MAX-PLANCK-GESELLSCHAFT

# Reliable Geometric Computation via Controlled Perturbation !?

Kurt Mehlhorn

Max-Planck-Institute für Informatik

# Overview

- Geometric Computing
- Pitfalls of Geometric Computing
- Approaches to Reliable Geometric Computing
- Controlled Perturbation
  - The Principle
  - Applicability and Limits
  - Practicability
  - Open Problems
- Sources
  - L. Kettner, KM, S. Pion, S. Schirra, C. Yap: Classroom Examples of Robustness Problems in Geometric Computations, ESA 2004, LNCS 3221, 702–713.
  - S. Funke, Ch. Klein, KM, S. Schmitt: Controlled Perturbation for Delaunay Triangulations, SODA 2005, 1047-1056.
  - KM and R. Osbild: Reliable and Efficient Computational Geometry via Controlled Perturbation (Extended Abstract), submitted
  - and the papers cited therein

# Key Messages

MAX-PLANCK-GESELLSCHAFT

**Reliable implementation of geometric algorithms is a challenging task**

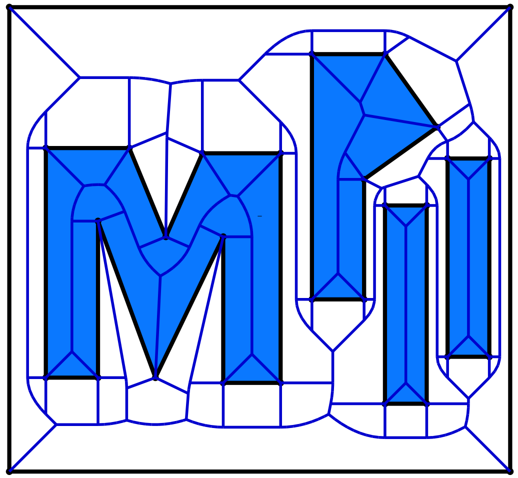
**Controlled Perturbation leads to reliable and efficient implementations with little additional work.**

**It is applicable to a large class of geometric algorithms.**

**Warning: the problem is not solved for the input given, but for a slightly perturbed input.**

# Geometric Computing I





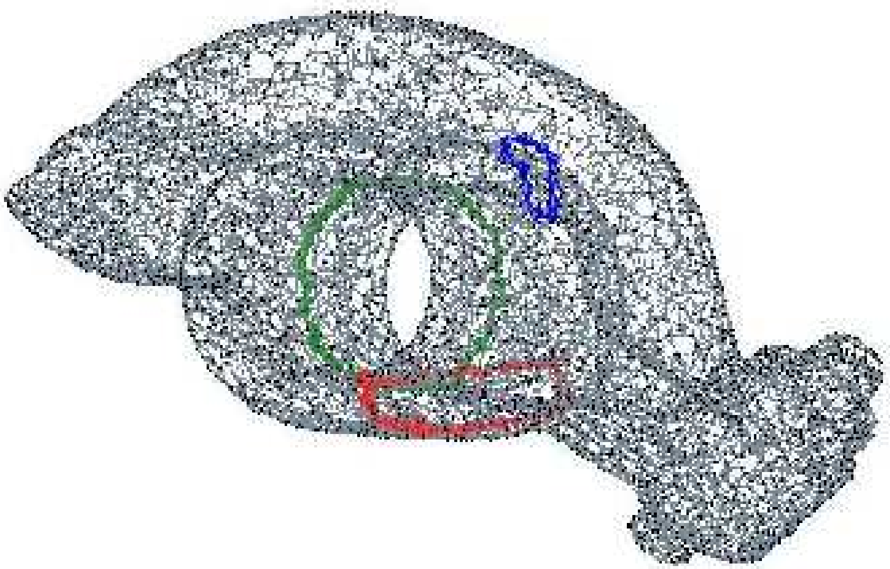
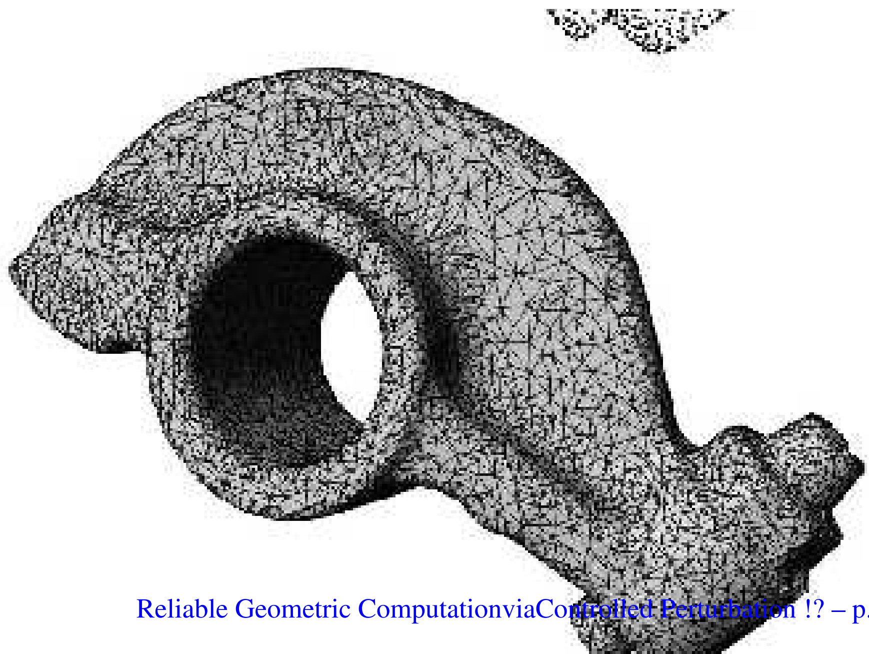
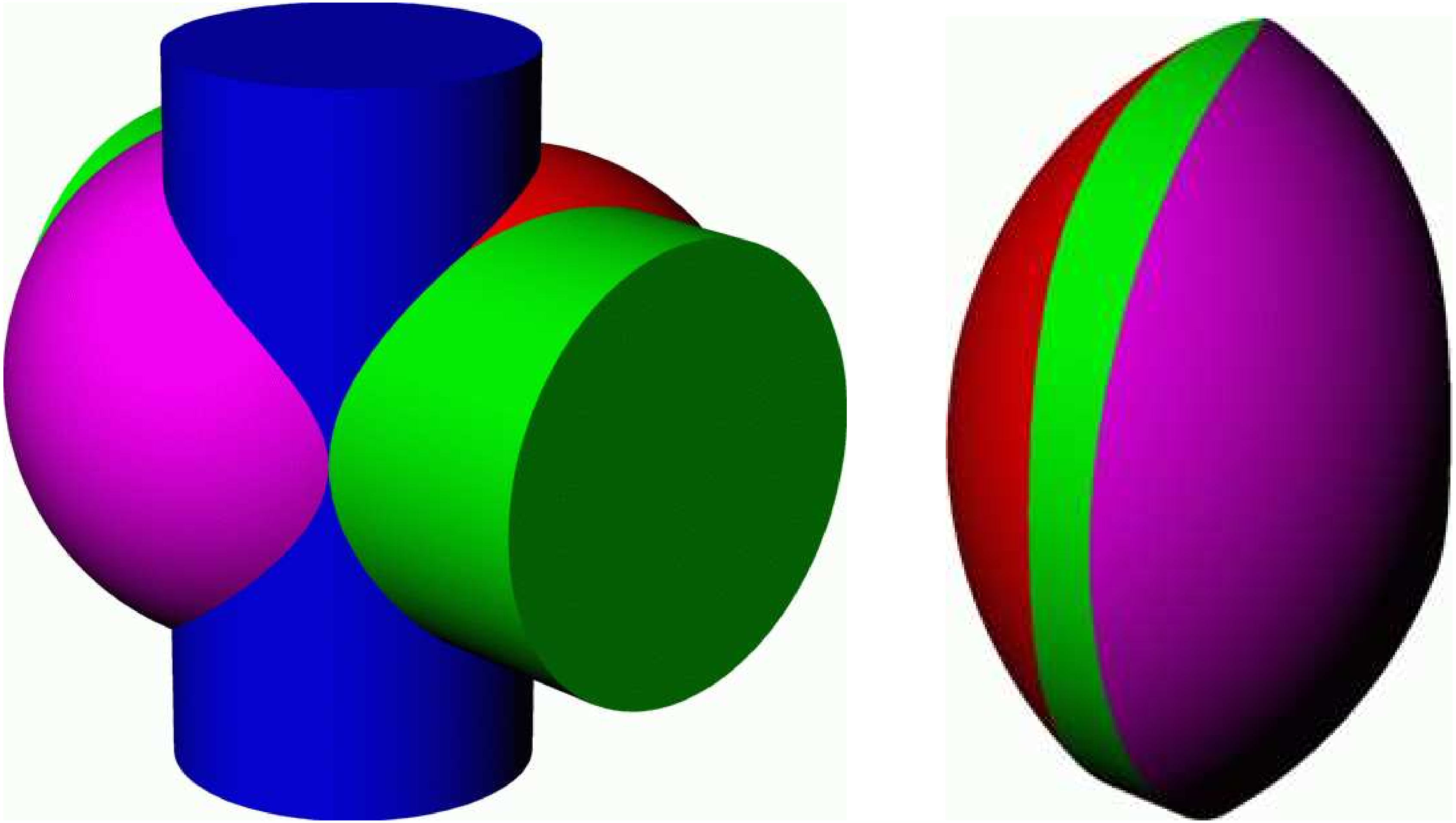


