CGAL

Computational Geometry Algorithms Library

Olivier Devillers
François Rebufat
Monique Teillaud

INRIA
SOPHIA ANTIPOLIS

Computational Geometry from the 70's

Lot of papers

Several implementations

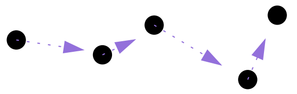
Very few distributed implementations

Libraries ?

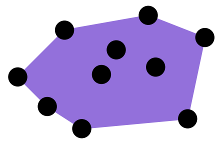
Computational Geometry from the 70's

What is a computational geometry problem ?

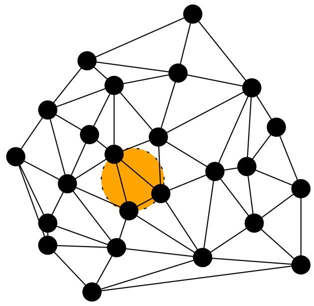
Combinatorial nature



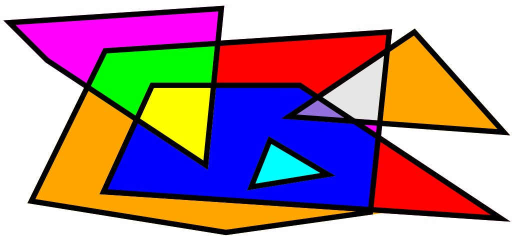
x-Sorting



Convex hull



Delaunay



Arrangement

Lower envelope

Computational Geometry from the 70's

Theoretical computational geometry

Design Algorithms

Worst case complexity

Asymptotic complexity

$O(n \log n)$

$\exists A \exists N \forall n \geq N f(n) \leq An$

$f(n)$

number of basic operations

A **?** N **?** basic operation **?**

Computational Geometry from the 70's

basic operation **?**

Real RAM model

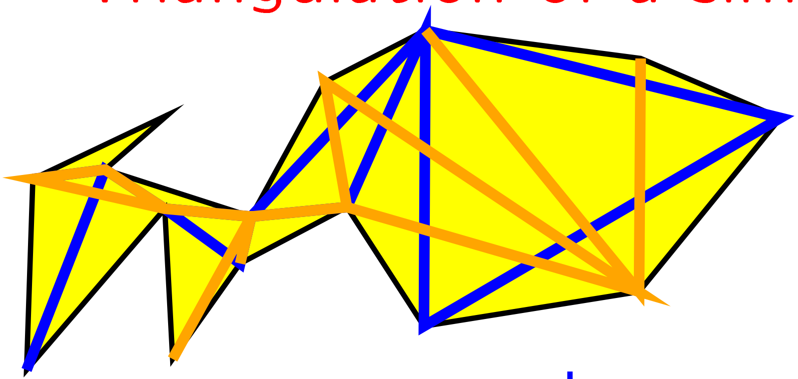
Basic arithmetic operations on real numbers cost 1

Challenge :

Search optimal algorithm for a given problem

Computational Geometry from the 70's

Example:
Triangulation of a simple polygon



$O(n \log n)$

$O(n \log \log n)$

$O(n \log^\star n)$

$O(n)$

Increasing complication

Computational Geometry from the 70's

Implementing is difficult

Optimal algorithms
are very complicated
→ code is hard to write

Particular cases are not
considered in papers
→ code is hard to write
code crashes

