Controlled Perturbation

Certified geometric computing with fixed precision arithmetic

Dan Halperin

danha@tau.ac.il

Tel Aviv University

Prevailing assumptions in Comp Geo theory

- worst-case asymptotic complexity measures

Prevailing assumptions in Comp Geo theory

- worst-case asymptotic complexity measures
- unit cost of operations on a constant number of simple objects

Prevailing assumptions in Comp Geo theory

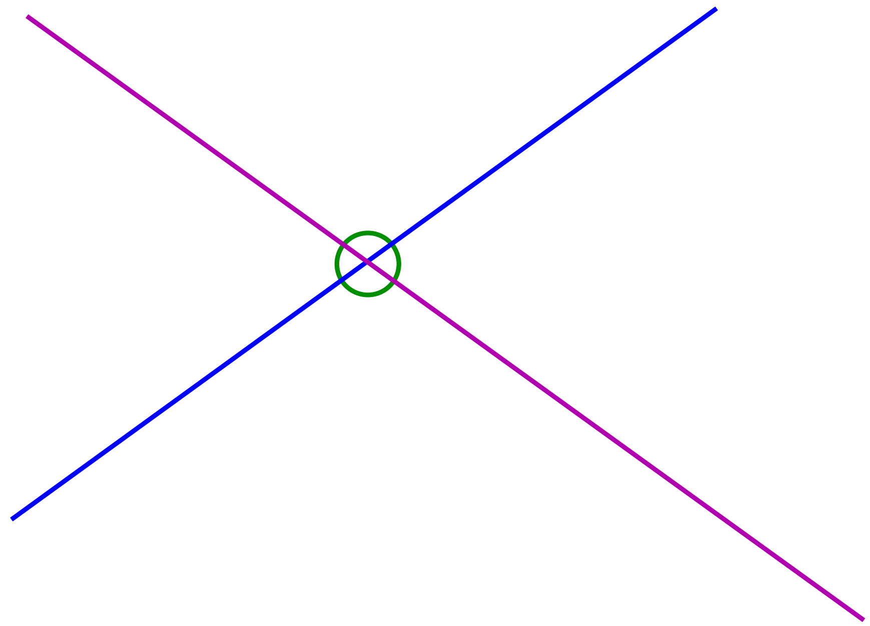
- worst-case asymptotic complexity measures
- unit cost of operations on a constant number of simple objects
- the real RAM model, infinite precision real arithmetic

Prevailing assumptions in Comp Geo theory

- worst-case asymptotic complexity measures
- unit cost of operations on a constant number of simple objects
- the real RAM model, infinite precision real arithmetic
- general position, no degeneracies

Question

given two lines ℓ_1, ℓ_2 that intersect in a single point p, does p lie on ℓ_1?

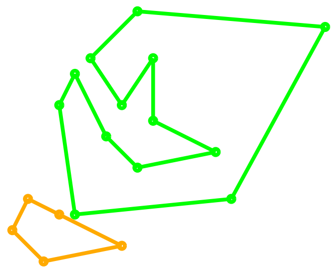


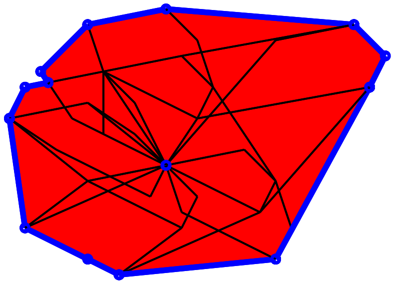
Talk outline

- background
  - robustness and precision
  - the CGAL project
  - arrangements
- controlled perturbation
  - preliminaries
  - the case of circles
  - applications
  - further directions

Robustness: the problematic assumptions

- infinite precision real arithmetic
- general position



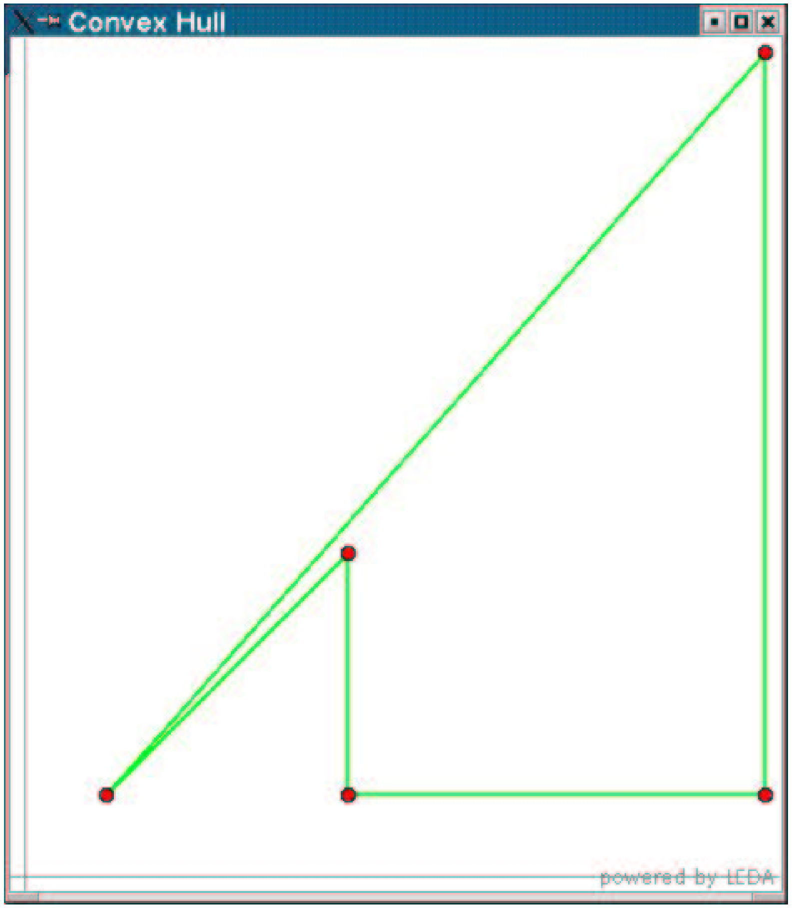


the two issues are intertwined: (near) degenerate configurations incur precision problems

geometric algorithms: interplay between numerics and combinatorics

Interplay between numerics and combinatorics, example # 1

convex hulls



[Kettner et al, ’04]