**Figure 2:** Three MCB cycles on a KNNG of a point cloud: trivial (blue) and non-trivial (red and green). The first should be closed and the latter two not.

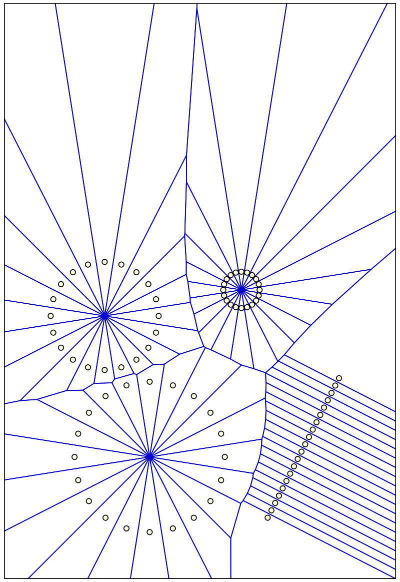


# Geometric Computing II



# Pitfalls of Geometric Computing

- algs are designed for a Real-RAM, a machine which can compute with real numbers in the sense of mathematics (basic arithmetic, square-roots, roots of polynomials, sine, . . . )
- and for non-degenerate inputs (no three collinear points, no four co-circular points)

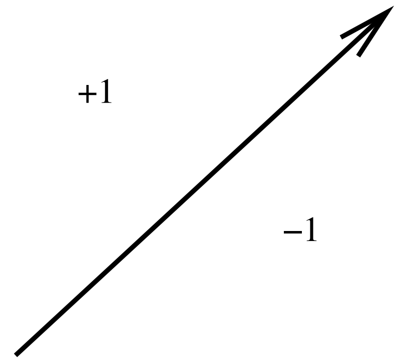


- but real machines (pun intended) have floating point and bounded integer arithmetic and
- real inputs are frequently degenerate

- as a consequence, implementing the algs of computational geometry is non-trivial enterprise, (examples from preceding slide)
- the goal of realiable and efficient implementations is still elusive
- theory (exact algs, alg numbers, . . . ) and practice (LEDA, CGAL, EXACUS) have made tremendous progress, but there is still a long way to go

# The Orientation Predicate

- three points $p$, $q$, and $r$ in the plane either lie on a common line or form a left or right turn $orient(p,q,r) = 0,\ +1,\ -1$