Floating point arithmetic
is a bad approximation of reals
→ code crashes

Computational Geometry from the 70's

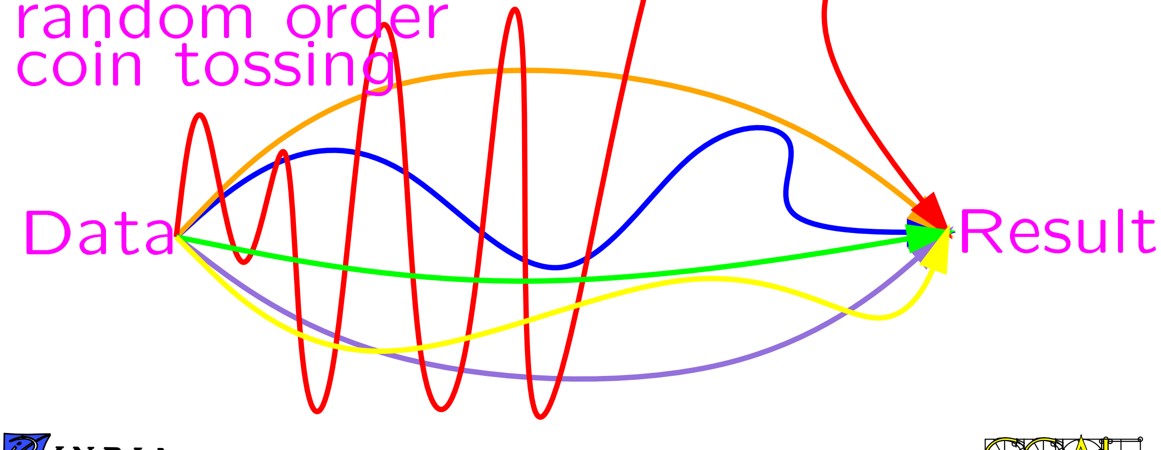
Simpler algorithms

Reasonable to implement

~~Worst case complexity~~

Randomized algorithms

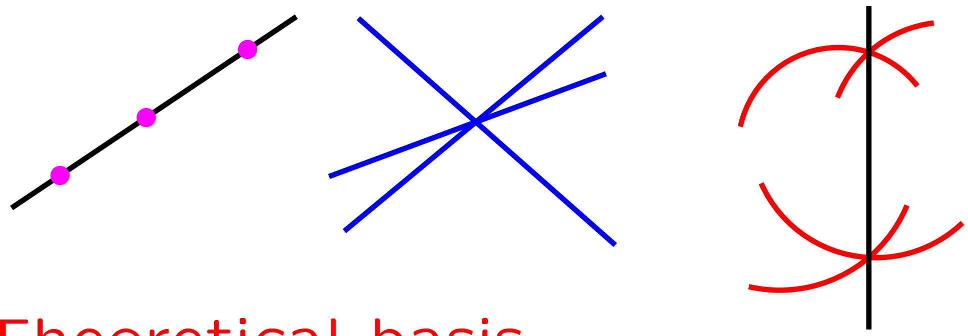
random order
coin tossing



Computational Geometry from the 70's

Particular cases

General position hypothesis



Theoretical basis
Simulation of Simplicity

Algorithm for non degenerate input

→ Algorithm for all input

Computational Geometry from the 70's

Floating point arithmetic

Exact arithmetic

Expensive

Analysis no longer valid

Supported operations ?

Computational Geometry from the 70's

Nothing easily available

for people wanting

just to use

computational geometry

Computational Geometry from the 70's

Aims

to have simple software solutions

Make solutions from
computational geometry
available to users

New domain

Geometric Computing

Computational Geometry from the 70's

Already existing software ?

Workbench for CG	Carleton
XYZ Geobench	Utrecht
Plageo	Zürich
C++GAL	INRIA
CEDA	Sarrebrücken
Gems	Minneapolis

1995 Idea

end 1996 European funding

Berlin	1989-95 Alcom AlcomII
INRIA	C++GAL
MPI, Saarbrücken	CEDA
Utrecht	Plageo

Linz

Tel Aviv

Zürich XYZ Geobench

Aims

Correctness

Robustness

Easy to use

Modularity

Flexibility (easy to reuse)

Efficiency

Advanced features of C++

Architecture

Basic Library

- polygon
- polyhedron
- convex hull
- planar map
- triangulation
- optimisation
- KD trees