Interplay, example # 2

Delaunay triangulations

4 *Jonathan Richard Shewchuk*

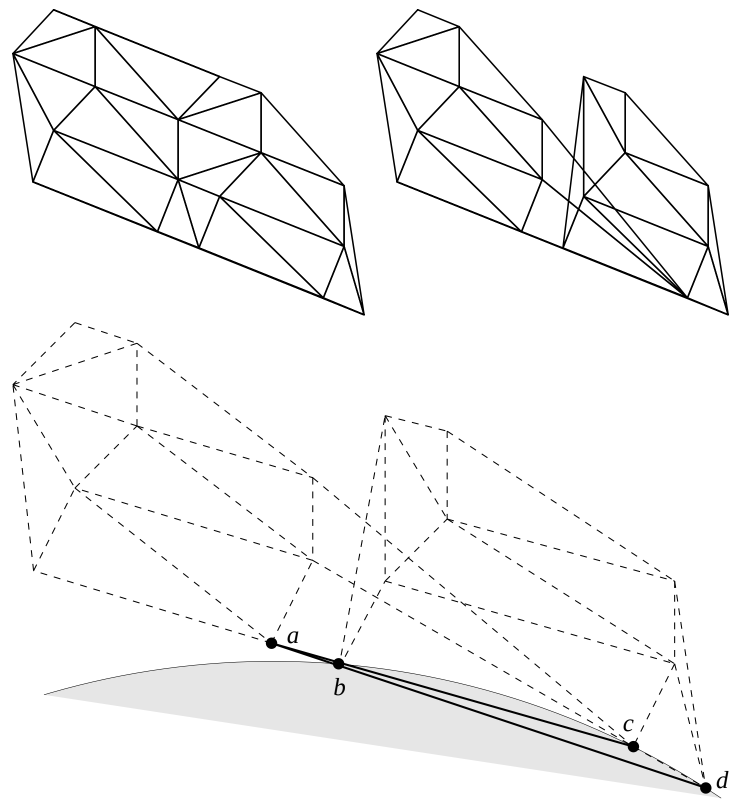


Figure 1: Top left: A Delaunay triangulation. Top right: An invalid triangulation created due to roundoff error. Bottom: Exaggerated view of the inconsistencies that led to the problem. The algorithm "knew" that the point b lay between the lines ac and ad, but an incorrect incircle test claimed that a lay inside the circle dcb.

Robustness, approaches

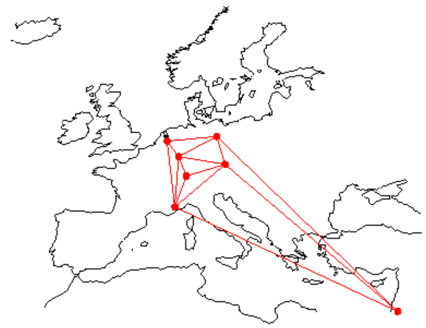
- exact computing
  [Karasick et al], [Mehlhorn et al], [Yap et al], [Brönnimann et al]; speedup: floating point filters [Fortune-Van Wyk], [Shewchuk]; symbolic perturbation schemes [Edelsbrunner-Mücke], [Yap], [Emiris-Canny-Seidel]; leave in the degeneracies [Burnikel-Mehlhorn-Schirra]; libraries CGAL, LEDA, CORE, EXACUS, . . .

- fixed precision approximation
  [Greene-Yao], [Guibas et al], [Fortune], [Milenkovic], [Sugihara et al], [H-Packer], . . .

Implementing Computational Geometry algorithms

- C++gal (INRIA), PlaGeo, SpaGeo (Utrecht), LEDA-Geometry (MPI Saarbrr̈cken), [XYZ GeoBench (Zurich), . . .]
- large effort, requires unique expertise and more research
  - 1995: the CGAL kernel
  - 1996: the official beginning of CGAL
  - 1998: GALIA, a continuation of CGAL
  - 2001: ECG
  - 2005: ACS
  - CGAL development still goes on

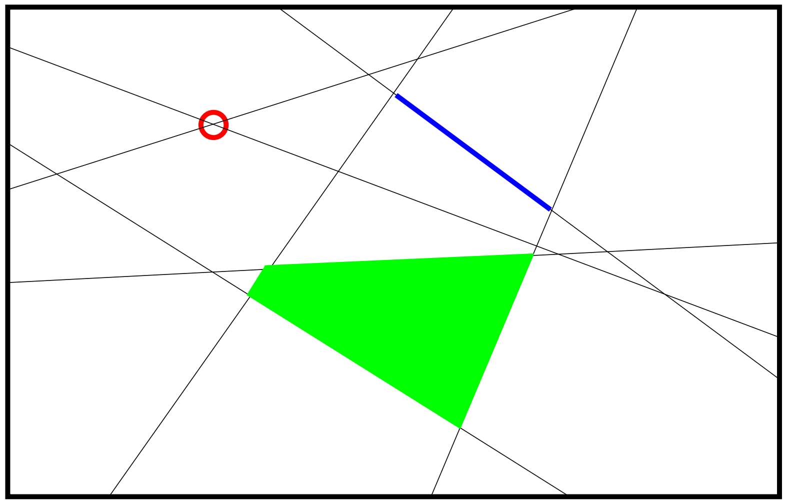
CGAL = Computational Geometry Algorithms Library



- ETH Zurich
- FU Berlin
- Trier University
- INRIA Sophia Antipolis
- MPI Saarbrücken
- Tel Aviv University
- Utrecht University

Arrangements (leitmotiv)

Example: an arrangement of lines



vertex

edge

face

Arrangements, cont'd

- an arrangement of a set S of geometric objects is the subdivision of space where the objects reside induced by S
- possibly non-linear objects (parabolas), bounded objects (segments, circles), higher dimensions (planes, simplices)
- numerous applications in robotics, molecular biology, vision, graphics, CAD/CAM, statistics, GIS
- have been studied for decades, originally mostly combinatorics

  Matoušek (2002) cites Steiner,1826

  nowadays mainly studied in combinatorial and computational geometry

Exact geometric computing: pros and cons

Exact geometric computing: pros and cons

pros

- the truth
- algorithms can be easily transcribed, up to the general position assumption (this is the CGAL approach)

Exact geometric computing: pros and cons

pros

- the truth
- algorithms can be easily transcribed, up to the general position assumption (this is the CGAL approach)

cons

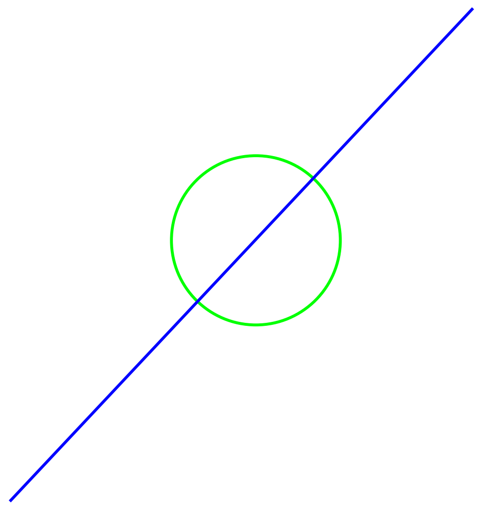
- requires special machinery (non-standard number types); available only for limited types of geometric primitives
- ever improving but still slow compared with machine arithmetic
- exact numerical output may be huge, when at all possible
- requires handling degeneracies