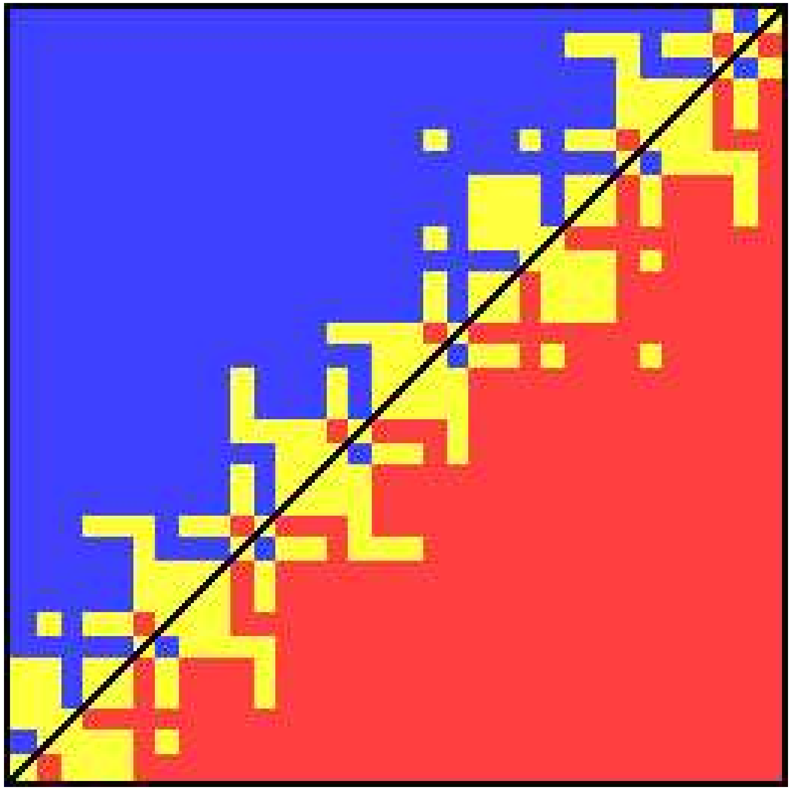

- analytically

$$orient(p,q,r) = sign(\det \begin{bmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{bmatrix})$$
$$= sign((q_x - p_x)(r_y - p_y) - (q_y - p_y)(r_x - p_x)).$$

- det is twice the signed area of the triangle $(p,q,r)$
- $float\_orient(p,q,r)$ is result of evaluating $orient(p,q,r)$ in floating point arithmetic

# Geometry of Float-Orient

- $p = (0.5, 0.5)$, $q = (12, 12)$ and $r = (24, 24)$

- 

  picture shows

  $$float\_orient((p_x + xu, p_y + yu), q, r)$$

  for $0 \le x, y \le 255$, where $u = 2^{-53}$.

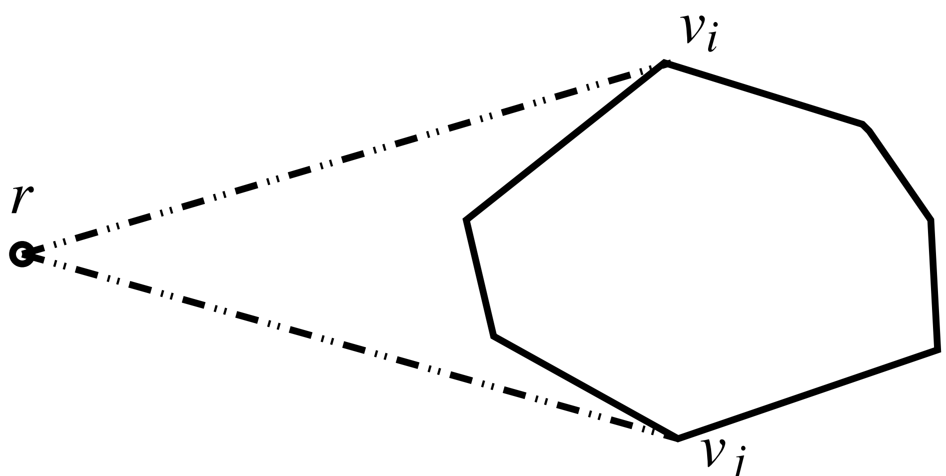
  the line $\ell(q, r)$ is shown in black

