Controlled Linear Perturbation

Elisha Sacks, Purdue University

joint work with

Victor Milenkovic, University of Miami

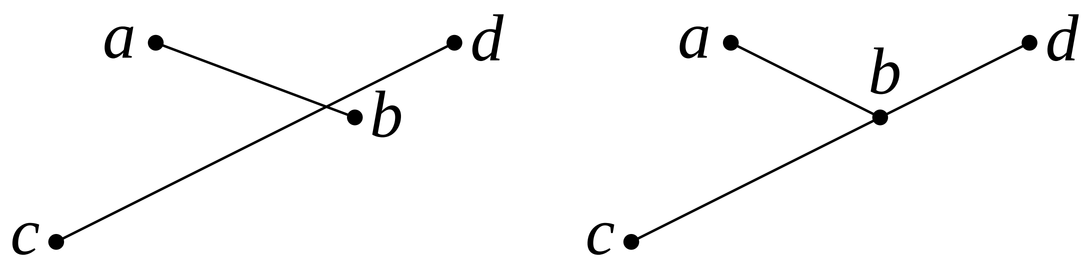
Min-Ho Kyung, University of Ajou

Research supported by a PLM grant and by NSF grants CCF-0904832 and CCF-0904707.

Robustness Problem

- Algorithms are expressed in real RAM model.
- Input is assumed in general position.
- Implementations must use computer arithmetic.
- Implementations must handle degenerate input.
- Implementations must be fast and accurate.

Geometric Predicates



unsafe degenerate

Main interface with real RAM model (also constructions).

- Predicate $P(x)$ is true when polynomial $f(x)$ is positive.
- Unsafe predicate: $|f(x)|$ near the rounding unit.
- Degenerate predicate: $f(x) = 0$.
- Example: b above cd if $(d - c) \times (b - c) > 0$;
  $f(b_x, b_y, c_x, c_y, d_x, d_y) = (d_x - c_x)(b_y - c_y) - (d_y - c_y)(b_x - c_x)$.

Exact Computational Geometry

Implement predicates exactly using algebra.

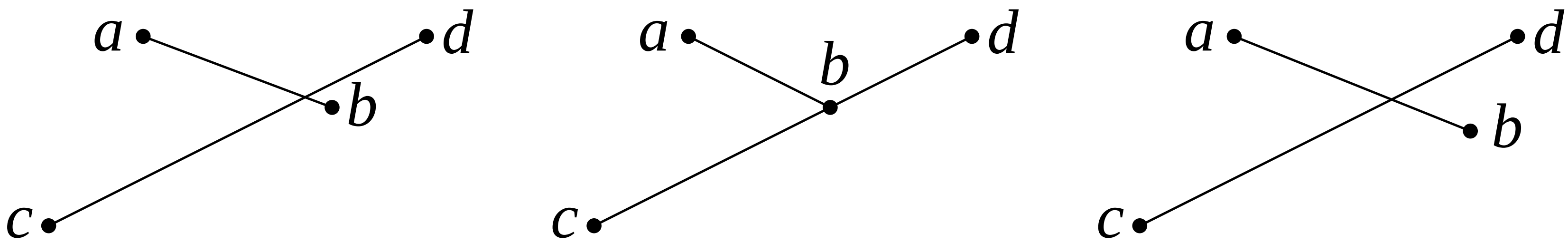
- Technical problems
  - Running time grows rapidly with algebraic degree.
  - Bit complexity grows rapidly.
  - Degeneracy requires separate, complex algorithm.
- Conceptual problem
  - Scientific computing is approximate because exact solutions are impractical and unnecessary.
  - Gold standard is fast algorithms with error bounds.
  - Why should computational geometry be exact?

Approximate Computational Geometry

Implement predicates approximately using floating point arithmetic and numerical solvers.

- Advantages:
  - Running time grows modestly with degree.
  - Constant bit complexity.
  - Small constant factors.
- Challenge: generate consistent output.

Consistency



unsafe i degenerate i consistent p

- An algorithm is consistent when for every input, i, there exists a perturbed input, p, such that the computed predicates are correct for p.
- The output error is the distance from i to p, $||p - i||$.
- Inconsistent algorithms can crash or output garbage.

Perturbation Strategy

1. Derive a safety threshold, e, such that $|f(x)| > e$ in floating point implies safety.
2. Every time a predicate polynomial is evaluated, check its threshold.
3. If the check fails, perturb the input.

