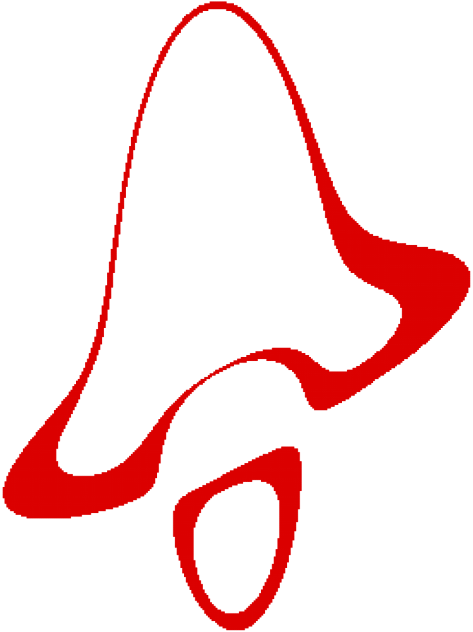
$$0.004+0.110x-0.177y-0.174x^2+0.224xy-0.303y^2$$
$$-0.168x^3+0.327x^2y-0.087xy^2-0.013y^3+0.235x^4$$
$$-0.667x^3y+0.745x^2y^2-0.029xy^3+0.072y^4=0$$



Overestimation

$$0.004+0.110x-0.177y-0.174x^2+0.224xy-0.303y^2$$
$$-0.168x^3+0.327x^2y-0.087xy^2-0.013y^3+0.235x^4$$
$$-0.667x^3y+0.745x^2y^2-0.029xy^3+0.072y^4=0$$

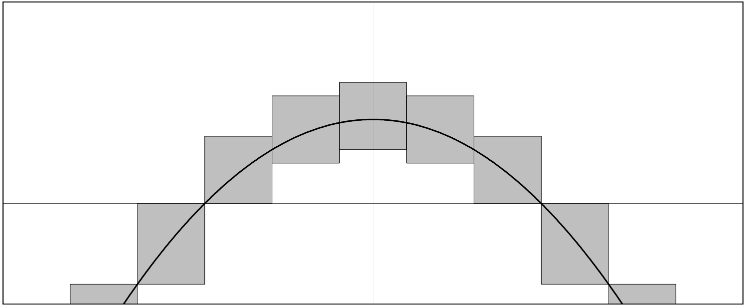
IA can't see correlations between operands



The dependency problem in interval arithmetic

$$f(x) = (10+x)(10-x) \text{ for } x \in [-2, 2]$$

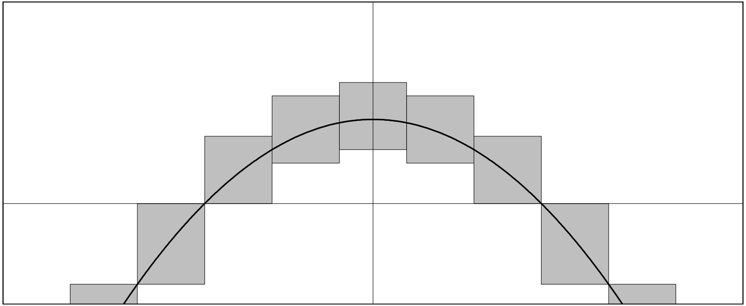
$$\begin{aligned} 10 + x &= [8, 12] \\ 10 - x &= [8, 12] \\ (10+x)(10-x) &= [64, 144] \qquad \text{diam} = 80 \\ \text{exact range} &= [96, 100] \qquad \text{diam} = 4 \end{aligned}$$



The dependency problem in interval arithmetic

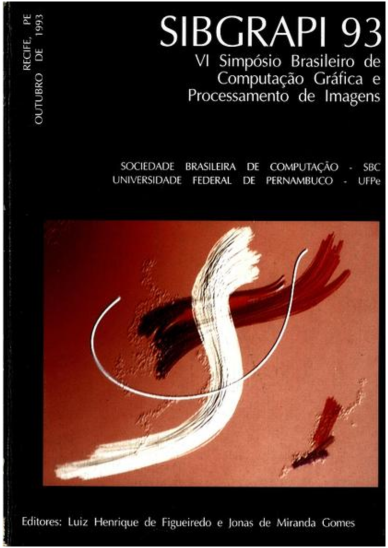
$$f(x) = (10 + x)(10 - x) \text{ for } x \in [-u, u]$$

$$\begin{aligned}
10 + x &= [10 - u, 10 + u] \\
10 - x &= [10 - u, 10 + u] \\
(10 + x)(10 - x) &= [(10 - u)^2, (10 + u)^2] \qquad \text{diam} = 40u \\
\text{exact range} &= [100 - u^2, 100] \qquad \text{diam} = u^2
\end{aligned}$$



affine arithmetic

Affine arithmetic and its applications to computer graphics Comba–Stolfi (1993)



Affine Arithmetic and its Applications to Computer Graphics

João Luiz Dihl Comba[1]
Jorge Stolfi[2]

[1]Computer Graphics Laboratory (LCG-COPPE)
Universidade Federal do Rio de Janeiro
Caixa Postal 68511 - Rio de Janeiro, RJ, Brasil
comba@lcg.ufrj.br

[2]Computer Science Department (DCC–IMECC)
Universidade Estadual de Campinas
Caixa Postal 6065 – 13081 Campinas, SP, Brasil
stolfi@dcc.unicamp.br

Abstract. We describe a new method for numeric computations, which we call *affine arithmetic* (AA). This model is similar to standard interval arithmetic, to the extent that it automatically keeps track of rounding and truncation errors for each computed value. However, by taking into account correlations between operands and sub-formulas, AA is able to provide much tighter bounds for the computed quantities, with errors that are approximately quadratic in the uncertainty of the input variables. We also describe two applications of AA to computer graphics problems, where this feature is particularly valuable: namely, ray tracing and the construction of octrees for implicit surfaces.

1 Introduction

Interval arithmetic (IA), also known as *interval analysis*, is a technique for numerical computation where each quantity is represented by an interval of floating-point numbers. Those intervals are added, subtracted, multiplied, etc. in such a way that each computed interval is guaranteed to contain the (unknown) value of the quantity it represents [3, 4].

Since its introduction in the 60's by R. E. Moore, IA became widely appreciated for its ability to manipulate imprecise data, keep track automatically of truncation and round-off errors, and probe the behavior of functions efficiently and reliably over whole sets of arguments at once.

Recently, this last feature of IA caught the attention of computer graphics researchers, who put it to good use in ray tracing (determining ray-surface intersections), solid modeling (constructing octrees for implicit surfaces), and other problems [1, 5, 6, 7].

The main weakness of IA is that it tends to be too conservative: the intervals it produces are often much wider than the true range of the corresponding quantities, often to the point of uselessness. This problem is particularly severe in long computation chains, where the intervals computed at one stage are inputs for the next stage. Unfortunately, such "deep" computations are not uncommon in the computer graphics applications mentioned above.