- **near the line many points are mis-classified**

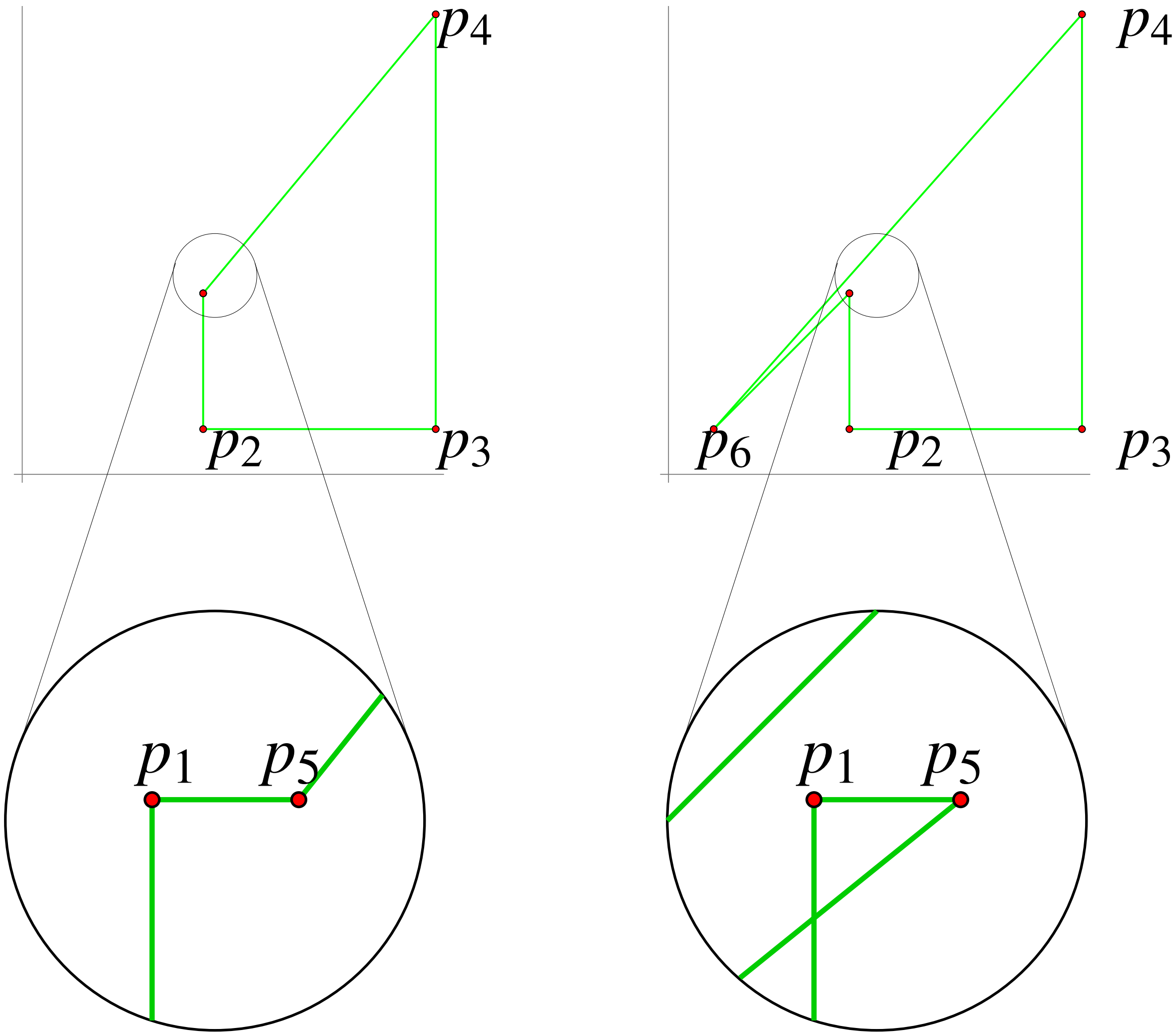
# A Simple Convex Hull Algorithm

- alg considers the points one by one, maintains vertices of current hull in counter-clockwise order

- Initialize $L$ to the counter-clockwise triangle $(a,b,c)$.
  **for all** $r \in S$ **do**
  **if** there is an edge $e$ visible from $r$ **then**
  ompute the sequence $(v_i,\ldots,v_j)$ of edges visible from $r$.
  replace the subsequence $(v_{i+1},\ldots,v_{j-1})$ by $r$.
  **end if**
  **end for**

- 

# The Effect on a Simple Convex Hull Algorithm

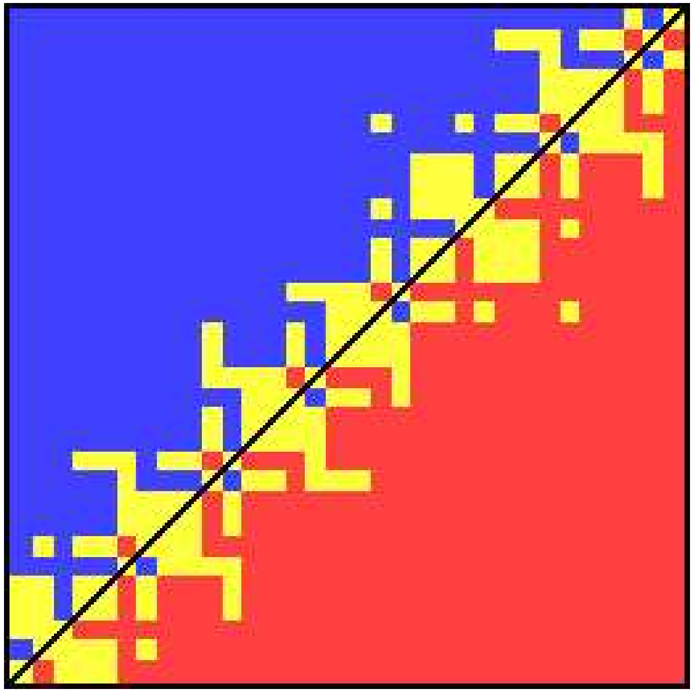



- the hull of $p_1$ to $p_4$ is correctly computed
- $p_5$ lies close to $p_1$, lies inside the hull of the first four points, but float-sees the edge $(p_1, p_4)$. The magnified schematic view below shows that we have a concave corner at $p_5$.
- point $p_6$ sees the edges $(p_1, p_2)$ and $(p_4, p_5)$, but does **not** see the edge $(p_5, p_1)$.
- we obtain either the hull shown in the figure on the right or ...

# Solutions

- The Exact Geometric Computation Paradigm (ECG)
  - implement a Real-RAM to the extent needed in computational geometry
  - the challenge is an efficient realization
  - not the subject of today's talk
- Approximation
  - compute the correct result for a slightly perturbed input
  - Controlled Perturbation
    - actively choose the perturbed input, so that the problem becomes simpler
    - initiated by Danny Halperin and co-workers
    - refined and generalized by us
    - message of the day: controlled perturbation works for a large class of geometric algorithms: predicates of bounded arity and decision trees of depth depending only on $n$

# Geometry of Float-Orient

- 

- picture shows

  $$float\_orient((p_x + xu, p_y + yu), q, r)$$

  for $0 \leq x, y \leq 255$, where $u = 2^{-53}$.

  the line $\ell(q, r)$ is shown in black
- near the line many points are mis-classified

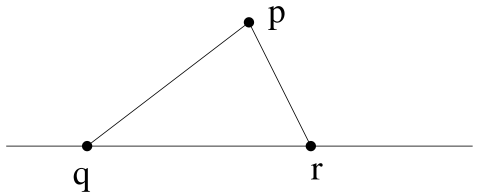
- **outside a narrow strip around the curve of degeneracy, points are classified correctly !!!**
  - how narrow is narrow?
  - true for all geometric predicates?
  - if true, can we exploit to design reliable algorithms

# How Narrow is Narrow?

- $orient(p,q,r) = sign((q_x - p_x)(r_y - p_y) - (q_y - p_y)(r_x - p_x)) = sign(E)$
- $E = 2\cdot$ signed area $\Delta$ of the triangle $(p,q,r)$
- if coordinates are bounded by $M$, maximal error in evaluating $E$ with floating point arithmetic with mantissa length $p$ is $28 \cdot M^2 \cdot 2^{-p}$
  - deal with numbers as large as $4M^2$
  - error in a single operation is at most $4M^2 2^{-p}$
  - 7 accounts for the number of operations