CP versus CLP

- Controlled perturbation (CP)
  - Perturb randomly and restart.
  - Error exponential in polynomial degree.
  - No equality constraints or parameter definitions.
- Controlled linear perturbation (CLP)
  - Perturb carefully and continue.
  - No error on safe polynomials.
  - Error grows modestly with degree.
  - Equality constraints and parameter definitions.

CLP Strategy

Given: polynomial $f(x)$ with input $x = a$.

1. If $|f(a)| > e$, return its sign.
2. Compute p such that $|f(p)| > e$ and likewise for previous polynomials.
3. Return the sign of $f(p)$.

CLP Algorithm

- Write $p = a + \delta v$ with $\delta \geq 0$ the perturbation size and v the perturbation direction.
- Linearize $f(p) \approx f(a) + \delta \nabla f \cdot v$ with ∇f the gradient.
- Linearization error is negligible because δ is tiny.
- Initialize $\delta = 0, v = 0$ and update for each unsafe f.
- Best v for sign s is $s\nabla f$ ignoring prior unsafe f_i.
- Define u by subtracting ∇f_i from ∇f and unitizing.
- Update v to $v + su$; pick $s = \pm 1$ with smaller $\delta = (2se - f(a))/(s\nabla f \cdot v)$; update δ.
- Prior f_i are uneffected by change in v; become safer due to increase in δ.
- Verify signs of $f(p)$ for final p.

Sorting Example

- Sort $x = (0, 0, 0, 1)$ in increasing order.
- Predicate $x_i < x_j$ has polynomial $x_j - x_i$ with $e = 2\mu$.
- Degenerate for $i, j < 4$; safe otherwise.

1) Evaluate $x_2 - x_1$ with $v = (0, 0, 0, 0)$ and $orth = \emptyset$.

- $\nabla f = (-1, 1, 0, 0)$, so $u_1 = \sqrt{0.5}(-1, 1, 0, 0)$.
- For $s = 1$, $\delta = \frac{2se - f(a)}{s\nabla f \cdot v} = \frac{4\mu}{u_1 \cdot \nabla f}$.
- For $s = -1$, $\delta = \frac{-4\mu}{-u_1 \cdot \nabla f}$.
- CLP picks $s = 1$, $\delta \approx 2.8\mu$, and $x_1 < x_2$.

Sorting Example Continued

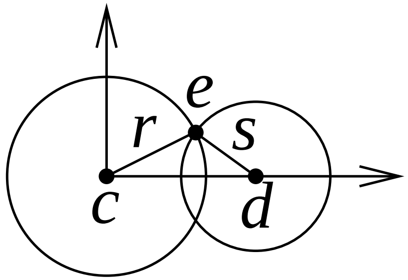
2) Evaluate $x_3 - x_1$ with $v = u_1$ and $orth = \{u_1\}$.

- $\nabla f = (-1, 0, 1, 0)$, so $u_2 = \sqrt{1/6}(-1, -1, 2, 0)$.
- For $s = 1$, $\delta < 2.8\mu$, so CLP picks it and $x_1 < x_3$.

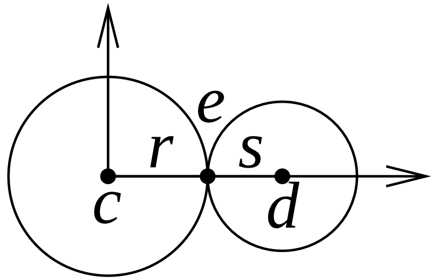
3) Evaluate $x_3 - x_2$ with $v = u_1 + u_2 \approx (-1.1, 0.3, 0.8, 0)$ and $orth = \{u_1, u_2\}$.

- $\nabla f = (0, -1, 1, 0)$, so $u_3 = (0, 0, 0, 0)$.
- Only choice is $s = 1$ with $\delta \approx \frac{4\mu}{-0.3+0.8} \approx 7.7\mu$ and $x_1 < x_2$.
- Final order, $x_1 < x_2 < x_3$, derives from v.

Parameter Definitions



full rank



rank deficient

Define new parameters, $y = b$, with equations $g(x, y) = 0$.

- Enforce linearized equations by adding to $orth$.
- Extend v to y by solving $g_x v + g_y w = 0$.
- Example: $e = (y_1, y_2)$, $b = (3.6, -1.8)$, equations $g(x, y)$
  $(y_1 - x_1)^2 + (y_2 - x_2)^2 - x_3^2 = 0$
  $(y_1 - x_4)^2 + (y_2 - x_5)^2 - x_6^2 = 0$.
- Rank deficient case requires special handling.

