Controlled Perturbation

Certified geometric computing with fixed precision arithmetic

Dan Halperin

danha@tau.ac.il

Tel Aviv University

Exact 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

Snap rounding: pros and cons

pros

- convenient numerical output
- fairly efficient
- preserves certain topological properties

cons

- very limited range of effective applicability
- requires special machinery for preprocessing (is it necessary?)
- promotes degeneracies

The dangers of floating point arithmetic

earlier we have seen examples for

- Delaunay triangulations
- convex hulls

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

Talk overview

- introduction
- the details: arrgs of circles
- CP in molecular modelling
- polyhedral surfaces and swept volumes
- RIC algorithms, Delaunay triangulations
- summary and future directions

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

IEEE Standard 754, double precision

requires a 64 bit word, which we number from 0 to 63, left to right

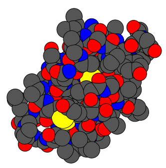
```
S EEEEEEEEEEE FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
0 1        11 12                                                63
```

the value V represented by the word is determined as follows:

- if E=2047 and F is nonzero, then V=NaN ("Not a number")
- if E=2047 and F is zero and S is 1, then V=-Infinity
- if E=2047 and F is zero and S is 0, then V=Infinity
- if 0<E<2047 then V=(-1)**S * 2 ** (E-1023) * (1.F) where "1.F" represents the binary number created by prefixing F with an implicit leading 1 and a binary point
- if E=0 and F is nonzero, then V=(-1)**S * 2 ** (-1022) * (0.F), "denormalized" values
- if E=0 and F is zero and S is 1, then V=-0
- if E=0 and F is zero and S is 0, then V=0

Controlled perturbation, history

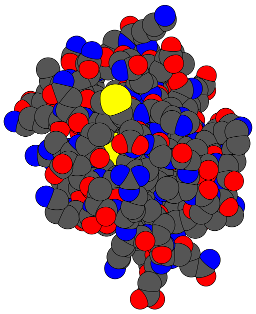
- introduction of the method, spheres in space, molecular modeling [H-Shelton '97]
- polyhedral surfaces, swept volumes [Raab-H '99]
- polygons [Packer '02]
- computing the resolution bound, circles [H-Leiserowitz '03]
- RIC algorithms, Delaunay Δs [Funke-Klein-Mehlhorn-Schmitt '05]
- dynamic molecular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



Is it OK to perturb?

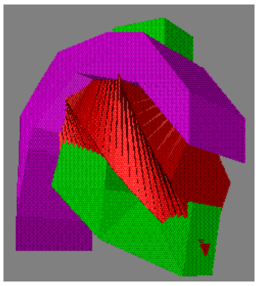
- in many scientific and industrial applications the model is approximate to begin with
- considerable slack for perturbation: typically, the maximum perturbation magnitude is well below the (in)accuracy of the model
- examples (take I) follow

Controlled perturbation, molecular surfaces



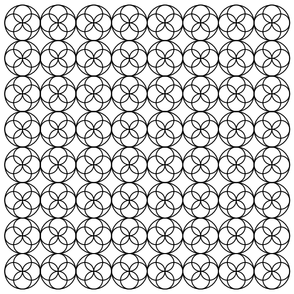
- sparse arrangement
- approximate model

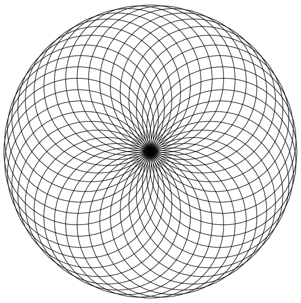
Controlled perturbation, approximate swept volumes



- arrangement of polyhedral surfaces
- approximation

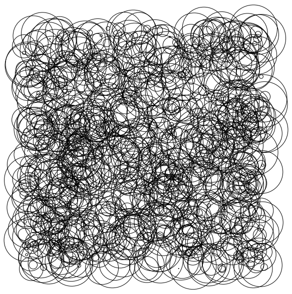
Arrangements of circles



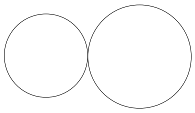


Arrangements of circles, demo

a first glimpse



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)