Kernel

- 2D
- 3D
- dD

Core Library

- STL-like
- circulators
- assertions

Support Library

Visu Number types Random

CGAL Status

Current release February 1999

$\boxed{\text{CGAL 1.2}}$

450 `.h` files

450.000 lines

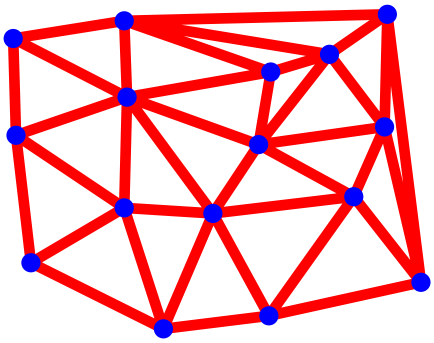
40 Mo

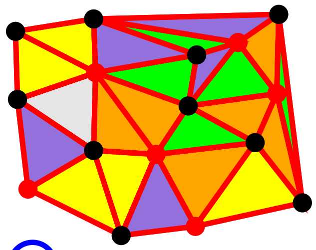
http://www.cs.uu.nl/CGAL/

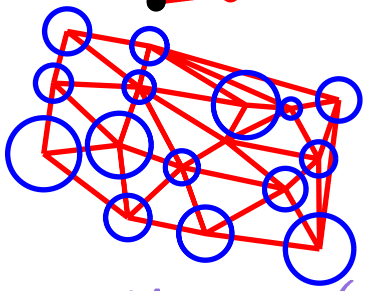
Flexibility (easy to reuse)

Triangulation algorithm

Set of points ⟶ Topology







User wants to

attach informations (colors)

not used standard points

Flexibility (easy to reuse)

