Robustness of Geometric Computation

Claudio Mirolo

Dip. di Scienze Matematiche, Informatiche e Fisiche
Università di Udine, via delle Scienze 206 – Udine

claudio.mirolo@uniud.it

Computational Geometry

www.dimi.uniud.it/claudio
Outline

1. Geometric computation
   - issues
   - numeric vs. symbolic
   - geometric algorithms

2. An experiment
   - constructions
   - approaches
   - in summary

3. References
Outline

1 Geometric computation
   - issues
   - numeric vs. symbolic
   - geometric algorithms

2 An experiment
   - constructions
   - approaches
   - in summary

3 References
Processing of...

- Numerical information (e.g. coordinates, equations...)
- Symbolic information (e.g. incidence, adjacency...)

Geometric Computing = Numerical + Combinatorial Computing
Geometric computation

Processing of...

- Numerical information (e.g. coordinates, equations...)
- Symbolic information (e.g. incidence, adjacency...)

Geometric Computing = Numerical + Combinatorial Computing
Issues

- Numerical precision/accuracy
- Interaction between numeric and symbolic data
- Degenerate configurations
Issues

- Numerical precision/accuracy
- Interaction between numeric and symbolic data
- Degenerate configurations
Issues

- Numerical precision/accuracy
- Interaction between numeric and symbolic data
- Degenerate configurations
Interaction between numeric and symbolic data

Example:

- $P$ is a simple polygon, represented by its counterclockwise sequence of vertices $p_1, p_2, p_3, \ldots, p_{n-1}, p_n$

- $Q$ is a simple polygon, with the same vertices as $P$ + one more vertex $q$  $p_1, p_2, p_3, \ldots, p_{n-1}, p_n, q$

- $p_n, q, p_1$ collinear, with $q$ strictly in between $p_n$ and $p_1$

- Are $P$ and $Q$ the same polygon? …
Interaction between numeric and symbolic data

Example:

- $P$ is a simple polygon, represented by its counterclockwise sequence of vertices $p_1, p_2, p_3, \ldots, p_{n-1}, p_n$

- $Q$ is a simple polygon, with the same vertices as $P$ + one more vertex $q$ \[ p_1, p_2, p_3, \ldots, p_{n-1}, p_n, q \]

- $p_n, q, p_1$ collinear, with $q$ strictly in between $p_n$ and $p_1$

- Are $P$ and $Q$ the same polygon? …
Interaction between numeric and symbolic data

Example:

- $P$ is a simple polygon, represented by its counterclockwise sequence of vertices $p_1, p_2, p_3, \ldots, p_{n-1}, p_n$

- $Q$ is a simple polygon, with the same vertices as $P$ + one more vertex $q$ $p_1, p_2, p_3, \ldots, p_{n-1}, p_n, q$

- $p_n, q, p_1$ collinear, with $q$ strictly in between $p_n$ and $p_1$

- Are $P$ and $Q$ the same polygon? …
Interaction between numeric and symbolic data

Example:

- $P$ is a simple polygon, represented by its counterclockwise sequence of vertices $p_1, p_2, p_3, \ldots, p_{n-1}, p_n$

- $Q$ is a simple polygon, with the same vertices as $P$ + one more vertex $q$ $p_1, p_2, p_3, \ldots, p_{n-1}, p_n, q$

- $p_n, q, p_1$ *collinear*, with $q$ strictly in between $p_n$ and $p_1$

- Are $P$ and $Q$ the same polygon? …
Do $P$ and $Q$ describe the same set of points in the plane?

Which *operational* way to answer this question?

Possible answer:
No, since $P$ and $Q$ have a different number of sides.

Cheap to achieve,
but assumes an “economical” polygon representation!
Interaction between numeric and symbolic data

- Do \( P \) and \( Q \) describe the same set of points in the plane?

- Which *operational* way to answer this question?