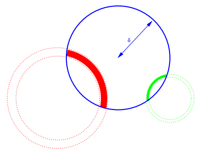
The scheme

- 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

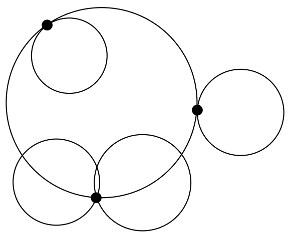
given the resolution bound ε the circle C_{i+1} will be moved by at most $\delta(\varepsilon, \cdot)$ from its original position such that no two 'features' will be less than ε apart

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



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

Degeneracies in arrangements of circles

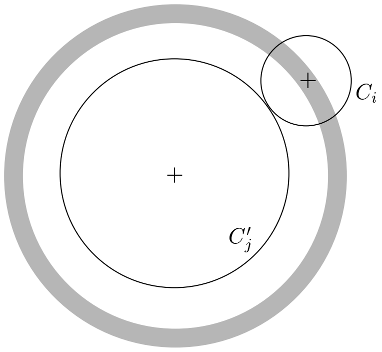


- outer tangency
- inner tangency
- three circles intersecting in a common point
- (the centers of two intersecting circles are too close)

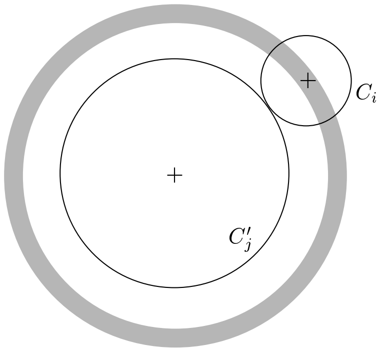
assertion: the centers are at least some fixed minimum distance apart

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 F_1: outer tangency

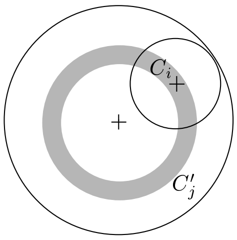


ρ denote the density parameter of $\mathcal{C}$

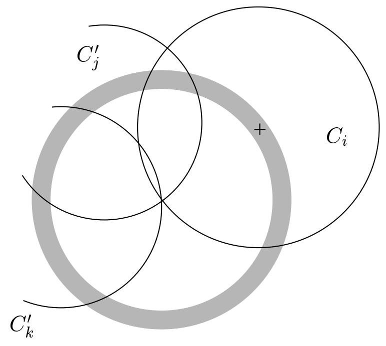
there are at most ρ circles defining the regions of F_1

$$\mathrm{Area}(F_1) \;\leq\; \rho 4\pi(R+R)\varepsilon$$

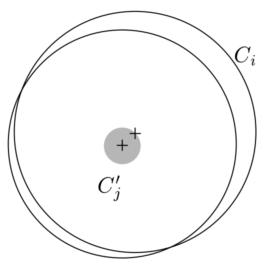
Forbidden regions F_2: inner tangency



Forbidden regions F_3: triple intersection



Forbidden regions F_4: centers coincide



Forbidden regions, summary

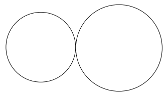
the forbidden volume for all degeneracies:
$VF = F_1 \bigcup F_2 \bigcup F_3 \bigcup F_4$
$VF \leq \pi\rho\varepsilon(12R + 4\rho R + \varepsilon)$
R - max radius, ρ - input density ($= n$ at worst)

for efficiency we wish that the total area of the forbidden regions will be less than half the area of the sampled region

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

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

the intersection of two circles

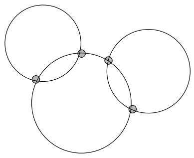
$$s = \frac{1}{2}\frac{{R_1}^2 - {R_2}^2}{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} + \frac{1}{2}$$

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

the intersection point $[x, y]$ is:

$$\begin{aligned}[x, y] &= [X_1, Y_1] + s[X_2 - X_1, Y_2 - Y_1] \\ &\pm t[Y_2 - Y_1, X_1 - X_2]\end{aligned}$$

Resolution bound, method II, cont'd



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 $O(\log \frac{\delta}{\varepsilon})$