To address this problem, we propose here a new model for numerical computation, which we call *affine arithmetic* (AA). Like standard IA, AA keeps track automatically of the round-off and truncation errors affecting each computed quantity. In addition, AA keeps track of *correlations* between those quantities. Thanks to this extra information, AA is able to provide much tighter intervals than IA, especially in long computation chains.

As one may expect, the AA model is more complex and expensive than ordinary interval arithmetic. However, we believe that its higher accuracy will be worth the extra cost in many applications, including computer graphics.

Section 2 of the paper is a brief review of standard IA and its "error explosion" problem. Section 3 defines *affine forms*, the representation of quantities in the AA model. Section 4 gives the basic principle for computing with affine forms, and applies it to a couple of basic operations (addition, multiplication, and square root). Section 5 describes some technical details of our implementation of AA. Finally, section 6 discusses some applications of AA in computer graphics.

Affine arithmetic

Comba–Stolfi (1993)

AA represents a quantity x with an affine form

$$\hat{x} = x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n$$

Noise symbols ε_i : independent, vary in $[-1, +1]$ but are otherwise unknown

Can compute arbitrary formulas on affine forms
Use affine approximations for non-affine operations
New noise symbols created during computation

AA generalizes IA:

$$x \sim \hat{x} \implies x \in [x_0 - \delta,\, x_0 + \delta] \quad \text{for} \quad \delta = |x_1| + \cdots + |x_n|$$

$$x \in [a, b] \implies x \sim \hat{x} = x_0 + x_1\varepsilon_1 \quad \text{for} \quad x_0 = (b+a)/2,\ x_1 = (b-a)/2$$

Affine arithmetic

Comba–Stolfi (1993)

AA represents a quantity x with an affine form

$$\hat{x} = x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n$$

Noise symbols ε_i : independent, vary in $[-1, +1]$ but are otherwise unknown

Can compute arbitrary formulas on affine forms
Use affine approximations for non-affine operations
New noise symbols created during computation

AA generalizes IA:

$$x \sim \hat{x} \quad \Longrightarrow \quad x \in [x_0 - \delta,\, x_0 + \delta] \quad \text{for} \quad \delta = |x_1| + \cdots + |x_n|$$

$$x \in [a, b] \quad \Longrightarrow \quad x \sim \hat{x} = x_0 + x_1\varepsilon_1 \quad \text{for} \quad x_0 = (b+a)/2,\; x_1 = (b-a)/2$$

AA automatically exploits first-order correlations in complex expressions

Affine arithmetic

Comba–Stolfi (1993)

AA represents a quantity x with an affine form

$$\hat{x} = x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n$$

Noise symbols ε_i : independent, vary in $[-1, +1]$ but are otherwise unknown

Can compute arbitrary formulas on affine forms
Use affine approximations for non-affine operations
New noise symbols created during computation

AA generalizes IA:

$$x \sim \hat{x} \quad \Longrightarrow \quad x \in [x_0 - \delta,\, x_0 + \delta] \quad \text{for} \quad \delta = |x_1| + \cdots + |x_n|$$

$$x \in [a, b] \quad \Longrightarrow \quad x \sim \hat{x} = x_0 + x_1\varepsilon_1 \quad \text{for} \quad x_0 = (b+a)/2,\ x_1 = (b-a)/2$$

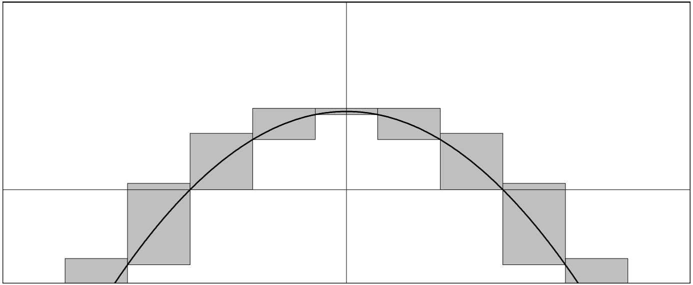
AA automatically exploits first-order correlations in complex expressions

$\Longrightarrow$ better interval estimates!

The dependency problem in interval arithmetic – with AA

$$f(x) = (10+x)(10-x) \text{ for } x \in [-u, u], \quad x = 0 + u\,\varepsilon_1$$

$$\begin{aligned}
10 + x &= 10 - u\,\varepsilon_1 \\
10 - x &= 10 + u\,\varepsilon_1 \\
(10+x)(10-x) &= 100 - u^2\,\varepsilon_2 \\
\text{range} &= [100 - u^2, 100 + u^2] \qquad \text{diam} = 2u^2 \\
\text{exact range} &= [100 - u^2, 100] \qquad \text{diam} = u^2
\end{aligned}$$

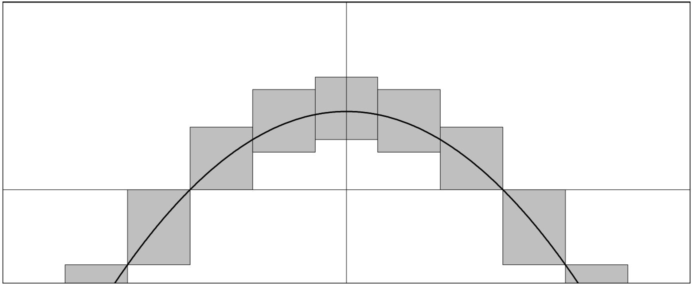


AA

The dependency problem in interval arithmetic – with AA

$$f(x) = (10+x)(10-x) \text{ for } x \in [-u,u], \quad x = 0 + u\,\varepsilon_1$$

$$\begin{aligned}
10 + x &= 10 - u\,\varepsilon_1 \\
10 - x &= 10 + u\,\varepsilon_1 \\
(10+x)(10-x) &= 100 - u^2\,\varepsilon_2 \\
\text{range} &= [100 - u^2, 100 + u^2] \qquad \text{diam} = 2u^2 \\
\text{exact range} &= [100 - u^2, 100] \qquad \text{diam} = u^2
\end{aligned}$$