T1:
x= 28027/25243, y= 43613/18457, z= 14423/37273
x= 20353/2617, y= 26497/32299, z= 3673/63667
x= 55897/42403, y= 499/27767, z= 31253/10243
T2:
x= 53593/24763, y= 62501/63317, z= 11827/5693
x= 57143/65423, y= 40483/59447, z= 27739/62327
x= 57283/22027, y= 41231/45817, z= 9433/48673
T3:
x= 5693/48527, y= 11597/7757, z= 58367/44017
x= 2377/59471, y= 23831/3163, z= 57287/25343
x= 16657/46507, y= 57283/14783, z= 9437/6911

Intersection:
x (normalized rational)=
2974289381842164777454668853064172074327166148258603101344807660820042660360261669891047833223434838758837012 / 831908664650212786986421527747500810108390259862075297035835614133471677136475796012631653754008326934339751

Beyond rationals

the intersection of the line $y = x$ with the circle $x^2 + y^2 = 1$



$$(\sqrt{2}/2, \sqrt{2}/2), (-\sqrt{2}/2, -\sqrt{2}/2)$$

Exact geometric computing: pros and cons

pros

- the truth
- algorithms can be easily transcribed, up to the general position assumption (this is the CGAL approach)

cons

- requires special machinery (non-standard number types); available only for limited types of geometric primitives
- ever improving but still slow compared with machine arithmetic
- exact numerical output may be huge, when at all possible
- requires handling degeneracies

Talk outline

- background
  - robustness and precision
  - the CGAL project
  - arrangements
- controlled perturbation
  - preliminaries
  - the case of circles
  - applications
  - further directions

Controlled perturbation

input: a set $\mathcal{C}$ of geometric objects (curves or surfaces), and the floating point precision

goal: perturb $\mathcal{C}$ slightly, $\mathcal{C} \Rightarrow \mathcal{C}'$, such that

- all the predicates arising in the construction of $\mathcal{A}(\mathcal{C}')$ are computed accurately, and
- $\mathcal{A}(\mathcal{C}')$ is degeneracy free

the description here is for arrangements, but the approach is applicable more generally

Controlled perturbation: preliminaries

- a fixed precision approximation

Controlled perturbation: preliminaries

- a fixed precision approximation
- resolution bound ε, perturbation bound δ (actual perturbation)

Controlled perturbation: preliminaries

- a fixed precision approximation
- resolution bound ε, perturbation bound δ (actual perturbation)
- degeneracy := potential degeneracy

Controlled perturbation: preliminaries

- a fixed precision approximation
- resolution bound ε, perturbation bound δ (actual perturbation)
- degeneracy := potential degeneracy
- no degeneracy $\Rightarrow$ no perturbation

Controlled perturbation: preliminaries

- a fixed precision approximation
- resolution bound ε, perturbation bound δ (actual perturbation)
- degeneracy := potential degeneracy
- no degeneracy $\Rightarrow$ no perturbation
- otherwise identify and remove all degeneracies

Controlled perturbation: preliminaries

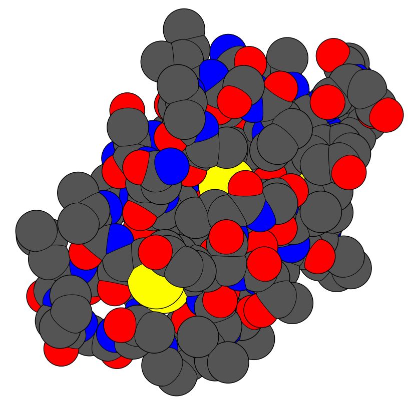
- a fixed precision approximation
- resolution bound ε, perturbation bound δ (actual perturbation)
- degeneracy := potential degeneracy
- no degeneracy $\Rightarrow$ no perturbation
- otherwise identify and remove all degeneracies
- predicates are accurately computed

Controlled perturbation: preliminaries

- a fixed precision approximation
- resolution bound ε, perturbation bound δ (actual perturbation)
- degeneracy := potential degeneracy
- no degeneracy $\Rightarrow$ no perturbation
- otherwise identify and remove all degeneracies
- predicates are accurately computed
- trade-off between perturbation magnitude and computation time

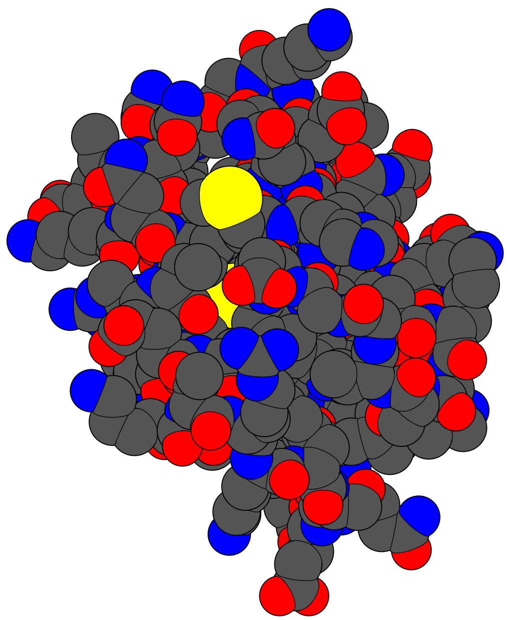
Controlled perturbation, history

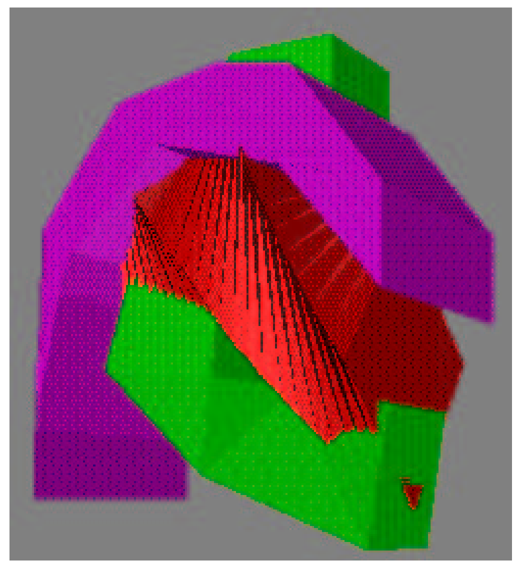
- introduction of the method, spheres in space, molecular modeling [H-Shelton ’97]
- polyhedral surfaces, swept volumes [Raab-H ’99]
- polygons [Packer ’02]
- computing the resolution bound, circles [H-Leiserowitz ’03]
- RIC algorithms, Delaunay Δs [Funke-Klein-Mehlhorn-Schmitt ’05]
- dynamic molecular surfaces [Eyal-H ’05]