CGAL approach

Traits class

$\boxed{\text{C++ Template}}$

User provides

its own classes

to CGAL algorithms

colored points
circles

CGAL triangulation $< Traits >$

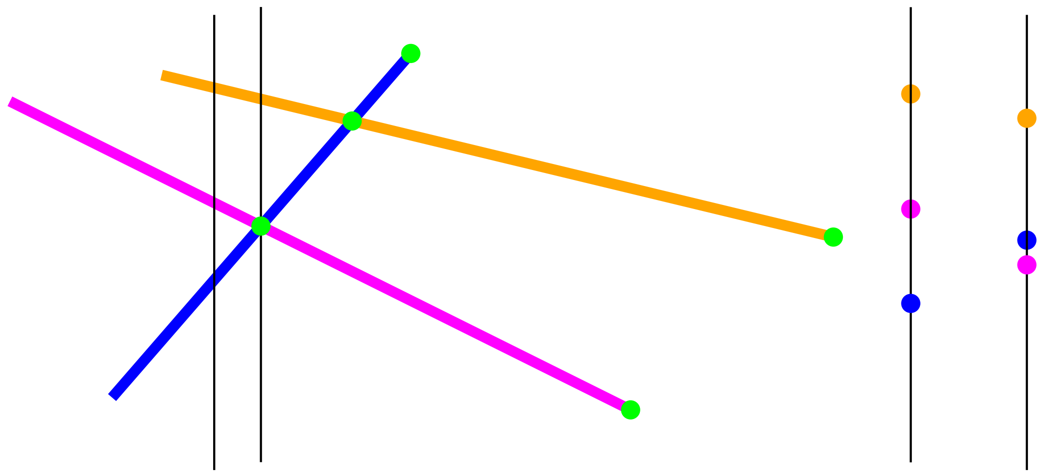
Robustness

Floating point arithmetic

is a bad approximation of reals

→ code crashes

Example: sweep line



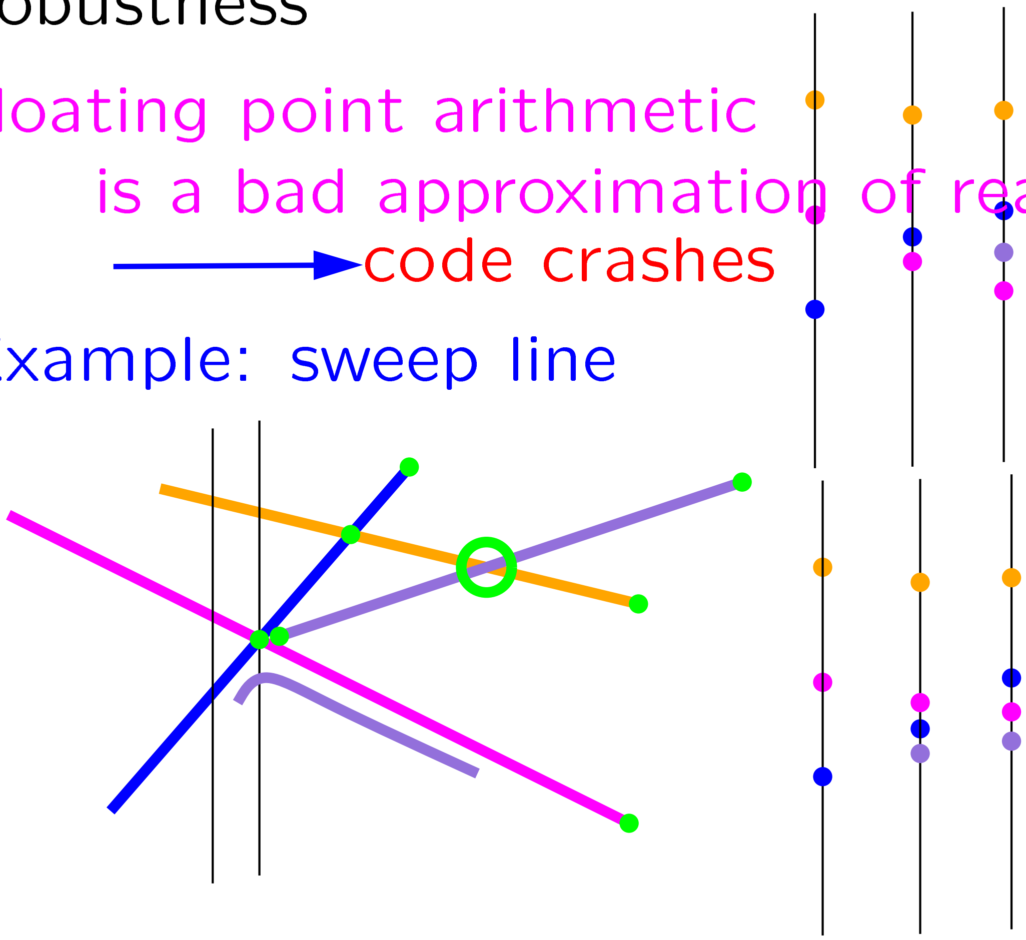
Robustness

Floating point arithmetic

is a bad approximation of reals

→ code crashes

Example: sweep line



Robustness

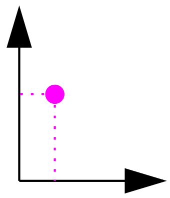
approach

CGAL Algorithms

Assume real RAM

CGAL Kernel

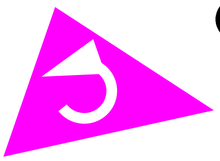
Use exact number types



Point<NT>

C++ Template

Use exact predicates



Orientation(p1,p2,p3);

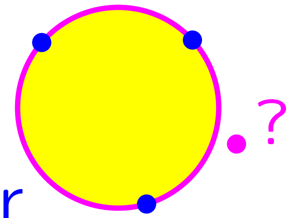
C++ Specialization

Correctness

CGAL approach

pre-conditions

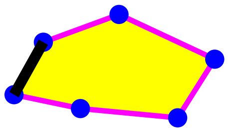
side of circle?



Outside

not collinear

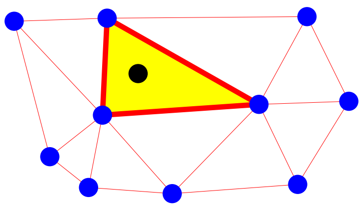
an edge ?



nb vertices > 1

post-conditions

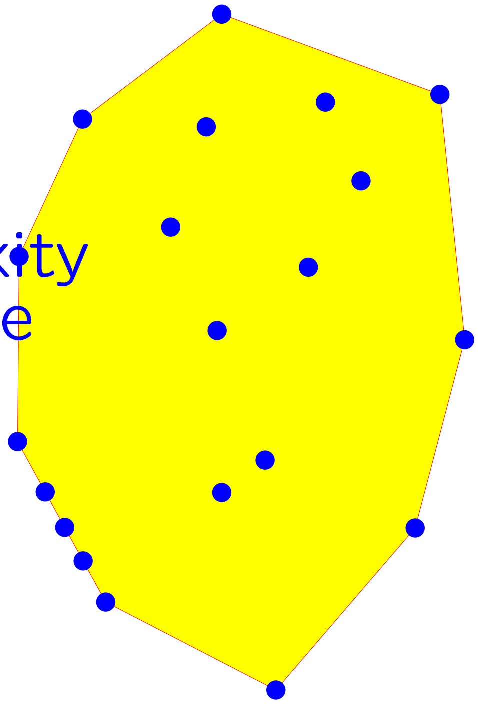
location query ?



contains query

Correctness

CGAL approach



checkers

Convex hull

Verify convexity
of an instance

test suites

Convex hull

Code testing
many cases
(all previous bugs)

Modularity

approach

Independant packages

Links through template parameters

Basic Library	Kernel	Core Library
polygon polyhedron convex hull planar map triangulation optimisation KD trees	2D 3D dD	STL-like circulators assertions

Support Library	Visu	Number types	Random

Easy to use

CGAL approach

the most difficult !