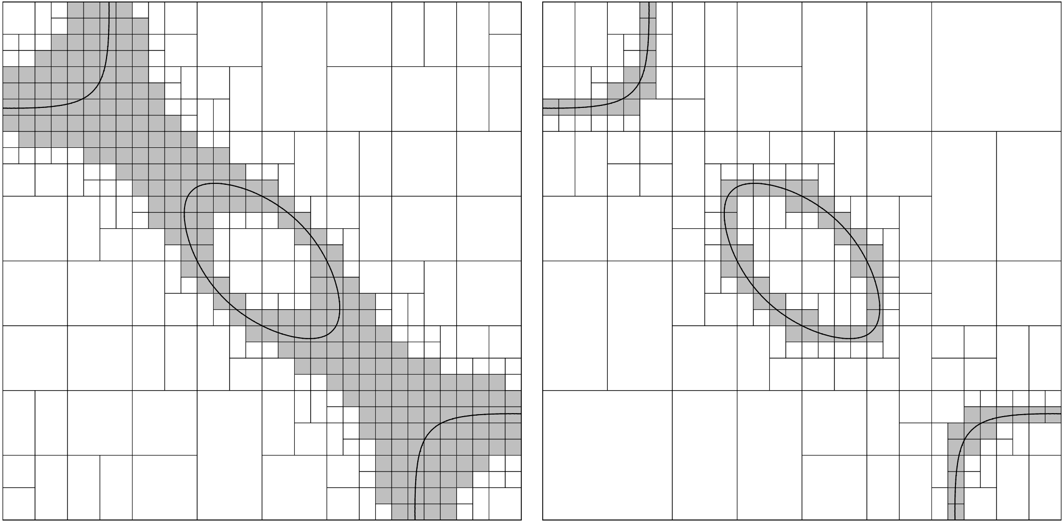
IA

replacing IA with AA

IA versus AA for plotting implicit curves

Comba–Stolfi (1993)

$$x^2 + y^2 + xy - (xy)^2/2 - 1/4 = 0$$



IA: 246 exact: 66 AA: 70

Interval method for intersecting two parametric surfaces

Gleicher–Kass (1992)

Parametric surfaces

$$g_1\colon D_1 \subset \mathbf{R}^2 \to \mathbf{R}^3$$

$$g_2\colon D_2 \subset \mathbf{R}^2 \to \mathbf{R}^3$$

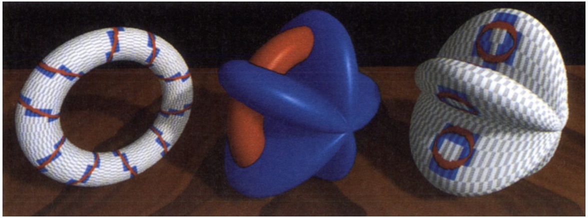
Intersection

$$g_1(u_1, v_1) - g_2(u_2, v_2) = 0$$

Interval test

$$G_1(U_1, V_1) \cap G_2(U_2, V_2) \neq \varnothing$$

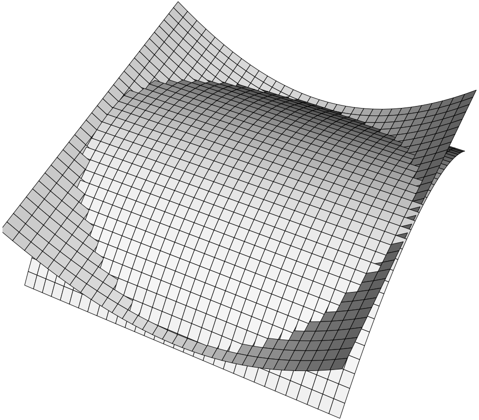
Intersect bounding boxes in space
Discard if no intersection
Subdivide until tolerance
String boxes into curves



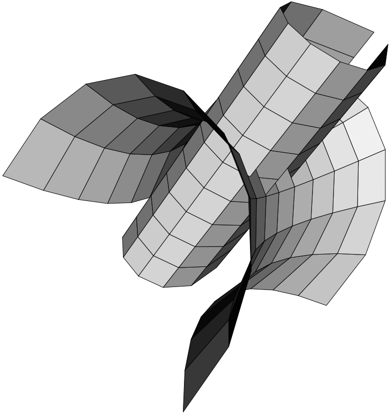
Replacing IA with AA for surface intersection

tensor product Bézier surfaces of degree (p, q)

$$s(u,v) = \sum_{i=0}^{p}\sum_{j=0}^{q} a_{ij} B_i^p(u) B_j^q(v), \quad B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad u,v \in [0,1]$$