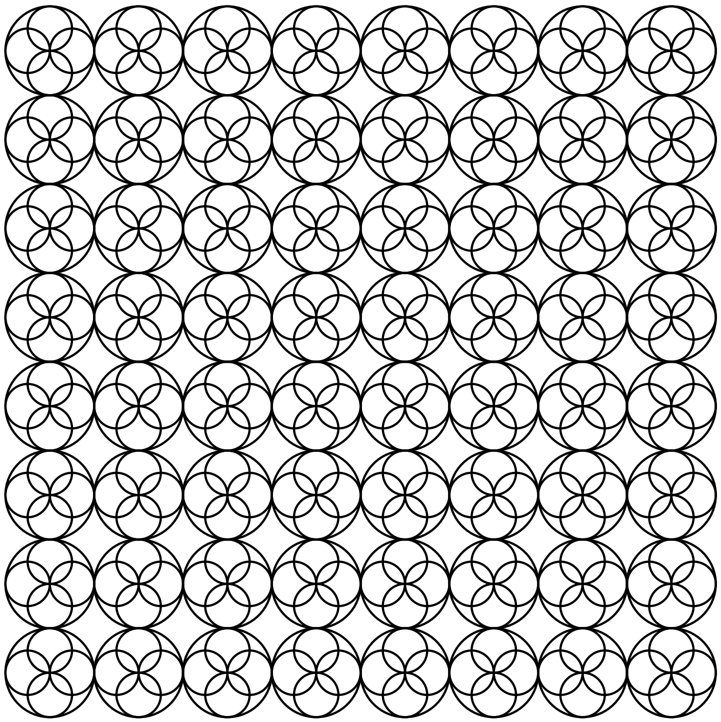
Is it OK to perturb?

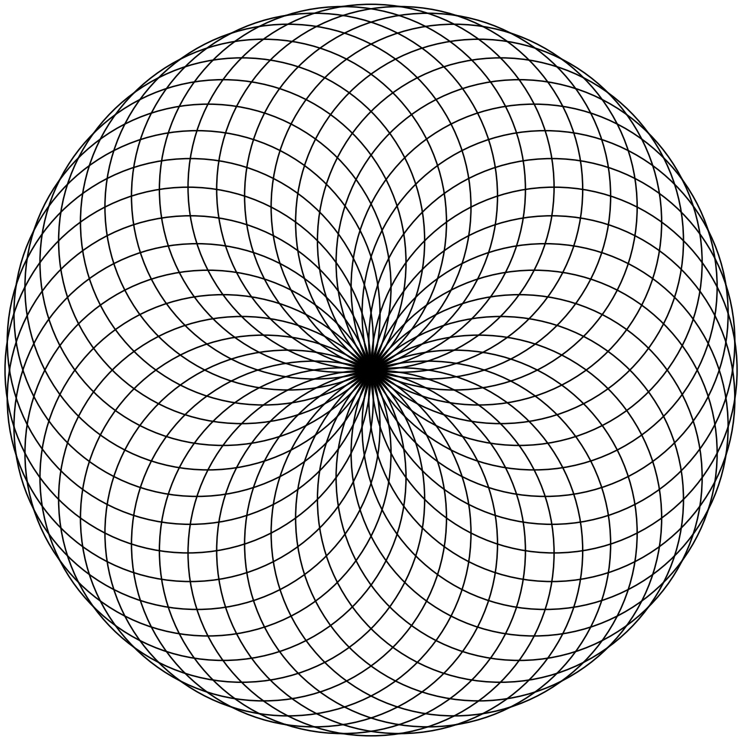
- in many scientific and industrial applications the model is approximate to begin with
- considerable slack for perturbation: typically, the maximum perturbation magnitude is well below the (in)accuracy of the model



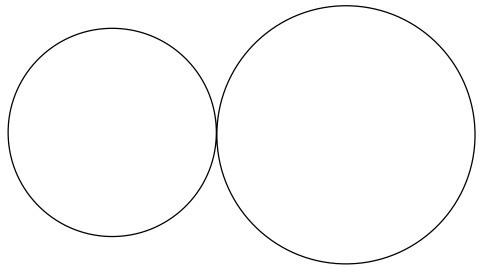


Arrangements of circles





Resolution bound, example



$$[(X_1 - X_2)^2 + (Y_1 - Y_2)^2]^{\frac{1}{2}} = R_1 + R_2$$

$$E = (X_1 - X_2)^2 + (Y_1 - Y_2)^2 - (R_1 + R_2)^2$$

outer tangency $\equiv E = 0$

the minimum distance to move a circle so that the predicate will be safely evaluated to a non-zero value (using fp)

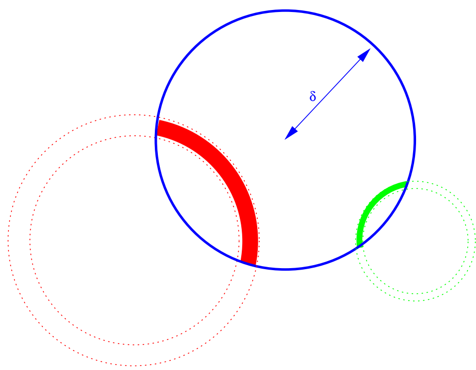
The scheme

- input: $C_1, C_2, \ldots, C_n$ by center coordinates and radii, floating-point precision p
- compute ε, δ
- handle the circles one by one, $C_i \Rightarrow C_i'$
- $\mathcal{C}_i' = \{C_1', \ldots, C_i'\}$, at the end of stage i, $\mathcal{A}(\mathcal{C}_i')$ is degeneracy free and C_i' will not be moved again
- if C_{i+1} does not induce any degeneracy with $\mathcal{C}_i'$ then $C_{i+1}' := C_{i+1}$, otherwise

Handling the current circle

given the resolution bound ε the circle C_{i+1} will be moved by at most $\delta(\varepsilon, \cdot)$ from its original position such that no two ‘features’ will be less than ε apart

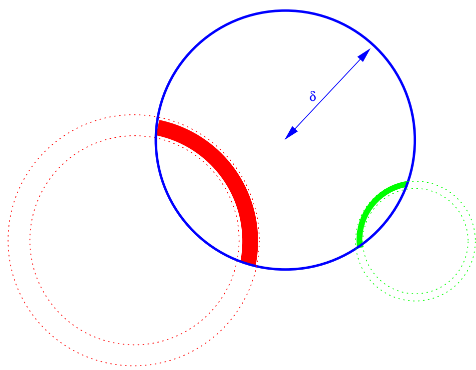
the new location of the center is chosen inside a δ-disc around the original center avoiding forbidden regions