Singular Predicate Polynomials

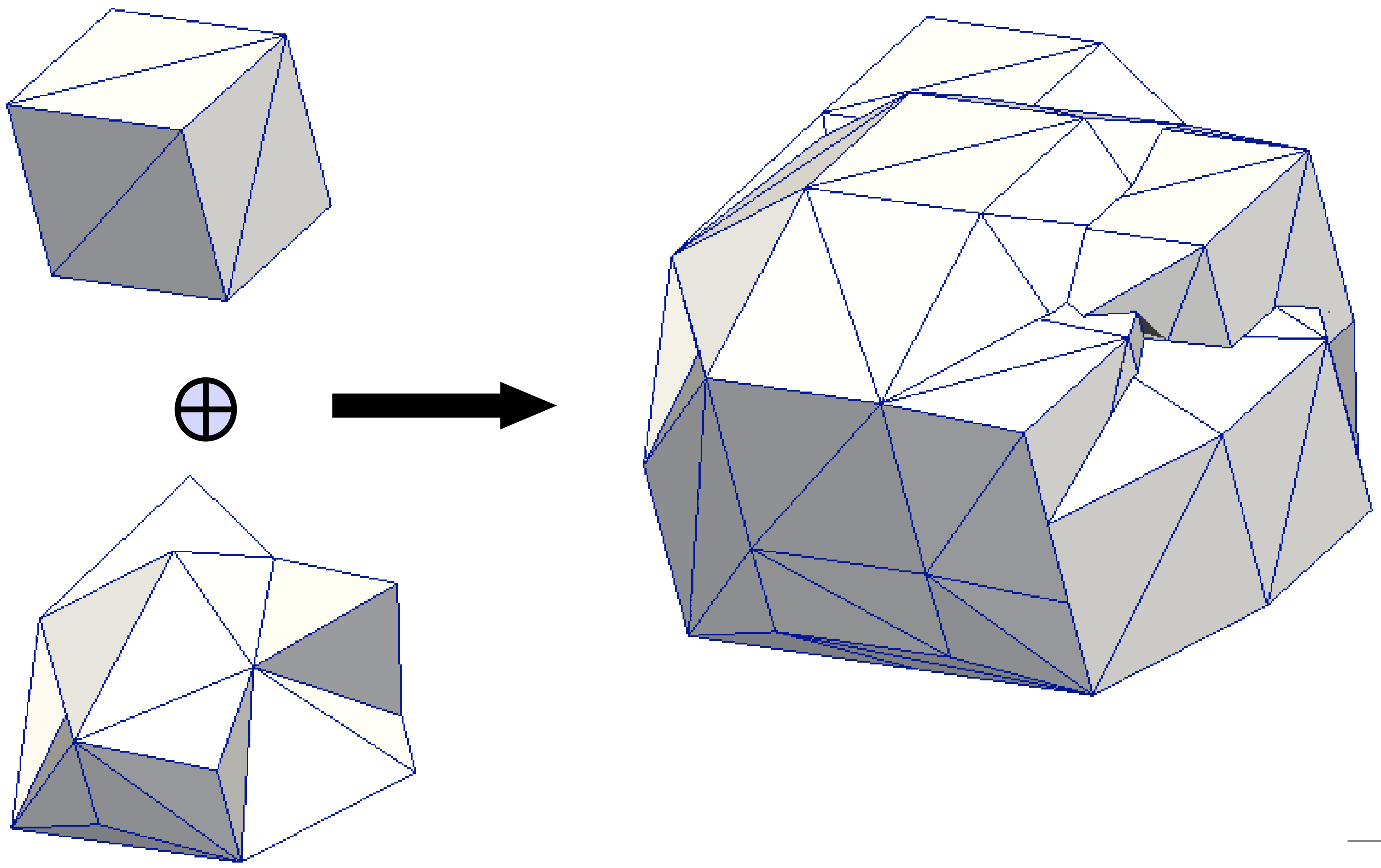
- Polynomial f is singular when $\nabla f = 0$.
- Core CLP fails on near singular unsafe polynomials.
- We avoid near singularity by predicate reformulation using judicious parameter definitions.
- Example: point b above line cd with $b = c = d$.
- Define unit vector, $u = (d - c)/||d - c||$, with equations $u \cdot u - 1 = 0$ and $u \times (d - c) = 0$.
- Reformulation: $u \times (b - c) = u_x(b_y - c_y) - u_y(b_x - c_x)$
- Gradient is large: $(-u_y, u_x, u_y, -u_x, b_y - c_y, c_x - b_x)$.