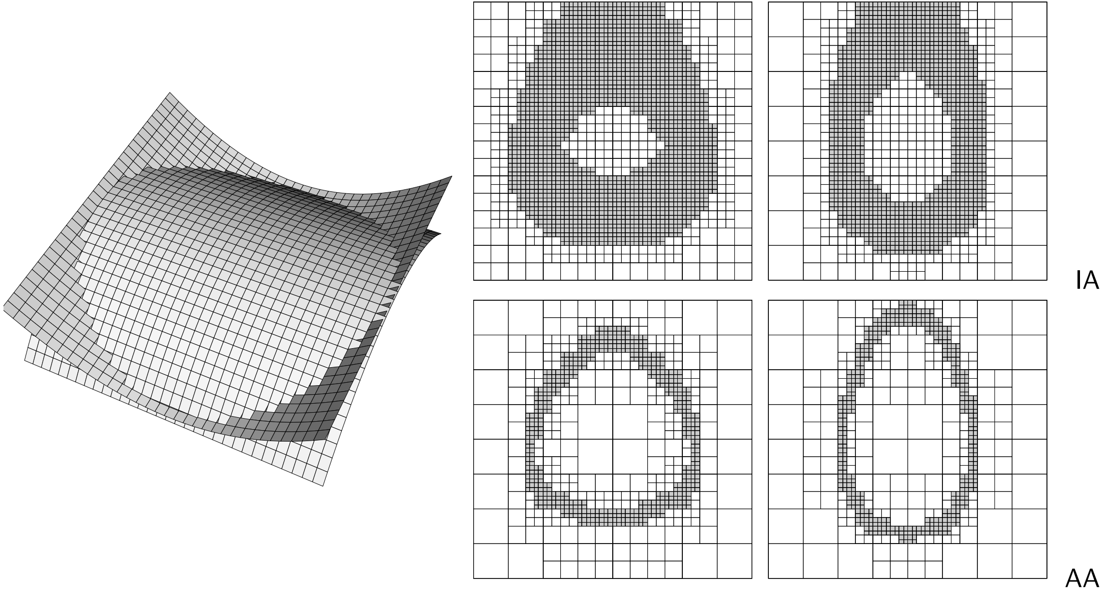
$(2, 1)$



$(3, 3)$

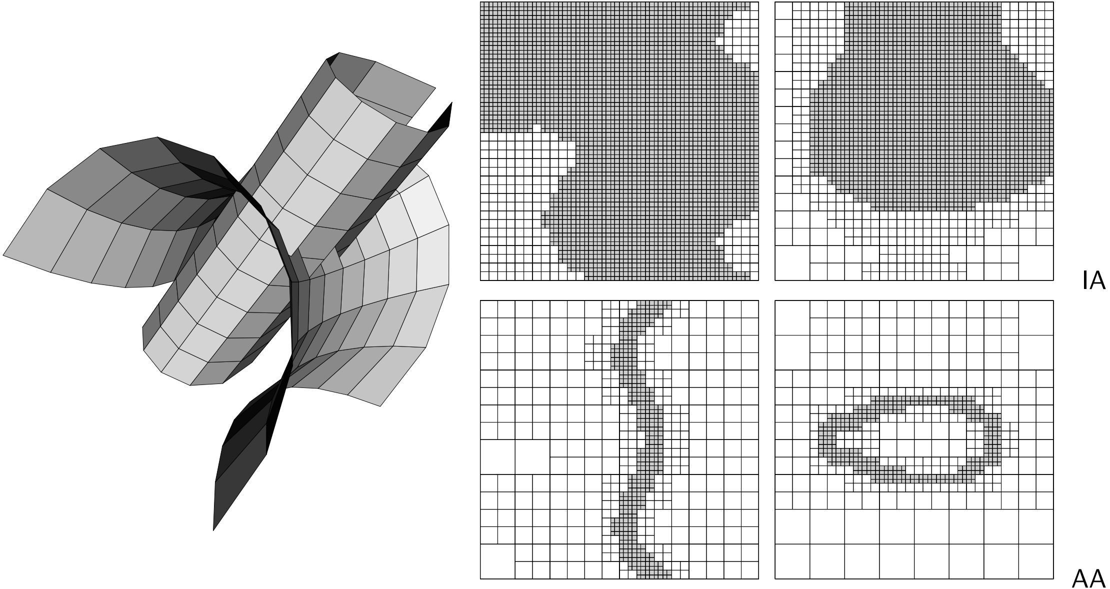
Replacing IA with AA for surface intersection

Figueiredo (1996)



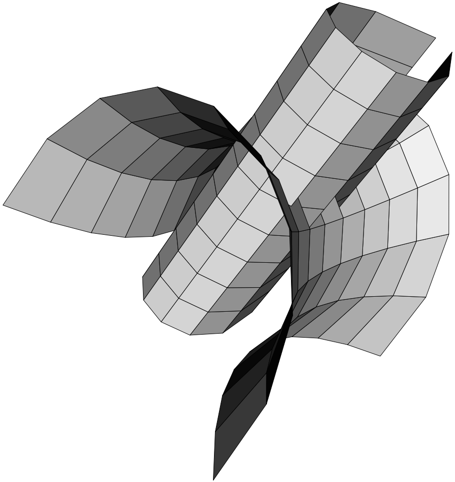
Replacing IA with AA for surface intersection

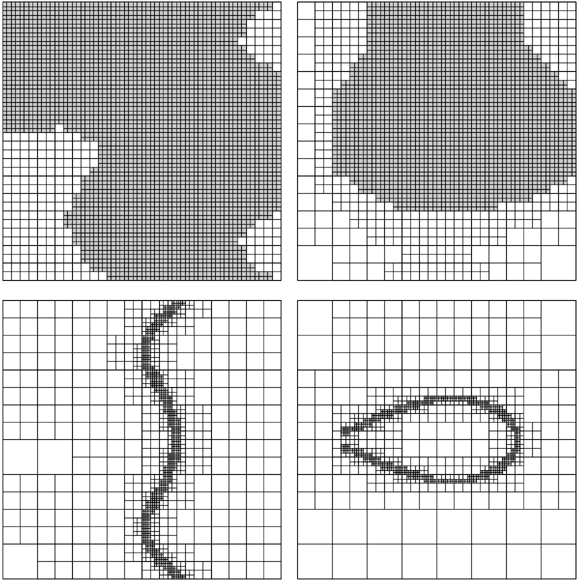
Figueiredo (1996)



Replacing IA with AA for surface intersection

Figueiredo (1996)





IA

AA

exploiting geometry in AA

Geometry of affine forms

Affine forms that share noise symbols are not independent:

$$\begin{aligned} \hat{x} &= x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n \\ \hat{y} &= y_0 + y_1\varepsilon_1 + \cdots + y_n\varepsilon_n \end{aligned}$$

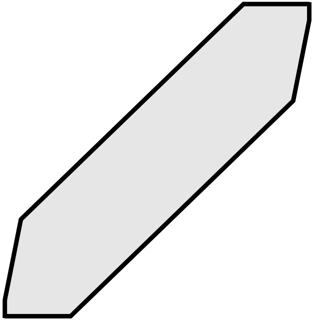
Joint range is a zonotope: centrally symmetric convex polygon

Image of hypercube $[-1,1]^n$ under affine transformation

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} x_1 & \cdots & x_n \\ y_1 & \cdots & y_n \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Minkowski sum of point and line segments

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}\varepsilon_1 + \cdots + \begin{bmatrix} x_n \\ y_n \end{bmatrix}\varepsilon_n$$



Geometry of affine forms

Affine forms that share noise symbols are not independent:

$$\begin{aligned}\hat{x} &= x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n \\ \hat{y} &= y_0 + y_1\varepsilon_1 + \cdots + y_n\varepsilon_n\end{aligned}$$

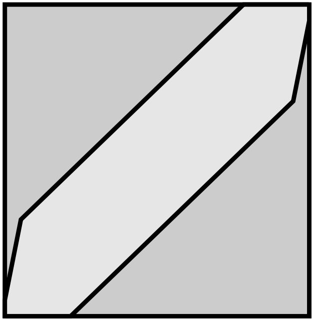
Joint range is a zonotope: centrally symmetric convex polygon

Image of hypercube $[-1,1]^n$ under affine transformation

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} x_1 & \cdots & x_n \\ y_1 & \cdots & y_n \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Minkowski sum of point and line segments

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \varepsilon_1 + \cdots + \begin{bmatrix} x_n \\ y_n \end{bmatrix} \varepsilon_n$$