# How Narrow is Narrow?

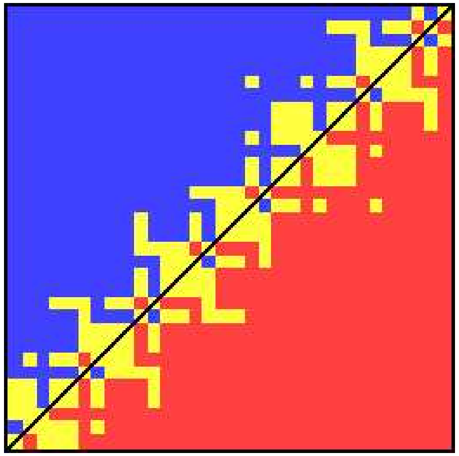
- $orient(p,q,r) = sign((q_x - p_x)(r_y - p_y) - (q_y - p_y)(r_x - p_x)) = sign(E)$
- $E = 2\cdot$ signed area $\Delta$ of the triangle $(p,q,r)$
- if coordinates are bounded by $M$, maximal error in evaluating $E$ with floating point arithmetic with mantissa length $p$ is $28 \cdot M^2 \cdot 2^{-p}$
- if $2|\Delta| > 28 \cdot M^2 \cdot 2^{-p}$, *float_orient* gives the correct result

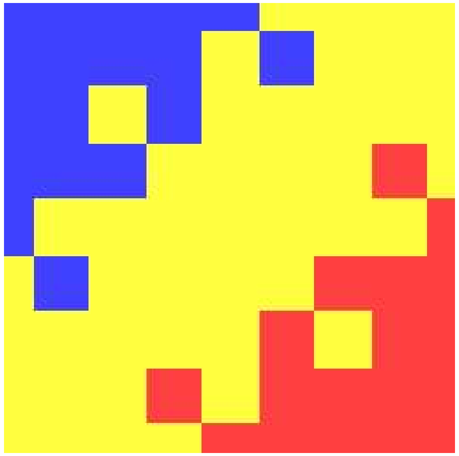



- $|\Delta| = (1/2) dist(q,r) \cdot dist(\ell(q,r),p)$

# How Narrow is Narrow?

- $orient(p,q,r) = sign((q_x - p_x)(r_y - p_y) - (q_y - p_y)(r_x - p_x)) = sign(E)$
- $E = 2\cdot$ signed area $\Delta$ of the triangle $(p,q,r)$
- if coordinates are bounded by $M$, maximal error in evaluating $E$ with floating point arithmetic with mantissa length $p$ is $28 \cdot M^2 \cdot 2^{-p}$
- Punch Line: if
  $dist(\ell((q,r),p)) \geq 28 \cdot M^2 \cdot 2^{-p} / dist(q,r)$,
  $float\_orient(p,q,r)$ gives the correct result.





on the right, $q$ and $r$ have one third the distance than in figure on the left

# How Narrow is Narrow?

- $orient(p,q,r) = sign((q_x - p_x)(r_y - p_y) - (q_y - p_y)(r_x - p_x)) = sign(E)$
- $E = 2\cdot$ signed area $\Delta$ of the triangle $(p,q,r)$
- if coordinates are bounded by $M$, maximal error in evaluating $E$ with floating point arithmetic with mantissa length $p$ is $28 \cdot M^2 \cdot 2^{-p}$
- forbidden region for $p$ = a strip of half-width $28 \cdot M^2 \cdot 2^{-p} / dist(q,r)$ about $\ell(q,r)$
- if $p$ lies outside the forbidden region, the evaluation of $orient(p,q,r)$ is floating-point safe (f-safe)

# Controlled Pertubation I

- consider algorithms using only the orientation predicate
- input points $q_1, \ldots, q_n$: perturb into $p_1, \ldots, p_n$ such that all evaluations for the perturbed points are f-safe.

# Controlled Pertubation I

MAX-PLANCK-GESELLSCHAFT

- consider algorithms using only the orientation predicate
- input points $q_1, \ldots, q_n$: perturb into $p_1, \ldots, p_n$ such that all evaluations for the perturbed points are f-safe.
- assume $p_1$ to $p_{n-1}$ are already determined: choose $p_n$ in a circle of radius $\delta$ about $q_n$ such that $p_n$ lies outside all strips of half-width $28 \cdot M^2 \cdot 2^{-p}/dist(p_i, p_j)$ about $\ell(p_i, p_j)$ for $1 \le i < j \le n-1$

# Controlled Pertubation I

- consider algorithms using only the orientation predicate
- input points $q_1, \ldots, q_n$: perturb into $p_1, \ldots, p_n$ such that all evaluations for the perturbed points are f-safe.
- assume $p_1$ to $p_{n-1}$ are already determined: choose $p_n$ in a circle of radius $\delta$ about $q_n$ such that $p_n$ lies outside all strips of half-width $28 \cdot M^2 \cdot 2^{-p}/dist(p_i, p_j)$ about $\ell(p_i, p_j)$ for $1 \leq i < j \leq n-1$
- HUCH: strips can be arbitrarily wide
- IDEA: also guarantee $dist(p_i, p_j) > \gamma$ for some $\gamma$
- then size of forbidden region $\leq n \cdot \pi \cdot \gamma^2 + n^2 \cdot (28 \cdot M^2 \cdot 2^{-p}/\gamma) \cdot 2 \cdot \pi \cdot \delta$

# Controlled Pertubation I

- consider algorithms using only the orientation predicate
- input points $q_1, \ldots, q_n$: perturb into $p_1, \ldots, p_n$ such that all evaluations for the perturbed points are f-safe.
- assume $p_1$ to $p_{n-1}$ are already determined: choose $p_n$ in a circle of radius $\delta$ about $q_n$ such that $p_n$ lies outside all strips of half-width $28 \cdot M^2 \cdot 2^{-p} / dist(p_i, p_j)$ about $\ell(p_i, p_j)$ for $1 \leq i < j \leq n-1$
- IDEA: also guarantee $dist(p_i, p_j) > \gamma$ for some $\gamma$
- then size of forbidden region $\leq n \cdot \pi \cdot \gamma^2 + n^2 \cdot (28 \cdot M^2 \cdot 2^{-p} / \gamma) \cdot 2 \cdot \pi \cdot \delta$
- min for $\gamma = (n \cdot 56 \cdot M^2 \cdot 2^{-p} \cdot \delta)^{1/3}$,
  size of FR $= 2\pi \cdot n^{5/3} \cdot (56 M^2 2^{-p} \delta)^{2/3}$