Minkowski Sums

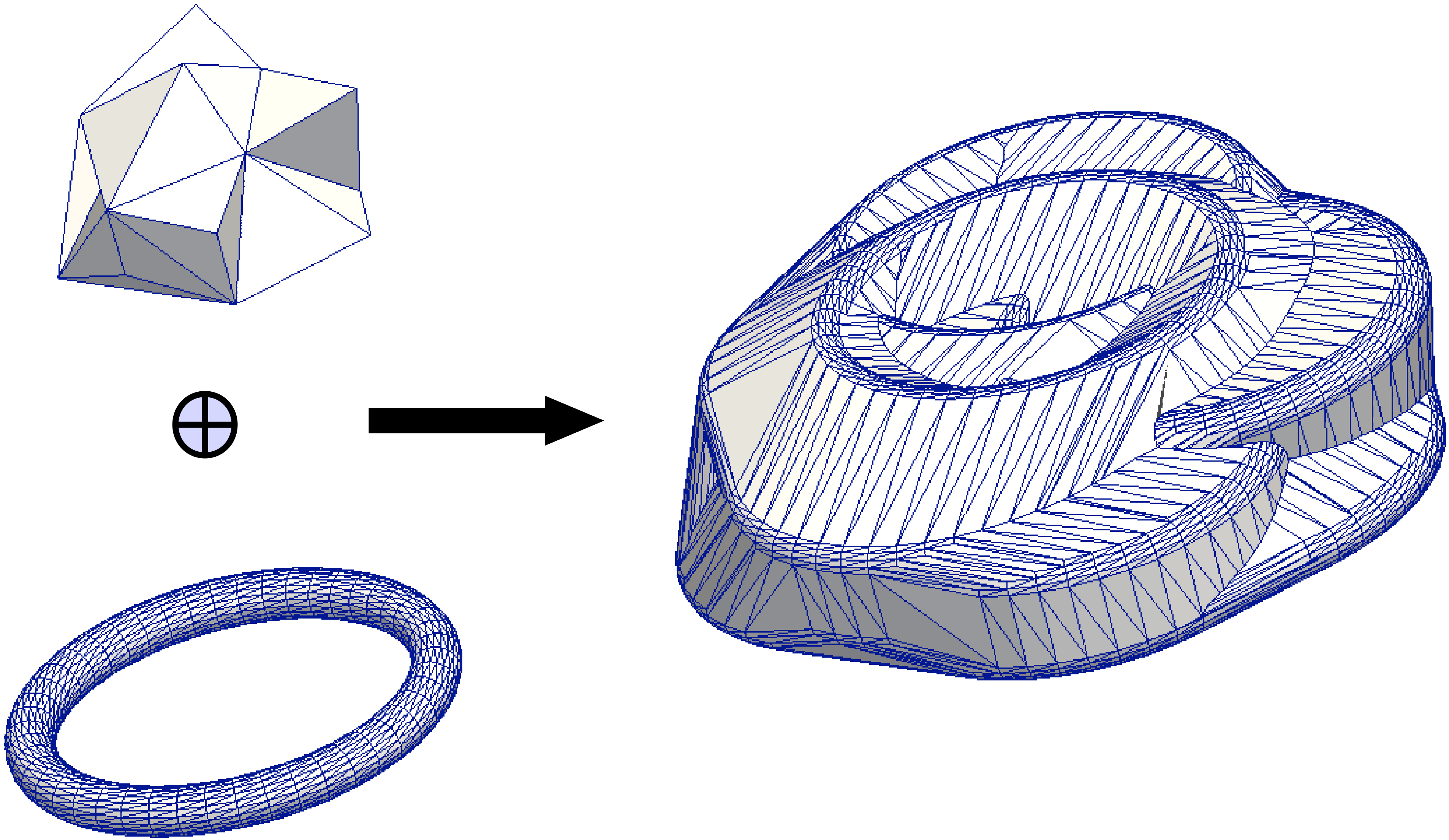
Minkowski sum of point sets A and B is the point set $A \oplus B = \{a + b \,|\, a \in A, b \in B\}$.

- Minkowski sums of polyhedra have many applications: packing, path planning, assembly, graphics, solid modeling, mechanics, simulation.
- Prior implementations decompose polyhedra into convex pieces.
- Complexity is $\Omega(n^4)$ for input size n; prohibitive.
- Approximate algorithms are tricky and slow.
- Efficient output sensitive algorithm known for 30 years.
- We use CLP to obtain the first robust implementation.