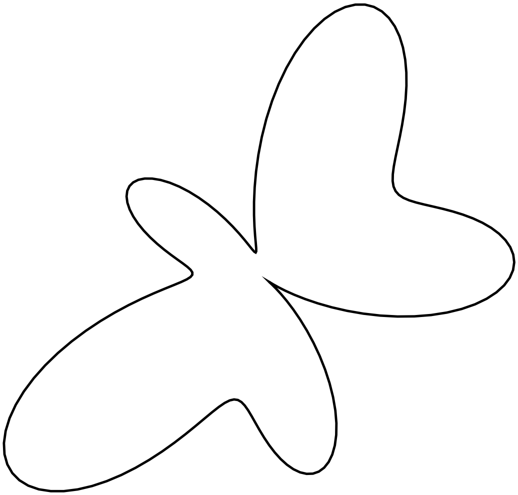
Approximating parametric curves

Parametric curve

$$\mathcal{C} = \gamma(I), \qquad \gamma\colon I \to \mathbf{R}^2$$

Compute good bounding rectangle for

$$\mathcal{P} = \gamma(T), \qquad T \subseteq I$$



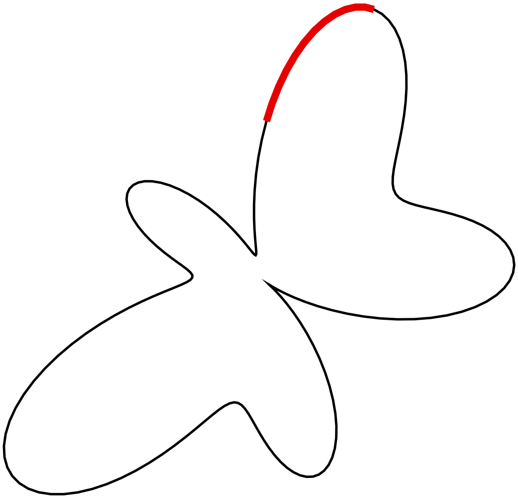
Approximating parametric curves

Parametric curve

$$\mathcal{C} = \gamma(I), \qquad \gamma\colon I \to \mathbf{R}^2$$

Compute good bounding rectangle for

$$\mathcal{P} = \gamma(T), \qquad T \subseteq I$$



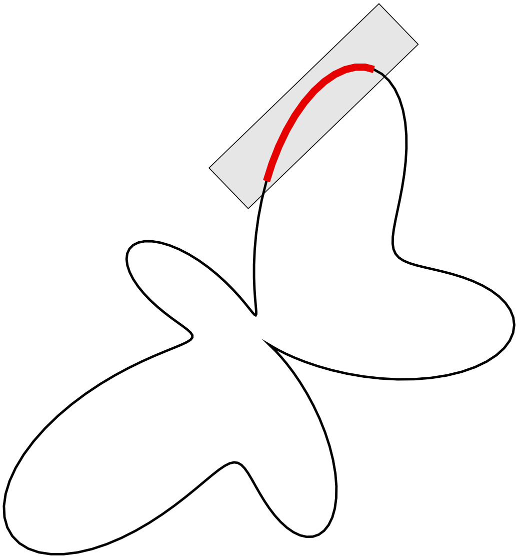
Approximating parametric curves

Parametric curve

$$\mathcal{C} = \gamma(I), \qquad \gamma\colon I \to \mathbf{R}^2$$

Compute good bounding rectangle for

$$\mathcal{P} = \gamma(T), \qquad T \subseteq I$$



Approximating parametric curves

Parametric curve

$$\mathcal{C} = \gamma(I), \qquad \gamma\colon I \to \mathbf{R}^2$$

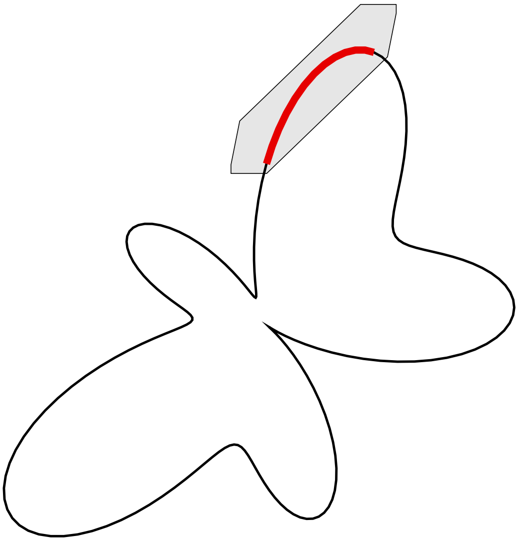
Compute good bounding rectangle for

$$\mathcal{P} = \gamma(T), \qquad T \subseteq I$$

Write

$$\gamma(t) = (x(t), y(t))$$

Find joint range of $\hat{x}(\hat{t})$ and $\hat{y}(\hat{t})$ with AA



Approximating parametric curves

Parametric curve

$$\mathcal{C} = \gamma(I), \qquad \gamma\colon I \to \mathbf{R}^2$$

Compute good bounding rectangle for

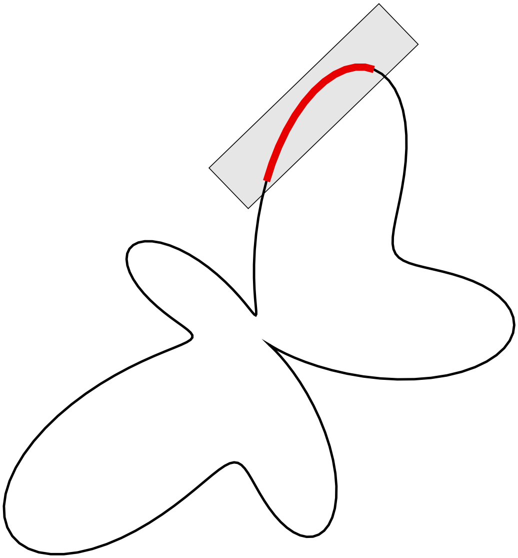
$$\mathcal{P} = \gamma(T), \qquad T \subseteq I$$

Write

$$\gamma(t) = (x(t), y(t))$$

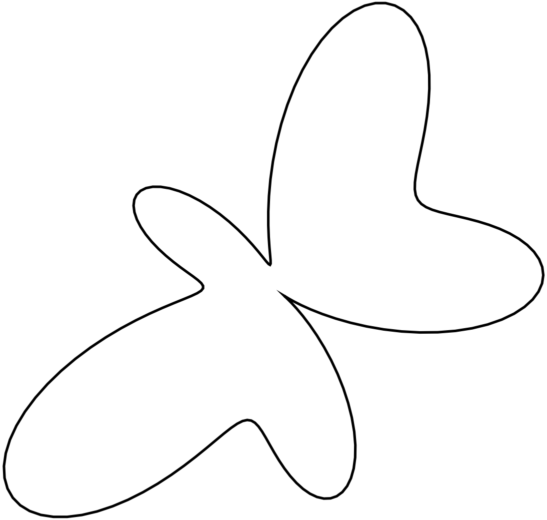
Find joint range of $\hat{x}(\hat{t})$ and $\hat{y}(\hat{t})$ with AA

Use bounding rectangle of zonotope



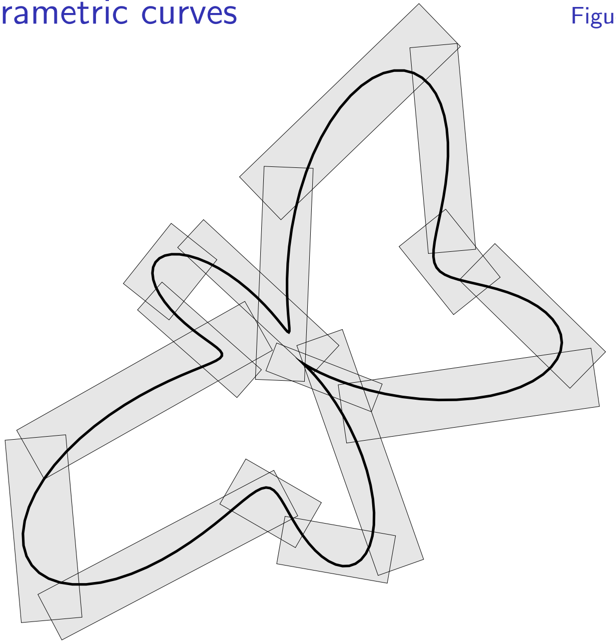
Approximating parametric curves

Figueiredo–Stolfi–Velho (2003)