# Controlled Pertubation I

- consider algorithms using only the orientation predicate
- input points $q_1, \ldots, q_n$: perturb into $p_1, \ldots, p_n$ such that all evaluations for the perturbed points are f-safe.
- assume $p_1$ to $p_{n-1}$ are already determined: choose $p_n$ in a circle of radius $\delta$ about $q_n$ such that $p_n$ lies outside all strips of half-width $28 \cdot M^2 \cdot 2^{-p} / dist(p_i, p_j)$ about $\ell(p_i, p_j)$ for $1 \le i < j \le n-1$
- IDEA: also guarantee $dist(p_i, p_j) > \gamma$ for some $\gamma$
- then size of forbidden region $\le n \cdot \pi \cdot \gamma^2 + n^2 \cdot (28 \cdot M^2 \cdot 2^{-p} / \gamma) \cdot 2 \cdot \pi \cdot \delta$
- min for $\gamma = (n \cdot 56 \cdot M^2 \cdot 2^{-p} \cdot \delta)^{1/3}$, size of FR $= 2\pi \cdot n^{5/3} \cdot (56 M^2 2^{-p} \delta)^{2/3}$
- want: size of FR $\le \pi \cdot \delta^2 / (2n)$
- why $\ldots/(2n)$ then total prob of failure less than $1/2$

# Controlled Pertubation I

MAX-PLANCK-GESELLSCHAFT

- consider algorithms using only the orientation predicate
- input points $q_1, \ldots, q_n$: perturb into $p_1, \ldots, p_n$ such that all evaluations for the perturbed points are f-safe.
- assume $p_1$ to $p_{n-1}$ are already determined: choose $p_n$ in a circle of radius $\delta$ about $q_n$ such that $p_n$ lies outside all strips of half-width $28 \cdot M^2 \cdot 2^{-p} / dist(p_i, p_j)$ about $\ell(p_i, p_j)$ for $1 \leq i < j \leq n-1$
- IDEA: also guarantee $dist(p_i, p_j) > \gamma$ for some $\gamma$
- then size of forbidden region $\leq n \cdot \pi \cdot \gamma^2 + n^2 \cdot (28 \cdot M^2 \cdot 2^{-p} / \gamma) \cdot 2 \cdot \pi \cdot \delta$
- min for $\gamma = (n \cdot 56 \cdot M^2 \cdot 2^{-p} \cdot \delta)^{1/3}$, size of FR $= 2\pi \cdot n^{5/3} \cdot (56 M^2 2^{-p} \delta)^{2/3}$
- want: size of FR $\leq \pi \cdot \delta^2 / (2n)$
- why $\ldots/(2n)$ then total prob of failure less than $1/2$
- **Punch Line: any $p \geq 2\log(M/\delta) + 4\log n + 9$ works**

$M = 1000,\ \delta = 0.001,\ n = 1000,\ p \geq 2 \cdot 20 + 4 \cdot 10 + 9 = 89$

# Converting a Program to Controlled Perturbation

MAX-PLANCK-GESELLSCHAFT

- guard every predicate evaluation, i.e.,

  replace branch on sign of $E$ by

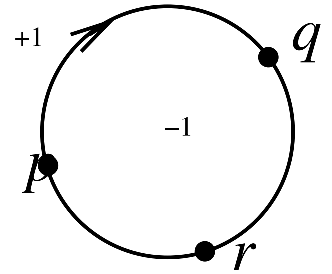
  if $(|E| \leq$ max error in evaluation of $E)$ stop;
  branch on sign of $E$

- and then run the following master program
  - initialize $\delta$ and $p$ to convenient values
  - loop
    - perturb input
    - run the guarded algorithm with floating point precision $p$
    - if the program fails, double $p$ and rerun

- theory tells us that program is guaranteed to terminate with prob $\geq 1/2$ whenever $p \geq 2\log(M/\delta) + 4\log n + 9$
- estimate is pessimistic: smaller $p$ works in practice.
- program solves problem for a perturbed input, not for the original input

# Side of Oriented Circle

- can we analyse other predicates in the same way?

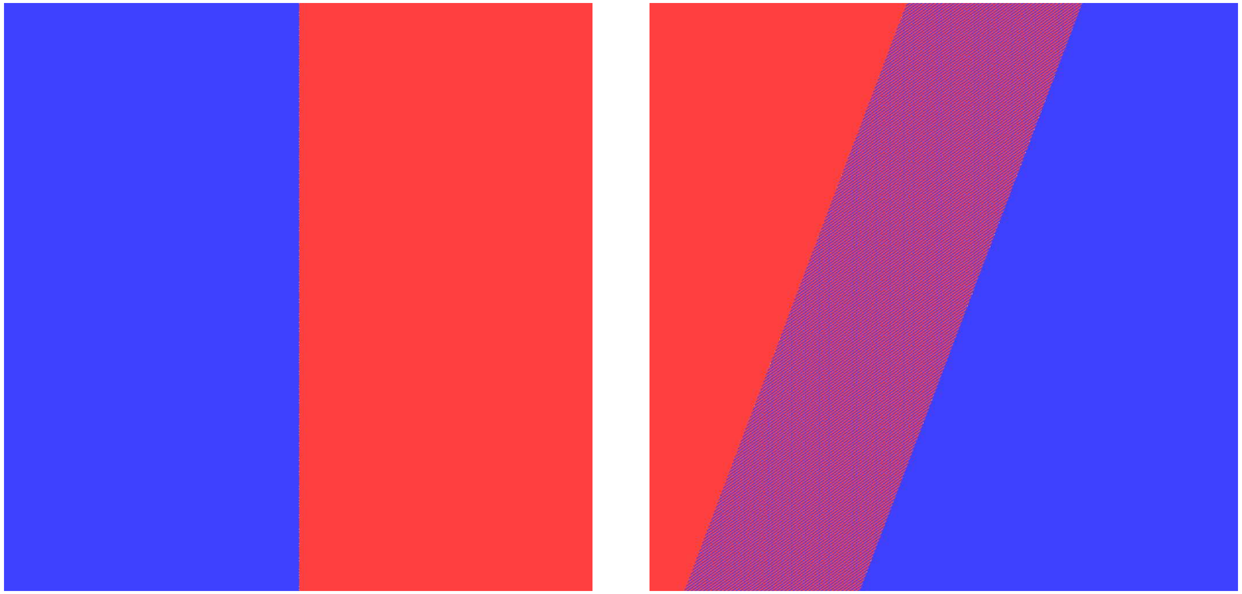



- $side\_of\_circle(p,q,r,s) = +1, 0, -1$ if $s$ lies left of , on, right of oriented circle $C(p,q,r)$