  Possible answer:
  No, since \( P \) and \( Q \) have a different number of sides

  Cheap to achieve,
  but assumes an “economical” polygon representation!
Interaction between numeric and symbolic data

- Do \( P \) and \( Q \) describe the same set of points in the plane?
- Which \textit{operational} way to answer this question?
- Possible answer: No, since \( P \) and \( Q \) have a different number of sides
- Cheap to achieve, but assumes an “economical” polygon representation!
Interaction between numeric and symbolic data

- Do $P$ and $Q$ describe the same set of points in the plane?
- Which *operational* way to answer this question?
- Possible answer:
  No, since $P$ and $Q$ have a different number of sides
- Cheap to achieve,
  but assumes an “economical” polygon representation!
Interaction between numeric and symbolic data

- Does $Q$ represent a convex polygon?

- Which *operational* way to answer this question?

- Possible answer:
  No, since moving to the next side of $Q$ does not always imply “turning left”

- Again cheap to achieve, but... Operationally, at least, it can make a difference!
Does $Q$ represent a convex polygon?

Which *operational* way to answer this question?

Possible answer:
No, since moving to the next side of $Q$ does not always imply “turning left”

Again cheap to achieve, but... Operationally, at least, it can make a difference!
Does $Q$ represent a convex polygon?

Which operational way to answer this question?

Possible answer:
No, since moving to the next side of $Q$ does not always imply “turning left”

Again cheap to achieve, but... Operationally, at least, it can make a difference!
Does $Q$ represent a convex polygon?

Which *operational* way to answer this question?

Possible answer:
No, since moving to the next side of $Q$ does not always imply “turning left”

Again cheap to achieve, but . . .
Operationally, at least, it can make a difference!
The *symbolic* outcome of a computation may crucially depend on the specific sequence of steps.

Different sequences of steps may result in contradictory answers/decisions.

Ill-conditioned problems are highly sensitive to minor perturbations.

Is the outcome of a geometric algorithm acceptable for all legitimate input data?
Numeric vs. symbolic data

- The *symbolic* outcome of a computation may crucially depend on the specific sequence of steps.
- Different sequences of steps may result in contradictory answers/decisions.
- Ill-conditioned problems are highly sensitive to minor perturbations.
- Is the outcome of a geometric algorithm acceptable for all legitimate input data?
Numeric vs. symbolic data

- The *symbolic* outcome of a computation may crucially depend on the specific sequence of steps.

- Different sequences of steps may result in contradictory answers/decisions.

- Ill-conditioned problems are highly sensitive to minor perturbations.

- Is the outcome of a geometric algorithm acceptable for *all* legitimate input data?
The *symbolic* outcome of a computation may crucially depend on the specific sequence of steps.

Different sequences of steps may result in contradictory answers/decisions.

Ill-conditioned problems are highly sensitive to minor perturbations.

Is the outcome of a geometric algorithm acceptable for *all* legitimate input data?
... Polygon representation example:

- And if the representation of vertex $q$ is inaccurate? (e.g. because it was computed with limited precision)

- How reliable is to test the convexity of $Q$?

- What about subsequent decisions based on the result of this test?
Is the output acceptable?

... Polygon representation example:

- And if the representation of vertex \( q \) is inaccurate? (e.g. because it was computed with limited precision)

- How reliable is to test the convexity of \( Q \)?

- What about subsequent decisions based on the result of this test?
Is the output acceptable?

... Polygon representation example:

- And if the representation of vertex $q$ is inaccurate? (e.g. because it was computed with limited precision)
- How reliable is to test the convexity of $Q$?
- What about subsequent decisions based on the result of this test?
Geometric algorithms

- *Theory*: Standard proofs of correctness assume exact computation with real numbers

- *Practice*: Inexact floating-point arithmetic in the implementation
Geometric algorithms

- **Theory**: Standard proofs of correctness assume exact computation with real numbers

- **Practice**: Inexact floating-point arithmetic in the implementation
Outline

1. Geometric computation
   - issues
   - numeric vs. symbolic
   - geometric algorithms

2. An experiment
   - constructions
   - approaches
   - in summary

3. References
Hoffmann (1989): Iterating inner/outer constructions
Hoffmann (1989): How accurate is the intersection?
Approaches to geometric computing

- **Fixed precision (usually floating point) computation:**
  - `float`
  - `double`
  - “heuristic $\varepsilon$”