Handling the current circle

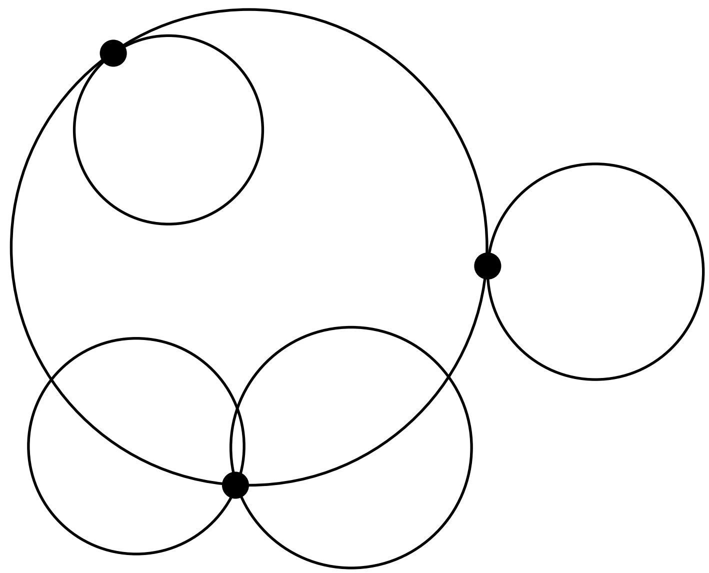
given the resolution bound ε the circle C_{i+1} will be moved by at most $\delta(\varepsilon, \cdot)$ from its original position such that no two 'features' will be less than ε apart

the new location of the center is chosen inside a δ-disc around the original center avoiding forbidden regions



the arrangement of forbidden regions is more complicated than the original arrangement, use randomization

Degeneracies in arrangements of circles

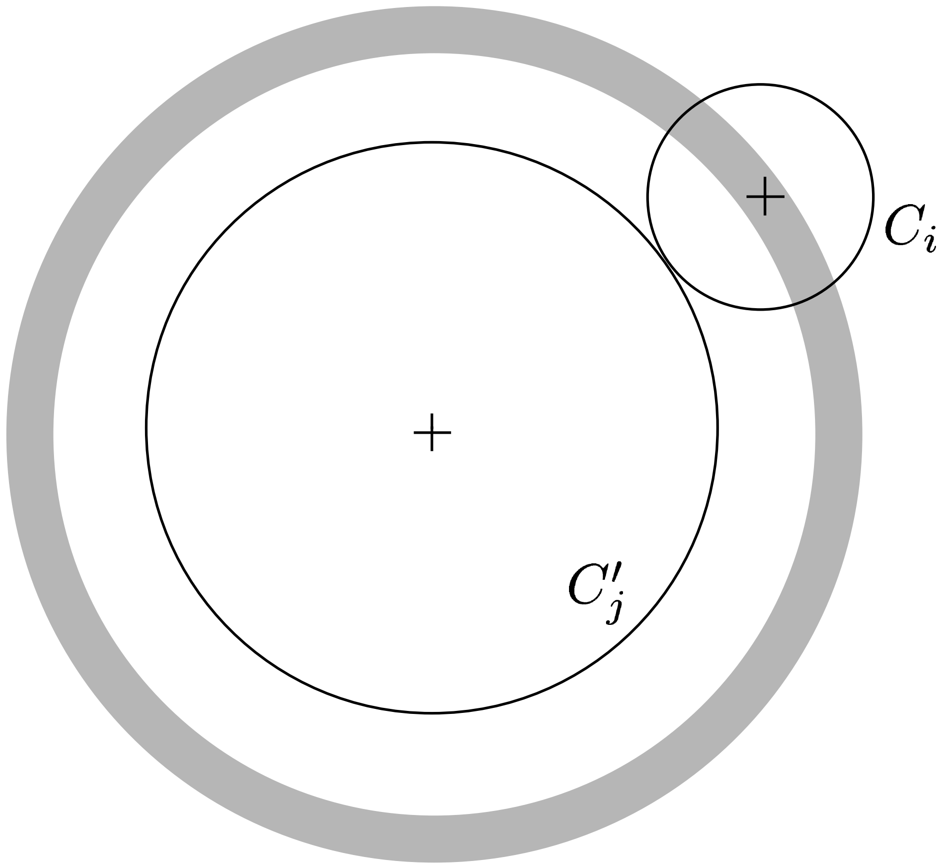


- outer tangency
- inner tangency
- three circles intersecting in a common point
- (the centers of two intersecting circles are too close)

assertion: the centers are at least some fixed minimum distance apart

Forbidden regions

the forbidden placements for the current center (of C_i) with respect to a degeneracy with already handled circles (C'_j)



Forbidden regions vs. valid placements

the forbidden volume for all degeneracies:
$VF = F_1 \bigcup F_2 \bigcup F_3 \bigcup F_4$
$VF \leq \pi\rho\varepsilon(12R + 4\rho R + \varepsilon)$
R - max radius, ρ - input density ($= n$ at worst)

for efficiency we wish that the total area of the forbidden regions will be less than half the area of the sampled region (the δ-disc)

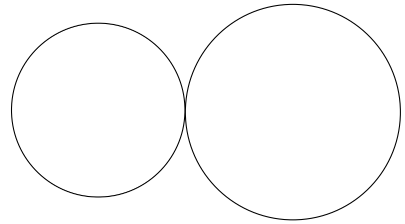
$$\pi\delta^2 > 2VF \Rightarrow \delta > \sqrt{2\rho\varepsilon(12R + 4\rho R + \varepsilon)}$$

Deriving the resolution bound, method I

outer tangency predicate (reminder)

$$E = (X_1 - X_2)^2 + (Y_1 - Y_2)^2 - (R_1 + R_2)^2$$

outer tangency $\equiv E = 0$



we use floating-point arithmetic, so we will compute

$$\tilde{E} = (X_1 \ominus X_2)^2 \oplus (Y_1 \ominus Y_2)^2 \ominus (R_1 \oplus R_2)^2$$

fp error bounds [Funke, others]:

E	$\tilde{E}$	$\tilde{E}_{sup}$	ind_E
A	A	$\|A\|$	0
$A + B$	$\tilde{A} \oplus \tilde{B}$	$\tilde{A}_{sup} \oplus \tilde{B}_{sup}$	$1 + max(ind_A, ind_B)$
$A - B$	$\tilde{A} \ominus \tilde{B}$	$\tilde{A}_{sup} \oplus \tilde{B}_{sup}$	$1 + max(ind_A, ind_B)$
$A \cdot B$	$\tilde{A} \odot \tilde{B}$	$\tilde{A}_{sup} \odot \tilde{B}_{sup}$	$1 + ind_A + ind_B$