- analytically: $side\_of\_circle(p,q,r,s) = \text{sign} \begin{vmatrix} 1 & x_1 & y_1 & x_1^2 + y_1^2 \\ 1 & x_2 & y_2 & x_2^2 + y_2^2 \\ 1 & x_3 & y_3 & x_3^2 + y_3^2 \\ 1 & x & y & x^2 + y^2 \end{vmatrix}$

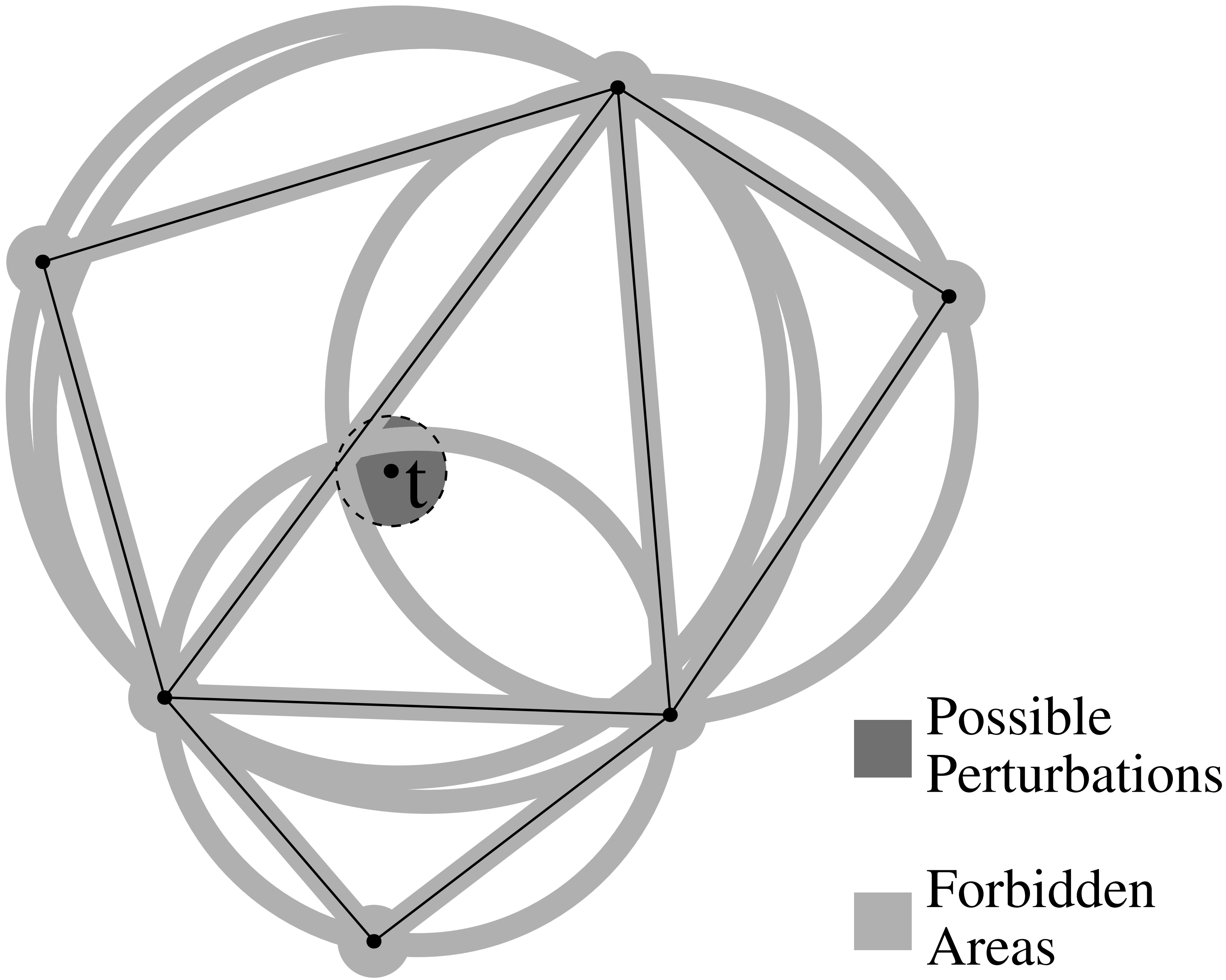
- $det = 2 \cdot \Delta \cdot (R + dist(C,s)) \cdot dist(C,s)$ and hence
- $|det| \geq 2 \cdot \Delta \cdot R \cdot dist(C,s)$
- max error in f-evaluation $= 40 \cdot M^4 \cdot 2^{-p}$
- f-eval is correct if $s$ lies outside an annulus of half-width $40 \cdot M^4 \cdot 2^{-p} / (2 \cdot \Delta \cdot R)$

# Visualization: Side of Circle



you see a circle of radius one probed on a $512 \times 512$ grid. In the figure on the right, the area of the defining triangle is about $0.001$, in the figure, on the left, the defining triangle has area about $1$.

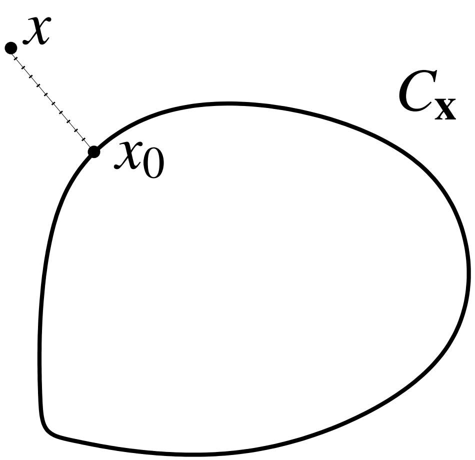
# A Visualization of Controlled Perturbation



# Generalization to All (??) Geometric Predicates

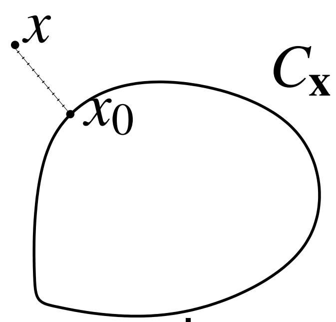
| general | orientation |
|---|---|
| predicate $P(x_1,\ldots,x_k) = \text{sign} f(x_1,\ldots,x_k)$ | $orient(p,q,r)$ |
| $x_1$ to $x_k$ points (in the plane) | |
| $\mathbf{x} = (x_1,\ldots,x_{k-1})$ fixed, $x = x_k$ variable | $q$, $r$ fixed, $p$ variable |
| $C_{\mathbf{x}} = \{\, x;\ f(\mathbf{x},x) = 0\,\}$, curve of degeneracy | $C = \{\, p;\ orient(p,q,r) = 0\,\}$ |
| $C_{\mathbf{x}}$ is either the entire plane or a smooth curve | plane or $\ell(q,r)$ |

Punch Line: Geometric predicates measure distance from curve of degeneracy and therefore forbidden region is a tubular neighborhood of this curve.