- **Exact computation:**
  - algebraic numbers
  - exact integer & rational arithmetic (e.g. via `multiple precision integers $\rightarrow$ GMP`)
  - adaptive evaluation
  - …
Approaches to geometric computing

- **Fixed precision (usually floating point) computation:**
  - float
  - double
  - “heuristic $\varepsilon$”

- **Exact computation:**
  - algebraic numbers
  - exact integer & rational arithmetic
    - (e.g. via *multiple precision integers* $\rightarrow$ GMP)
  - adaptive evaluation
    - ...
    - ...
Approaches to geometric computing

- **Fixed precision (usually floating point) computation:**
  - float
  - double
  - “heuristic $\varepsilon$”

- **Exact computation:**
  - algebraic numbers
  - exact integer & rational arithmetic
    (e.g. via *multiple precision integers* $\rightarrow$ GMP)
  - adaptive evaluation
  - ...
Approaches to geometric computing

- **Fixed precision (usually floating point) computation:**
  - float
  - double
  - “heuristic $\varepsilon$”

- **Exact computation:**
  - algebraic numbers
  - exact integer & rational arithmetic
    (e.g. via *multiple precision integers* $\rightarrow$ GMP)
  - adaptive evaluation
  - … …
Tradeoffs!

- Computational costs (time, space) ... and tractability

- “Combinatorial” soundness and constructions accuracy
Tradeoffs!

- Computational costs (time, space) . . . and tractability

  “Combinatorial” soundness and constructions accuracy
Tradeoffs!

- Computational costs (time, space) . . . and tractability
- “Combinatorial” soundness and constructions accuracy
Dealing with geometric structures

- Spatial positions, orientations, ...: Numerical data (measures)
- Relationships between items/components, e.g. topology, ordering, ...
- Algorithmic logic: decisions based on relationships (combinatorial)
- Construction of new geometric objects — including new numerical data
Dealing with geometric structures

- Spatial positions, orientations, ...: Numerical data (measures)
- Relationships between items/components, e.g. topology, ordering, ...
- Algorithmic logic: decisions based on relationships (combinatorial)
- Construction of new geometric objects — including new numerical data
Dealing with geometric structures

- Spatial positions, orientations, ...:
  Numerical data (measures)

- Relationships between items/components,
  e.g. topology, ordering, ...

- Algorithmic logic: decisions based on relationships
  (combinatorial)

- Construction of new geometric objects
  — including *new* numerical data
Dealing with geometric structures

- Spatial positions, orientations, ...: Numerical data (measures)

- Relationships between items/components, e.g. topology, ordering, ...

- Algorithmic logic: decisions based on relationships (combinatorial)

- Construction of new geometric objects — including *new* numerical data
General observations

- It is often possible to evaluate correctly relationships between objects represented accurately.

- It is not always possible/feasible to construct new sufficiently accurate objects.

- Even though the input data are assumed to be correct (or accurate enough), crucial problems may arise if algorithmic decisions depend on relations that involve new geometric objects.
General observations

- It is often possible to evaluate \textit{correctly} relationships between objects represented \textit{accurately}.

- It is not always possible/feasible to construct new \textit{sufficiently accurate} objects.

- Even though the input data are assumed to be correct (or accurate enough), crucial problems may arise if algorithmic \textit{decisions} depend on relations that involve new geometric objects.
General observations

- It is often possible to evaluate *correctly* relationships between objects represented *accurately*

- It is not always possible/feasible to construct new *sufficiently accurate* objects

- Even though the input data are assumed to be correct (or accurate enough) crucial problems may arise if algorithmic *decisions* depend on relations that involve new geometric objects
Outline

1. Geometric computation
   - issues
   - numeric vs. symbolic
   - geometric algorithms

2. An experiment
   - constructions
   - approaches
   - in summary

3. References
References

- C.M. Hoffmann (1989)  
The Problems of accuracy and robustness in geometric computation  
  *Computer, 22*(3)

- C.-K. Yap (1997)  
  Towards exact geometric computation  
  *Computational Geometry, 7*(1)

- S. Schirra (2000)  
  Robustness and precision issues in geometric computation  
  *Handbook of Computational Geometry, Ch. 14*