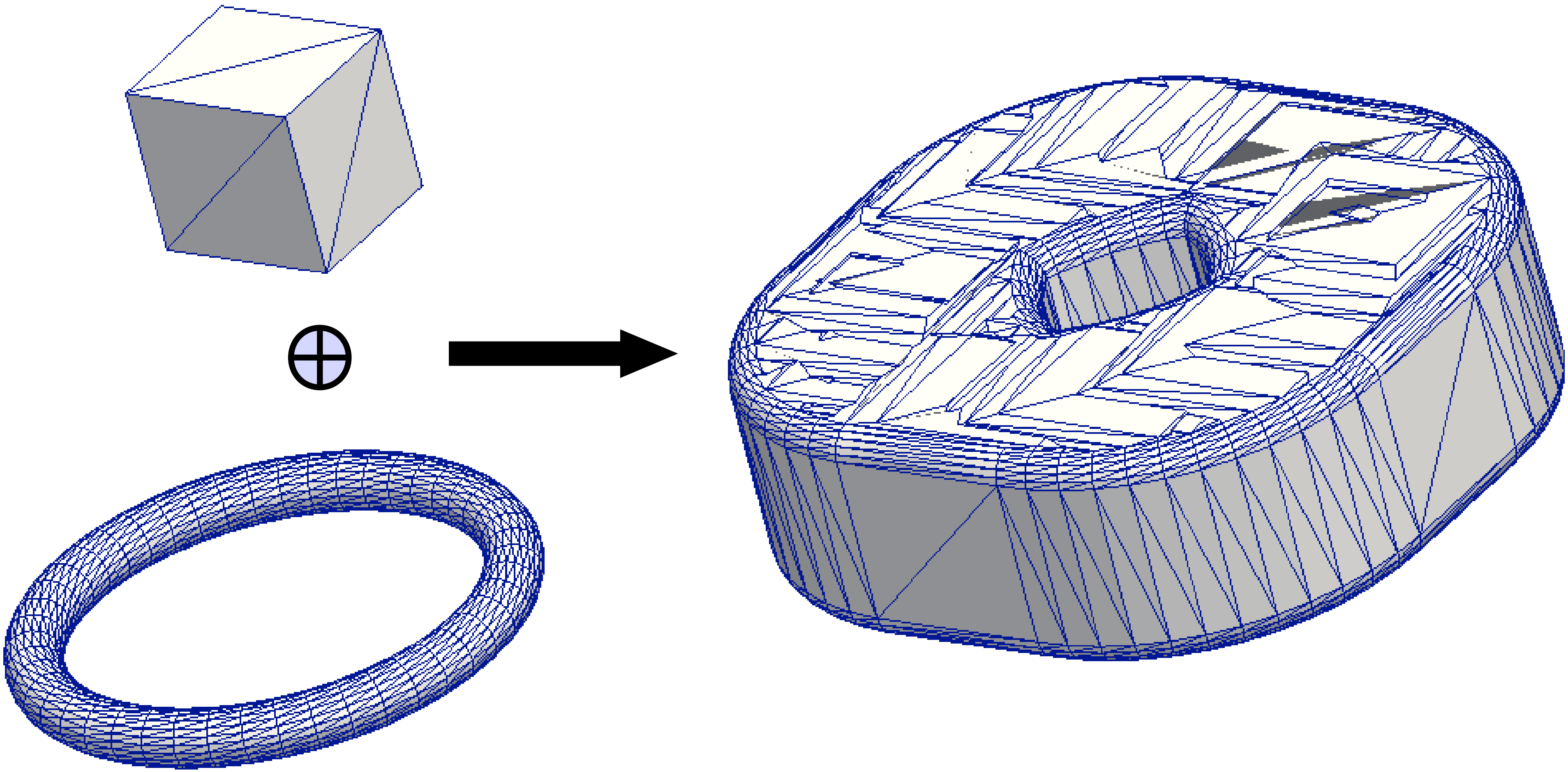
Cube + Polyhedron



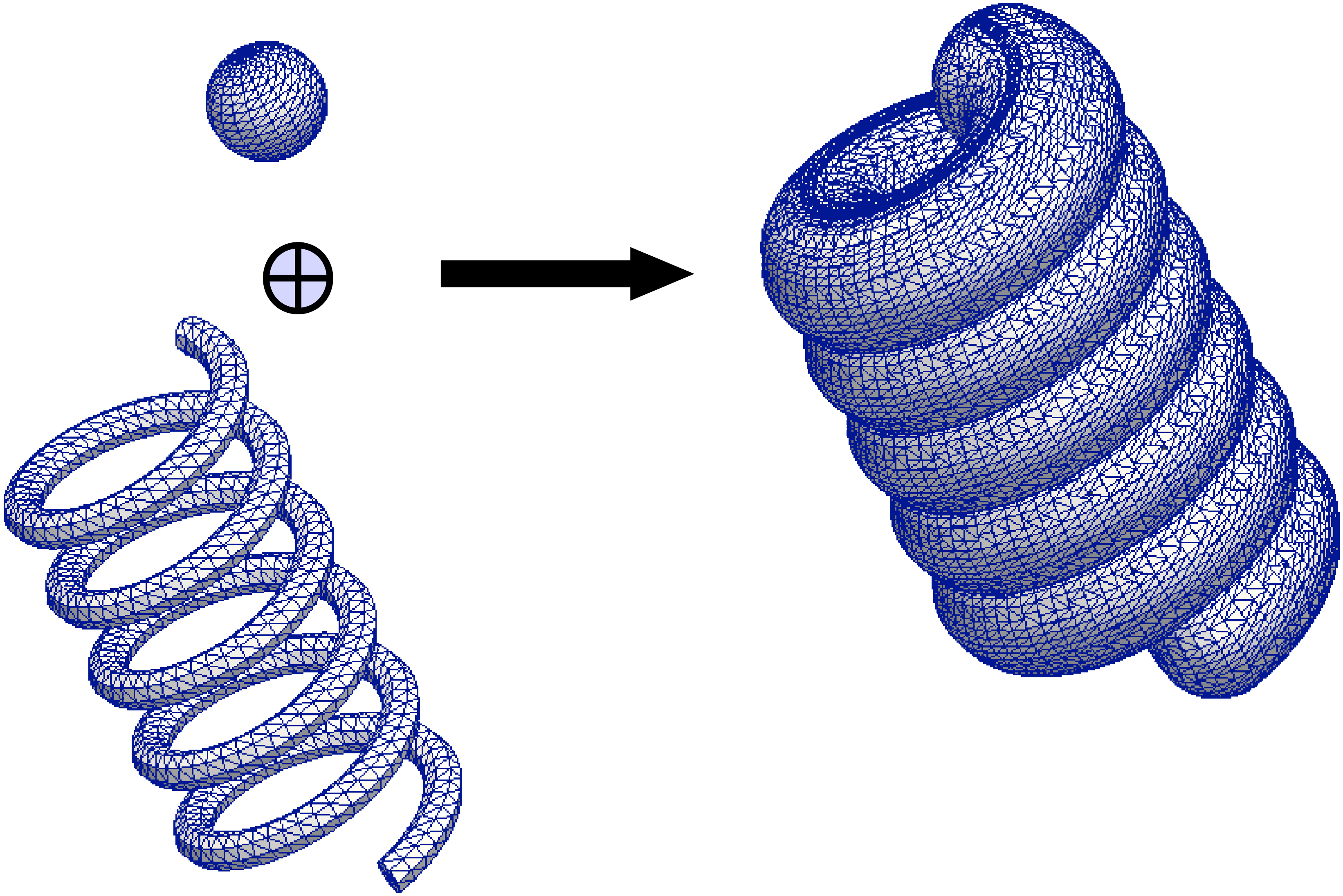
Torus + Polyhedron



Cube + Torus



Sphere + Helix



Results

A, B, and *conv* the number of triangles in part A, part B, and the convolution, *time* the running time in seconds, *safe* and *unsafe* the number of safe and unsafe predicate polynomials, and δ the final perturbation size.

	A	B	*conv*	*time*	*safe*	*unsafe*	δ
a	12	32	130	0.1	4e5	5,393	3e-12
b	32	2068	4336	5	2.2e6	13,145	8e-10
c	12	2068	2946	4.3	1.1e6	15,991	2e-10
d	760	4012	40,212	49	27e7	43,752	1e-10

Future Work

- Research
  - Automated handling of near singular polynomials.
  - Output simplification.
- Education
  - Computational geometry curriculum organized around robustness.
  - Computational geometry textbook organized around robustness.
- Applications
  - Patent.
  - Software library.
  - Robust applications software.