Implementation details, outer tangency

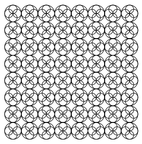
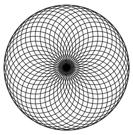
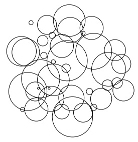
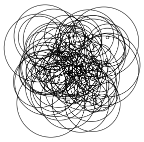
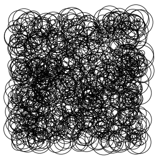
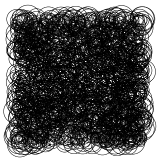
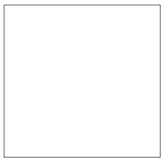
computing ε_1:

```
/* NT is the number type (the default is 'double'), machine_eps is
the machine epsilon (for 'double' it is 2^-52) and M is the maximal
input size */
NT temp = 10*machine_eps*12*M*M;
// set UP rounding mode
_controlfp(_RC_UP,_MCW_RC);
// epsilon for F_1 and F_2
NT eps1_2=sqrt(temp);
// restore normal rounding mode
_controlfp( _CW_DEFAULT, 0xfffff );
```

the predicate itself:

```
/* test for outer tangency.  temp_x, temp_y and temp_r refer to
an existing circle.  new_x, new_y and new_r refer to the newly
added circle.  */
NT E1 = fabs((temp_x-new_x)*(temp_x-new_x)+
(temp_y-new_y)*(temp_y-new_y)-(temp_r+new_r)*(temp_r+new_r));
if(E1 <= 5*machine_eps*12*M*M)
{ // a potential degeneracy exists ...  }
```


Arrangements of circles: Experiments


grid

flower

rand_sparse

rand_100

rand_1000

rand_2000

rand_10000

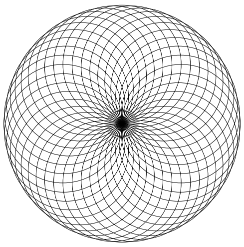
Experiments, cont'd

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

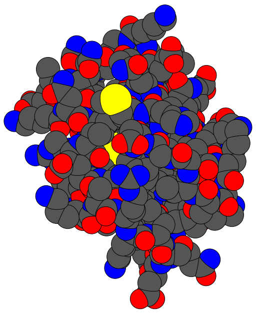


Alternative view of CP

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

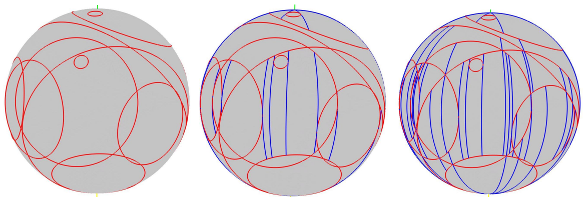
given a desired bound δ one can determine the necessary fp precision p

Molecular surfaces



sparse arrangement of spheres

Decomposition related degeneracies



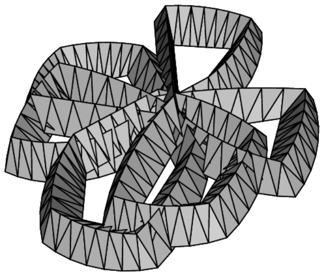
partial decomposition

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

(issues in dynamization, next week)

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

cons in red

- 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 on controlled perturbation

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

References

the material presented in class is mainly based on the following paper:

Halperin and Leiserowitz '04, Controlled perturbation of arrangements of circles, *International J. Computational Geometry and Application*, vol. 14, pp. 277–310

more details can be found at Leiserowitz's thesis:

www.cs.tau.ac.il/CGAL/Theses/leiserow/Thesis.ps.gz

Additional references

- Halperin and Shelton '98, A perturbation scheme for spherical arrangements with application to molecular modeling, *Computational Geometry: Theory & Applications*, Vol. 10, pp. 273–287
- Raab '00, Controlled perturbation for arrangements of polyhedral surfaces with application to swept volumes, *Proc. 15th ACM Symposium on Computational Geometry (SOCG)*, pp. 163–172
- Funke-Klein-Mehlhorn-Schmitt '05, Controlled Perturbation for Delaunay Triangulations, *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1047–1056.

THE END