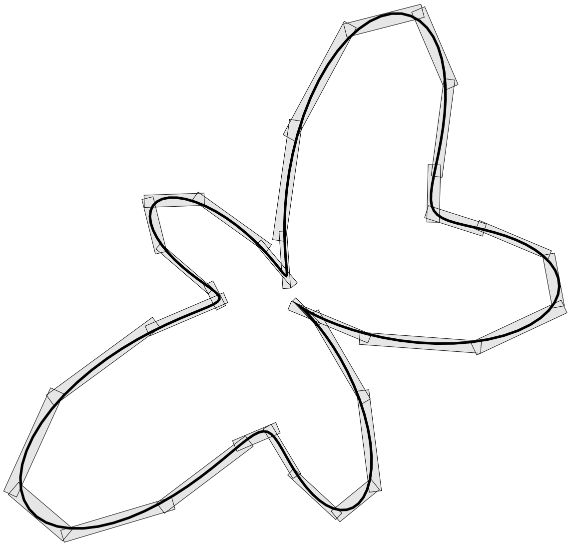
Approximating parametric curves

Figueiredo–Stolfi–Velho (2003)



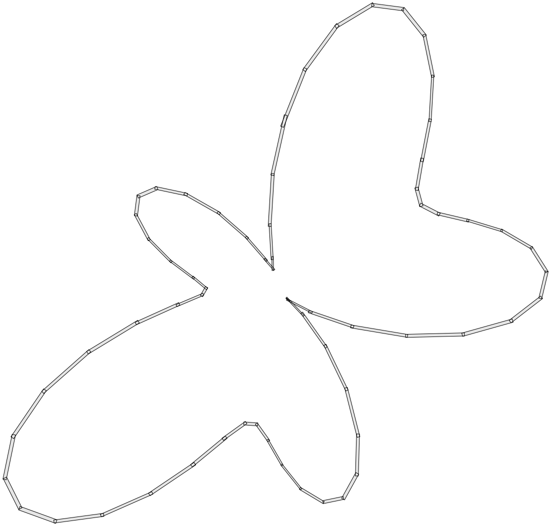
Approximating parametric curves

Figueiredo–Stolfi–Velho (2003)



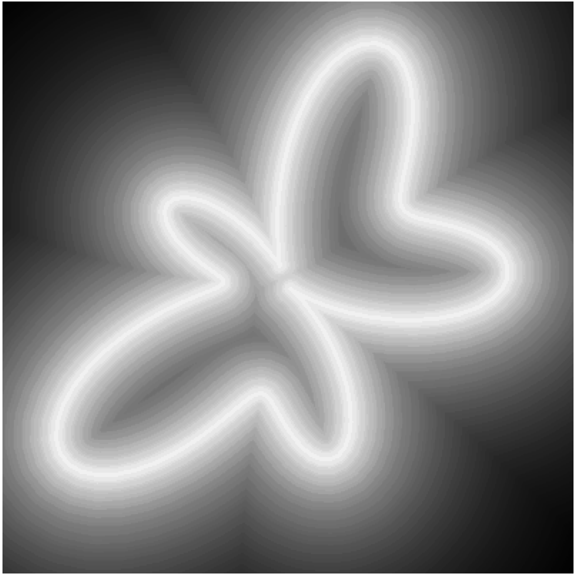
Approximating parametric curves

Figueiredo–Stolfi–Velho (2003)



Distance fields for parametric curves

Figueiredo–Stolfi–Velho (2003)



Ray casting implicit surfaces

Implicit surface

$$h(x, y, z) = 0, \qquad h\colon \mathbf{R}^3 \to \mathbf{R}$$

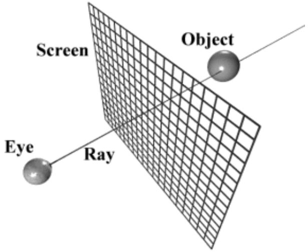
Ray

$$r(t) = e + t \cdot v = (x(t), y(t), z(t)), \quad t \in [0, \infty)$$

Ray intersects surface when

$$f(t) = h(r(t)) = 0$$

First intersection occurs at smallest zero of f in $[0, \infty)$



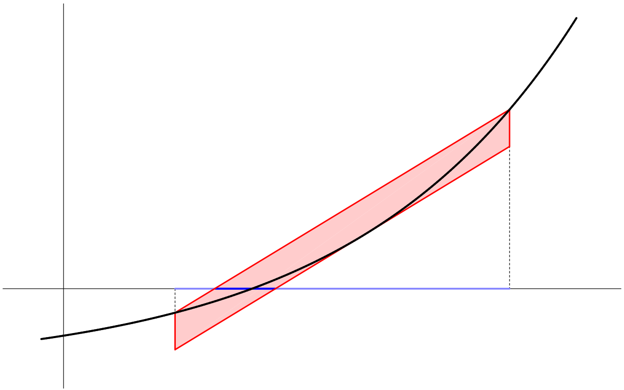
Ray casting implicit surfaces

```
procedure interval-bisection([a, b])
    if 0 ∈ F([a, b]) then
        c ← (a + b)/2
        if (b − a) < ε then
            return c
        else
            interval-bisection([a, c])      ← try left half first!
            interval-bisection([c, b])
        end
    end
end
```

Call interval-bisection($[0, t_\infty]$) to find the first zero

Ray casting implicit surfaces

Custatis–Figueiredo–Gattass (1999)

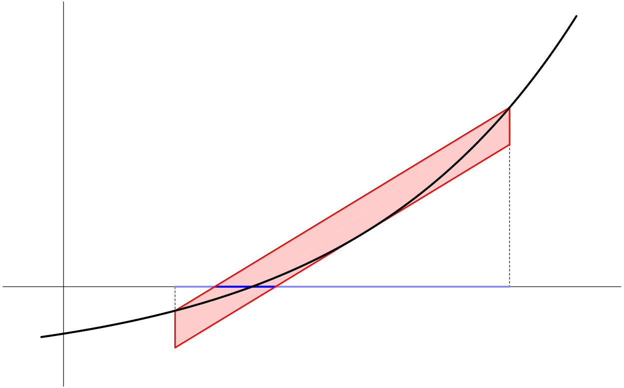


AA exploits linear correlations in

$$f(t) = h(r(t))$$
$$r(t) = (x(t), y(t), z(t))$$

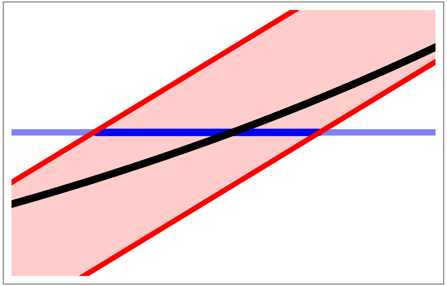
Ray casting implicit surfaces

Custatis–Figueiredo–Gattass (1999)