Resolution bound, method I, cont'd

it follows that
$\tilde{E_{sup}} = (|X_1| \oplus |X_2|)^2 \oplus (|Y_1| \oplus |Y_2|)^2 \oplus (|R_1| \oplus |R_2|)^2$
$ind_E = 5$
$B = 2^{-p} \odot ind_E \odot \tilde{E_{sup}}$, where p is the mantissa length

$|E - \tilde{E}| \leq B$
if $\tilde{E} > B$ then $E > 0$, and if $\tilde{E} < -B$ then $E < 0$

a potential outer tangency between two circles C_1 and C_2

when $|\tilde{E}| \leq B$

Resolution bound, method I, cont'd

$|E - \tilde{E}| \leq B \Rightarrow$ if $|E| > 2B$ then $|\tilde{E}| > B$

$[(X_1 - X_2)^2 + (Y_1 - Y_2)^2]^{\frac{1}{2}} = R_1 + R_2 \pm \varepsilon$

after squaring both side, and rearranging terms we get:
$(X_1 - X_2)^2 + (Y_1 - Y_2)^2 - (R_1 + R_2)^2 = \pm 2(R_1 + R_2)\varepsilon + \varepsilon^2$

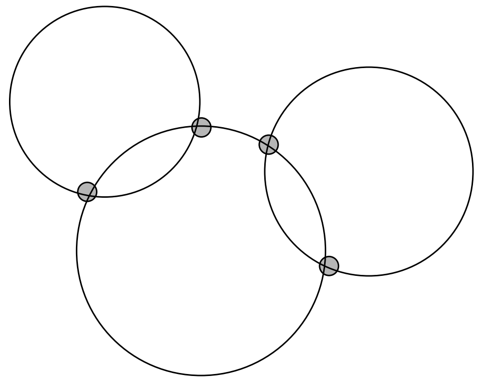
the left-hand side is exactly E, so we can rewrite our requirement, this time in terms of ε, that is

$|\pm 2(R_1 + R_2)\varepsilon + \varepsilon^2| > 2B$
M - maximum input number, p - mantissa length,
$\varepsilon_1 \equiv \varepsilon$ for outer tangency

$$\varepsilon_1 > \sqrt{10 \odot 2^{-p} \odot 12 \odot M^2}$$

Deriving the resolution bound, method II

applied to the common intersection of three circles



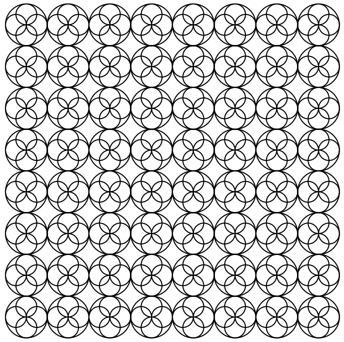
step 1: use interval arithmetic and `nextafter` to derive a bound η on the error in computing an intersection point
step 2: inflate a disk of radius η around each approximate intersection point and require that the disks are disjoint using method I

$$\varepsilon_3 > 6 \odot \eta \oplus \sqrt{(10 \odot 2^{-p} \odot (32 \odot M^2 \oplus 36 \odot \eta^2))}$$

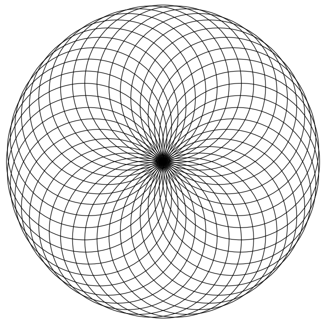
Implementation details

- intersection points sorted along each circle
- total running time $O(n^2 \log n)$
- start with $\delta_0 := 2\varepsilon$, if after a small number of guesses no valid placement found then $\delta_{i+1} := 2\delta_i$, till placement found or guaranteed δ reached
- multiplies the running time by $O(\log \frac{\delta}{\varepsilon})$

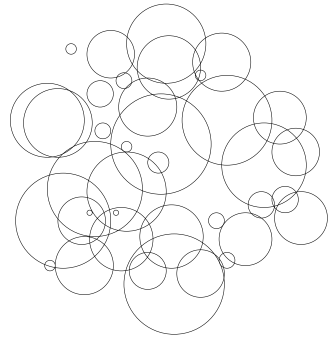
Arrangements of circles: Experiments



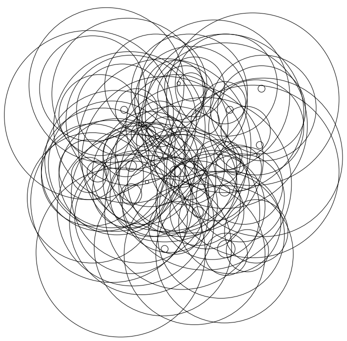
grid



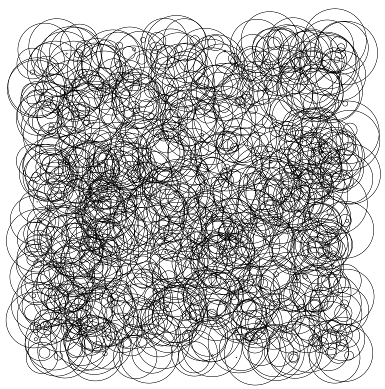
flower



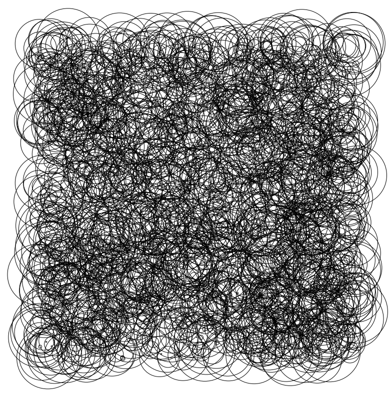
rand_sparse



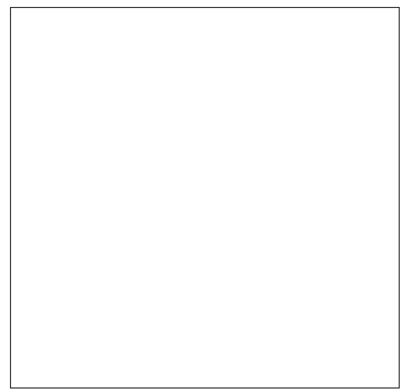
rand_100



rand_1000



rand_2000



rand_10000

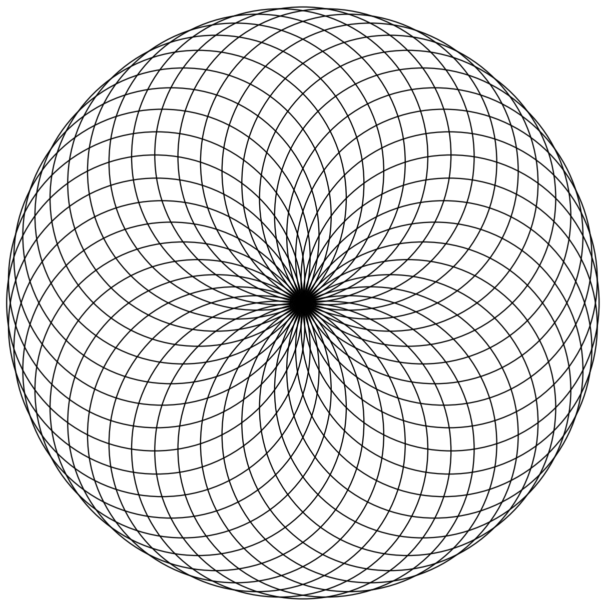
Experiments, cont'd

on an Intel Pentium III 1 GHz with 2 GB RAM (Linux Redhat 7.3, gcc 2.95.3), time in seconds