# Generalization II

| general | orientation |
| --- | --- |
| predicate $P(x_1,\ldots,x_k) = \operatorname{sign} f(x_1,\ldots,x_k)$ | $orient(p,q,r)$ |
| $\mathbf{x} = (x_1,\ldots,x_{k-1})$ fixed, $x = x_k$ variable | $q$, $r$ fixed, $p$ variable |
| $C_{\mathbf{x}} = \{\, x;\ f(\mathbf{x},x) = 0 \,\}$, curve of degeneracy | $C = \{\, p;\ orient(p,q,r) = 0 \,\}$ |




concentrate on regular $\mathbf{x}$ and arbitrary $x$, let $x_0$ be point closest to $x$ on $C_{\mathbf{x}}$ and define

$$h(d) = 2 \cdot dist(q,r) \cdot d$$

$$h(d) = f\left(\mathbf{x}, x_0 + d\frac{x - x_0}{||x - x_0||}\right)$$

$$h(dist(x, C_{\mathbf{x}})) = f(\mathbf{x}, x)$$

$h(d) \approx c \cdot d^k$ for some small $k$ (usually one), $d$ small, and $c$ depending on $\mathbf{x}$

$$f(\mathbf{x},x) \approx c_{\mathbf{x}} \cdot dist(x, C_{\mathbf{x}})^k$$

$$orient(p,q,r) = 2dist(q,r)dist(p,\ell(q,r))$$

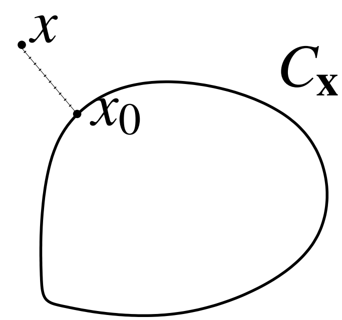
# Generalization III

| general | orientation |
|---|---|
| predicate $P(x_1,\ldots,x_k) = \operatorname{sign} f(x_1,\ldots,x_k)$ | $orient(p,q,r)$ |
| $C_{\mathbf{x}} = \{x;\ f(\mathbf{x},x) = 0\}$, curve of degeneracy | $C = \{p;\ orient(p,q,r) = 0\}$ |
| $f(\mathbf{x},x) \approx c_{\mathbf{x}} \cdot dist(x,C_{\mathbf{x}})^k$ | $orient(p,q,r) = 2dist(q,r)dist(p,\ell(q,r)$ |
| maximal error in evaluating $f$ is $c_f \cdot M^a \cdot 2^{-p}$ | max error = $28 \cdot M^2 \cdot 2^{-p}$ |
| if $c_{\mathbf{x}} \cdot dist(x,C_{\mathbf{x}})^k \geq c_f \cdot M^a \cdot 2^{-p}$, f-eval is correct | if $2 \cdot dist(q,r) \cdot dist(p,\ell(q,r)) > 28 \cdot M^2 \cdot$ $\cdots$ |
| use recursive argument to bound $c_{\mathbf{x}}$ from below | bound $dist(q,r)$ from below |

controlled perturbation works for a large class of geometric algorithms: predicates of bounded number of arguments and decision trees of depth depending only on $n$.

# Generalization II, revisited

- predicate $P(x_1,\ldots,x_k) = \operatorname{sign} f(x_1,\ldots,x_k)$




- $C_{\mathbf{x}} = \{x;\, f(\mathbf{x},x) = 0\}$, curve of degeneracy
- let $x_0$ be point closest to $x$ on $C_{\mathbf{x}}$ and assume that normal at $x_0$ exists.
- then $x$ is in direction of curve normal $\nabla f = \begin{pmatrix} \partial f/\partial x \\ \partial f/\partial y \end{pmatrix}$ at $x_0$ and

$$f(\mathbf{x},x) \approx f(\mathbf{x},x_0) + (\nabla f)(\mathbf{x},x_0)\cdot \mathit{dist}(x,x_0) = (\nabla f)(\mathbf{x},x_0)\cdot \mathit{dist}(x,x_0)$$

- if $C_{\mathbf{x}}$ has no singularity,

$$c_{\mathbf{x}} := \min_{x\in C_{\mathbf{x}}} ||(\nabla f)(\mathbf{x},x_0)|| > 0$$

$$f(\mathbf{x},x) \approx c_{\mathbf{x}}\cdot \mathit{dist}(x,C_{\mathbf{x}})$$

# Summary

- controlled perturbation works for a large class of geometric algorithms:
  - predicates of bounded arity
  - decision trees of depth depending only on number of points in input, but not on actual coordinates
- algs in the class: Delaunay, Voronoi, Arrangements, ....
- used successfully for arrangements of spheres and cycles and Delaunay diagram computations
- algs outside the class
  - Gaussian elimination
  - roots of a polynomial by iterative method

# Summary

- good predicates have $k = 1$ in $f(\mathbf{x}, x) \approx c_{\mathbf{x}} \cdot dist(x, C_{\mathbf{x}})^k$
- a guideline for designing good predicates
- it works: Delaunay triangulations, arrangements of circles and spheres,

# Open Problems

- good evaluation schemes for predicates, e.g., Clarkson's work on determinants and Fortunes's work for orientation. Recall that we only want the sign and not the value.
- good versus bad formulas for the same predicate
- redo the Halperin etal and Funke etal papers according to general theory
- do all predicates of the Voronoi diagrams of line segments
- explain the fine structure of the pictures
- arrangements of circular arcs, ellipsoids, ...
- implementation for Voronoi diagrams of line segments competitive to VRONI
- can we turn the general scheme into a program transformer, a Controlled-Perturbation-CGAL
- a good talk on the subject
- final version of the SODA 05 paper
- long version of the new paper
- packing arguments and number and size of forbidden regions