AA exploits linear correlations in

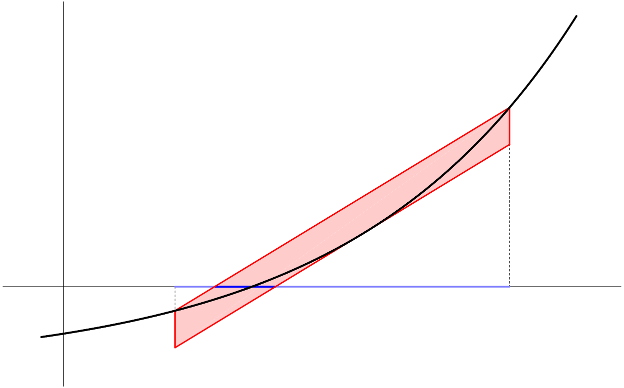
$$f(t) = h(r(t))$$
$$r(t) = (x(t), y(t), z(t))$$



root must lie in smaller interval

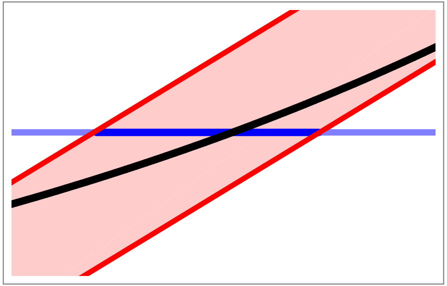
Ray casting implicit surfaces

Custatis–Figueiredo–Gattass (1999)



AA exploits linear correlations in

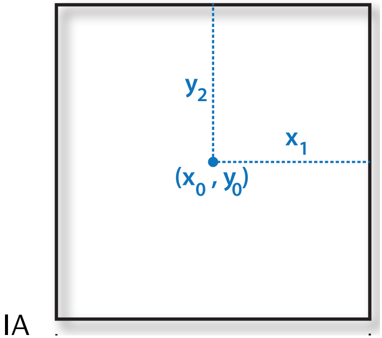
$$f(t) = h(r(t))$$
$$r(t) = (x(t), y(t), z(t))$$

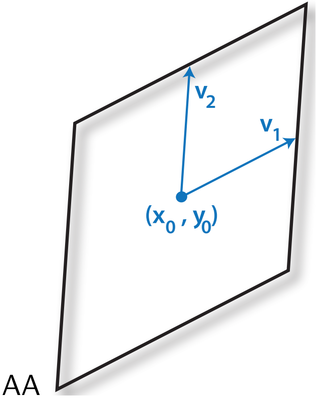


root must lie in smaller interval

quadratic convergence

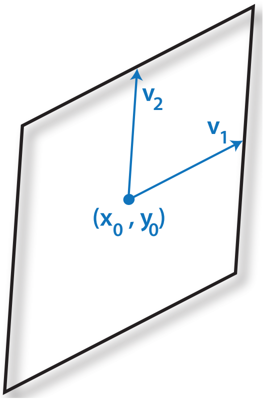
Natural domains





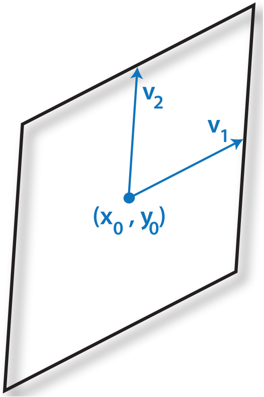
$$(\hat{x}, \hat{y}) = (x_0, y_0) + v_1\varepsilon_1 + v_2\varepsilon_2$$

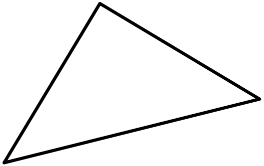
AA on triangles



$$(\hat{x}, \hat{y}) = (x_0, y_0) + v_1\varepsilon_1 + v_2\varepsilon_2$$

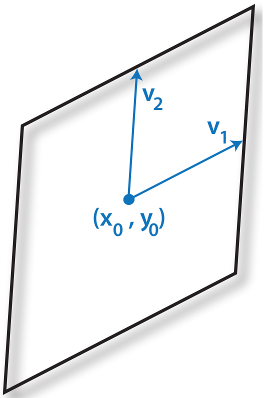
AA on triangles

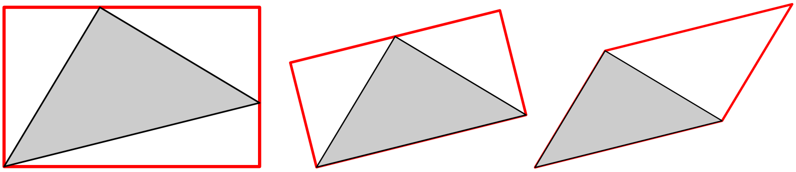




$$(\hat{x}, \hat{y}) = (x_0, y_0) + v_1\varepsilon_1 + v_2\varepsilon_2$$

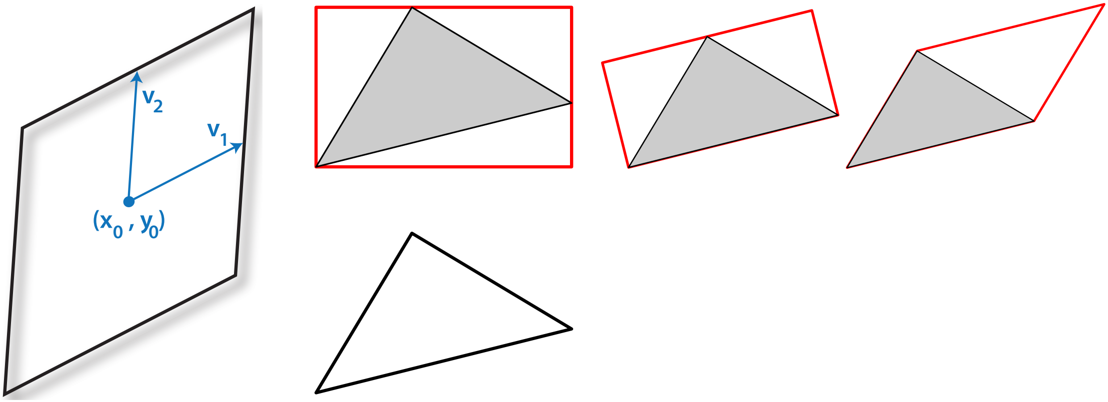
AA on triangles





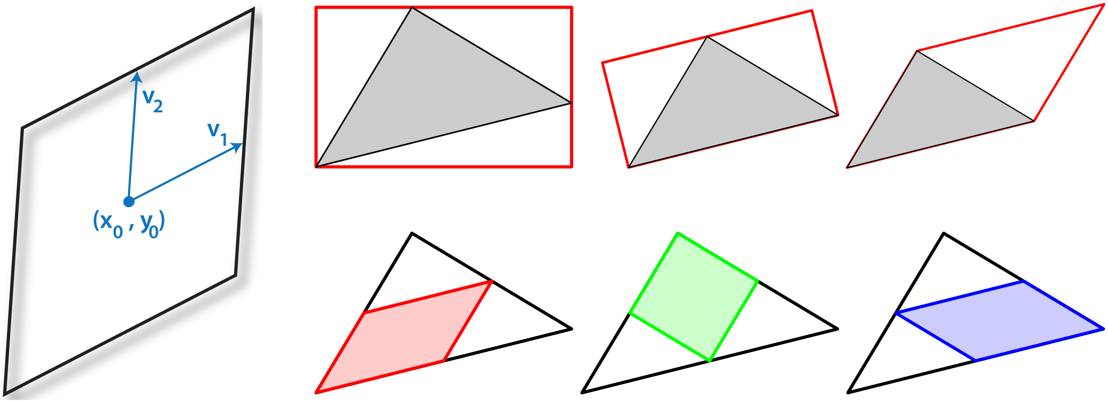
$$(\hat{x}, \hat{y}) = (x_0, y_0) + v_1\varepsilon_1 + v_2\varepsilon_2$$