Unique look

STL like

Documentation

Efficiency

CGAL approach

compile time

C++ Template

C++ Specialization

distance(Point,Line)

no virtual functions

distance(Object,Object)

Examples

Future status

CGAL 1.2

Current release
February 1999

- C++ Template
- C++ Specialization

SunPro CC4.2
SGI CC7.2
g++ 2.8
egcs

CGAL 2.0

Summer 1999

- C++ Namespace
- Template members

~~SunPro CC~~
Visual C++6.0

New kernel design

CGAL 2.1

End 1999

- Template argument

New functionalities
3D triangulations
New triangulation design

Curso de CGAL

Programa

Jueves

Presentation
Olivier Devillers

C++, STL
François Rebufat

The library
Kernel
Basic library
François Rebufat

Viernes

Triangulations
Representation
Software design
Algorithms
Monique Teillaud

Robustness
Precision problems
Degeneracies
Predicates
Olivier Devillers

Presentation

Olivier Devillers

C++, STL

François Rebufat

Object oriented

Inheritance

Generic programming

Template

Traits class

Standard Template Library

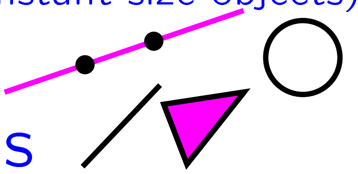
Containers

Iterators

Kernel

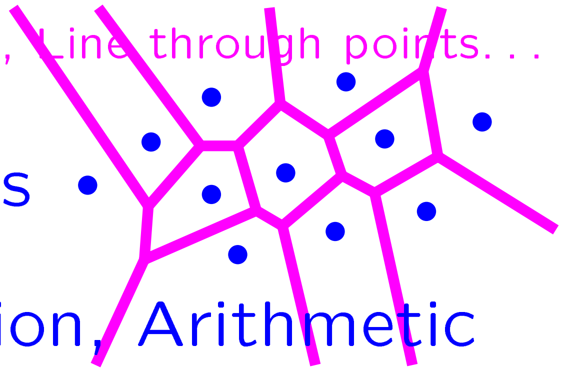
Basic objects (constant size objects)

Points, Lines, . . .



Basic operations

Orientation, Line through points. . .

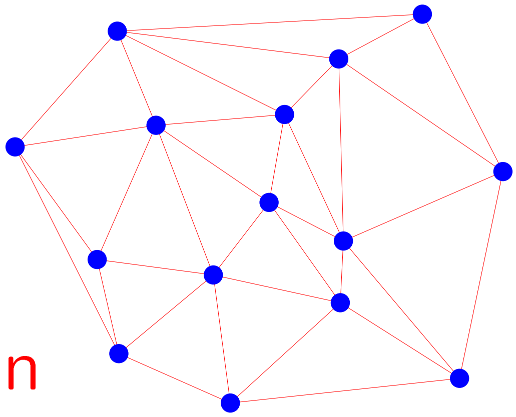


Basic Library

Algorithms

Support Library

Visualization, Arithmetic



Representation

Design

Traits class

geometry

Triangulation data structure

topology

Functionalities

Future design

Olivier Devillers

Theoretical algorithms

rely on

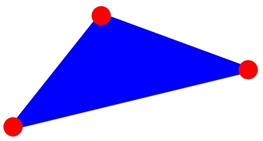
geometric theorems

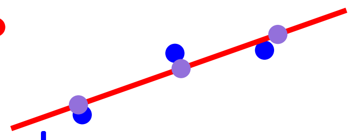
Arithmetic

- Floating point
- Exact
- Interval
- Fixed precision

Predicates

- Exact
- Filters





Degeneracies

- Perturbation scheme