name	n	max radius	max coord	avg pert	max pert	time
grid	320	10	140	0.1122	0.6320	0.114
flower	40	100	100	0.8819	2.3360	0.132
rand_sparse	40	20	100	0.0424	0.0493	0.002
rand_100	100	49	100	0.0597	0.4017	0.130
rand_1000	1000	100	1000	0.0497	0.3994	0.556
rand_2000	2000	100	1000	0.1815	1.0856	2.804
rand_10000	10000	35	1000	0.3412	1.4527	9.478

Name	#vertices	#halfedges	#faces
rand_10000	346954	1388506	347301

Arrangements of circles, demo



Alternative view of CP

controlled perturbation moves the original input so that if the algorithm is run on the perturbed input with fixed precision floating point filter, the filter will always succeed and will never have to resort to higher precision or exact computation

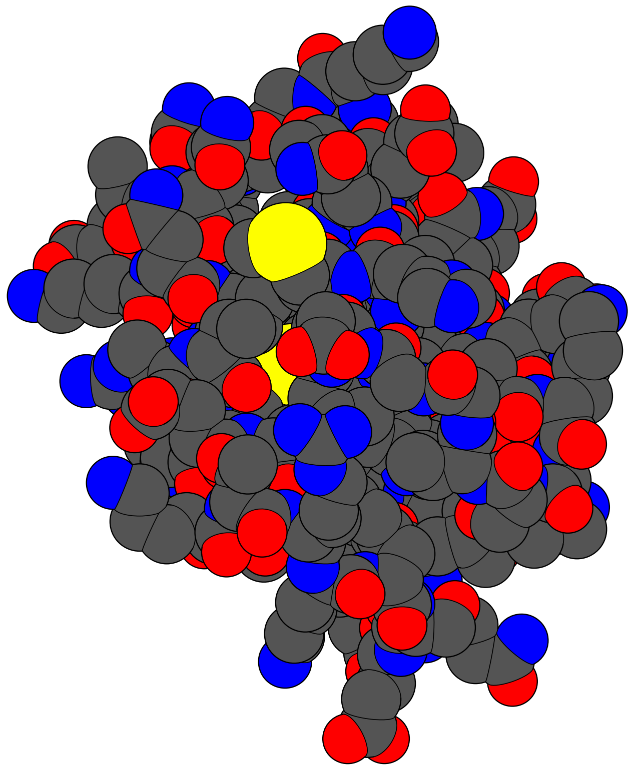
Analysis?

the method can be safely implemented with hardly any analysis, but with no guaranteed perturbation bound

with analysis:

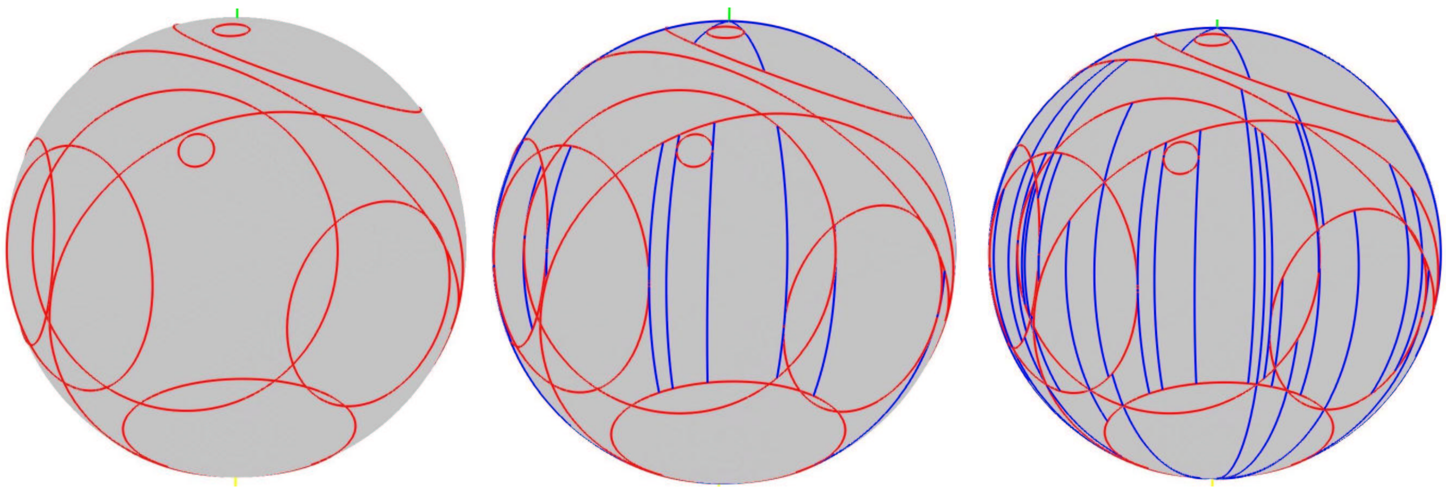
- (i) given a desired bound δ one can determine the necessary fp precision p, or
- (ii) given the desired precision, one can bound the maximum perturbation magnitude