AA on triangles



$$(\hat{x}, \hat{y}) = (x_0, y_0) + v_1\varepsilon_1 + v_2\varepsilon_2$$

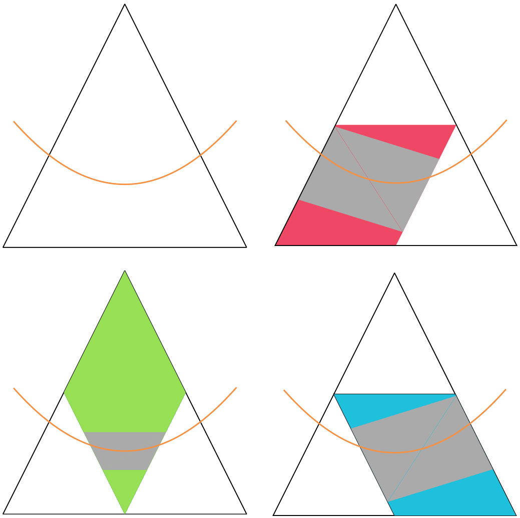
AA on triangles



$$(\hat{x}, \hat{y}) = (x_0, y_0) + v_1\varepsilon_1 + v_2\varepsilon_2$$

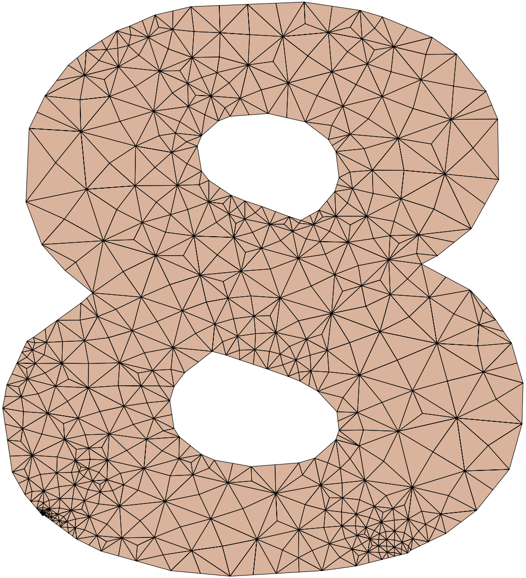
Implicit curves on triangles

Nascimento–Paiva–Figueiredo–Stolfi (2014)



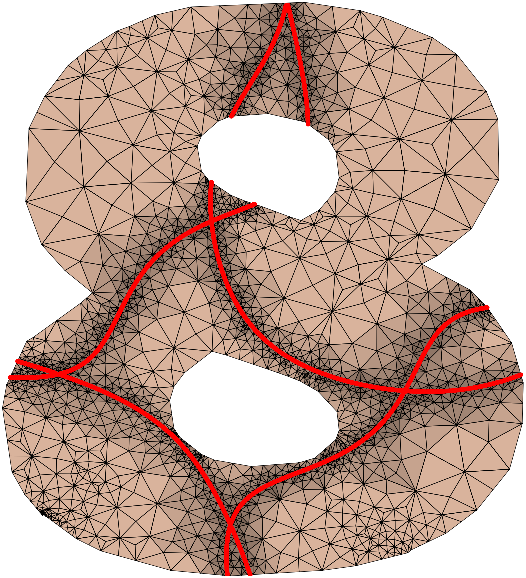
Implicit curves on triangulations

Nascimento–Paiva–Figueiredo–Stolfi (2014)



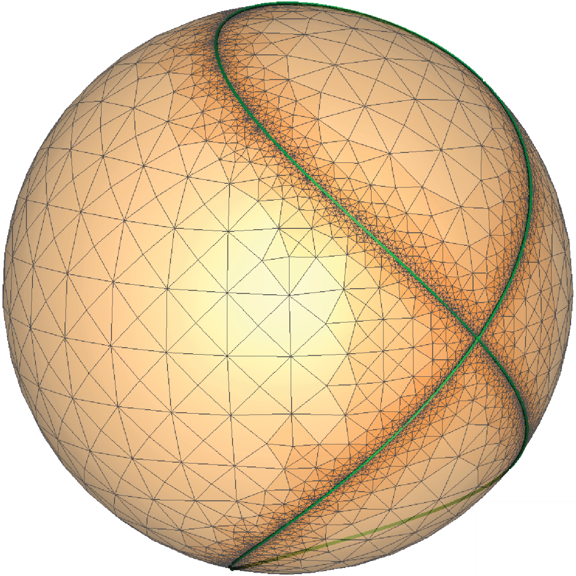
Implicit curves on triangulations

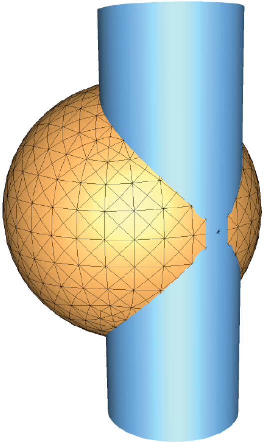
Nascimento–Paiva–Figueiredo–Stolfi (2014)



Implicit curves on triangulations

Nascimento–Paiva–Figueiredo–Stolfi (2014)





Conclusion

Interval methods

- can reliably probe the global behavior of functions without sampling
- lead naturally to robust adaptive algorithms
- useful in many domains

Affine arithmetic is a useful tool for interval methods

- AA can replace IA transparently
- AA more accurate than IA
- AA locally more expensive than IA but globally more efficient
- AA provides geometric information that can be exploited
- AA can be used on triangles

Conclusion

Interval methods

- can reliably probe the global behavior of functions without sampling
- lead naturally to robust adaptive algorithms
- useful in many domains

Affine arithmetic is a useful tool for interval methods

- AA can replace IA transparently
- AA more accurate than IA
- AA locally more expensive than IA but globally more efficient
- AA provides geometric information that can be exploited
- AA can be used on triangles

Lots more to be done!

SIBGRAPI'17

30th Conference on Graphics, Patterns and Images

Interval methods

for computer graphics and geometric modeling

Luiz Henrique de Figueiredo

impa
Instituto de Matemática Pura e Aplicada