Molecular surfaces



sparse arrangement of spheres

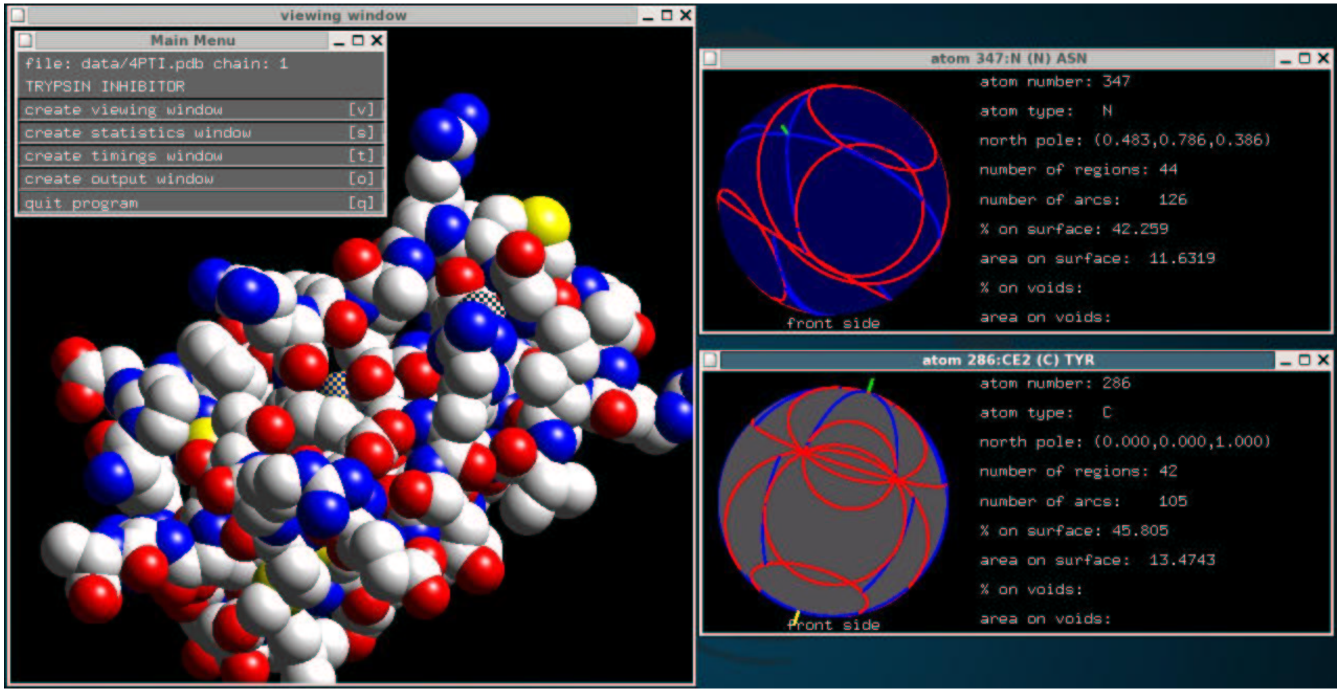
Decomposition related degeneracies



partial decomposition

choosing a unique pole direction $\Rightarrow$ tremendous computation burden

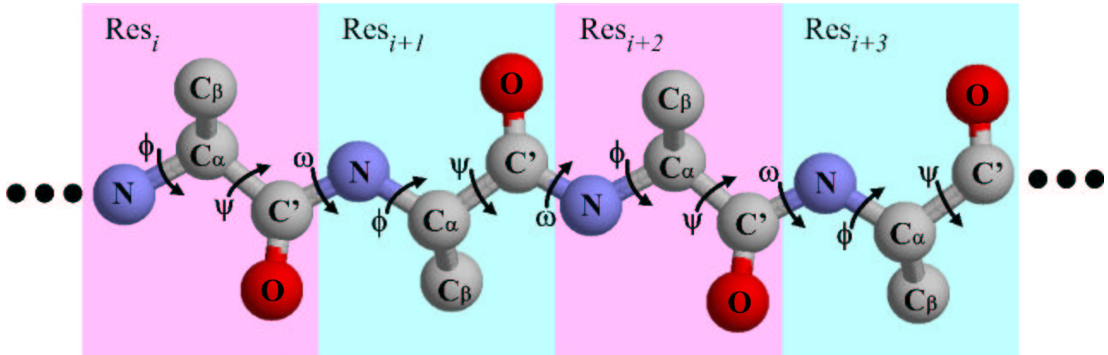
Dynamic maintenance of molecular surfaces



speedy update of the surface in Monte Carlo type simulations,

where a small number of degrees of freedom changes at a step

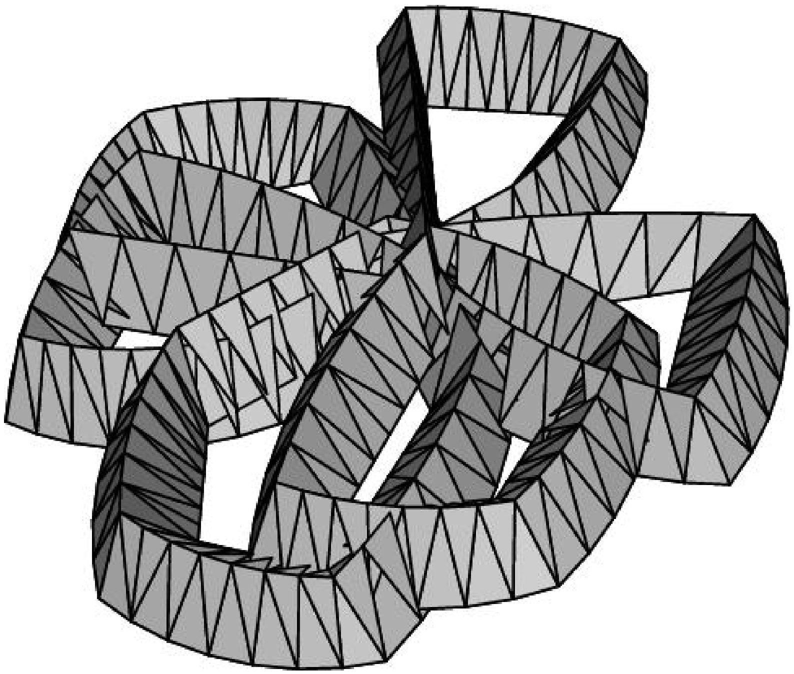
Dynamic maintenance, perturbation



- perturb as few atoms as possible at each step
- always perturb from exact placement in local frame
- avoid accumulating transformation errors by adding up the rotation angles and then computing the transformation

Polyhedral surfaces and approximate swept volumes

- no longer fixed-size basic entities
- numerous types of degeneracies



Randomized incremental construction and Delaunay triangulations

controlled perturbation is suitable for randomized incremental construction of geometric objects with little effect on the running time (under reasonable assumptions)

concrete example: planar Delaunay triangulations — no construction of new geometric objects

standard vs. lazy perturbation: standard analyzed, lazy (as we saw for circles) results in smaller perturbation

Controlled perturbation, summary

- a fixed precision approximation method, actual (not symbolic) perturbation; justified in many applications
- guarantees robustness while using floating-point arithmetic
- for circles: (i) about 40 times faster than state-of-the-art exact arithmetic, (ii) separation bound for same size input numbers requires ≈ 900 bits
- no degeneracies $\Rightarrow$ no perturbation
- otherwise removes all degeneracies (good for exact computation as well)
- trade-off between the magnitude of perturbation and the time of computation
- easy to program
- less easy to analyze: requires special analysis in each case

Further work

- additional types of arrangements and other geometric structures
- improve the bounds
- dynamic bounds
- automatic analysis

THE END