Building blocks towards modeling the physical world: analysis, geometry, computer arithmetics

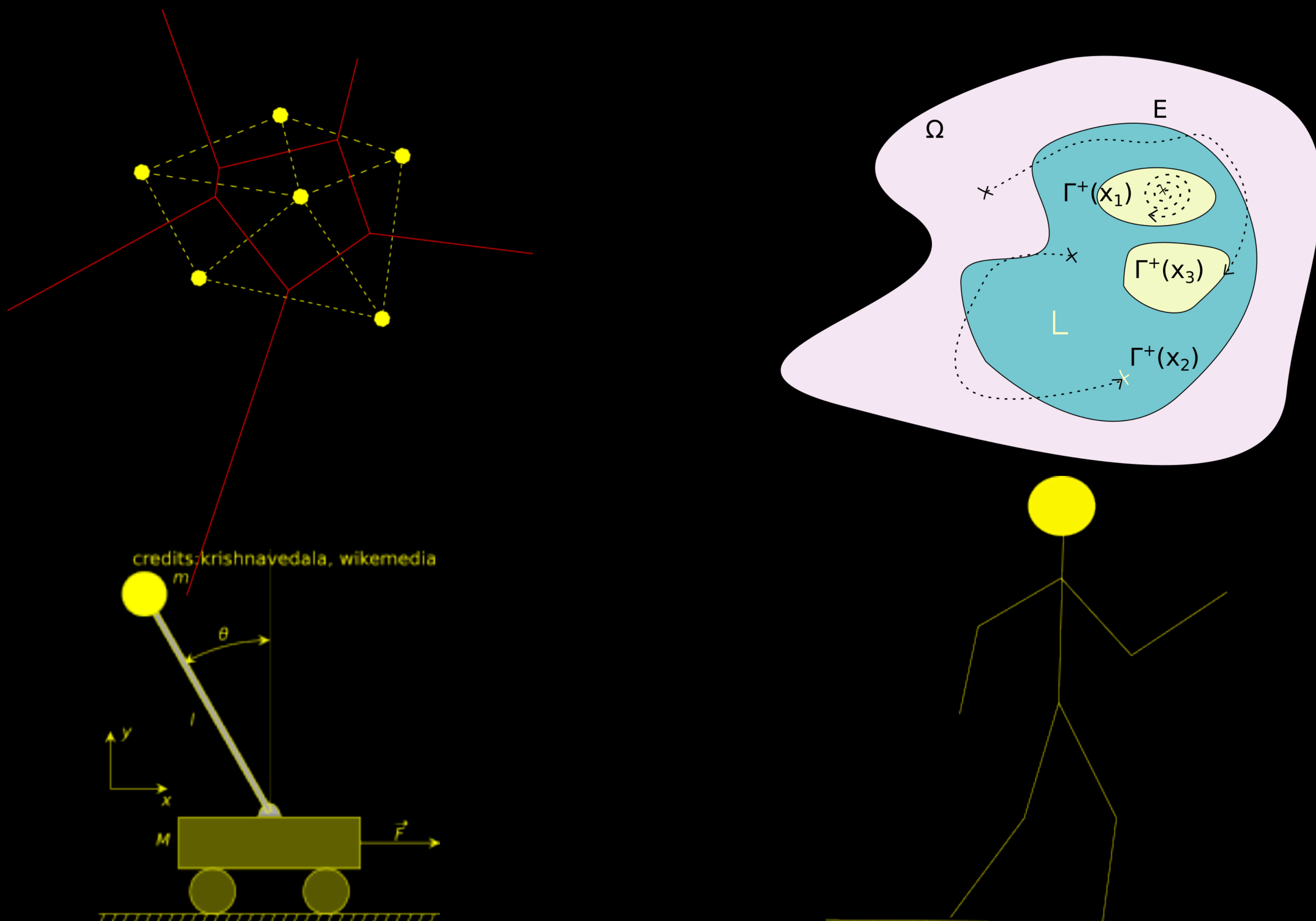
Yves Bertot
Université Côte d’Azur, Inria

Robot invasion

- Robots are progressing into our daily life
- How much of their design is relevant for formal proof
  - Time-to-market concerns
  - Public safety concerns
- Sobering note: no need for formal proofs

Sample design questions

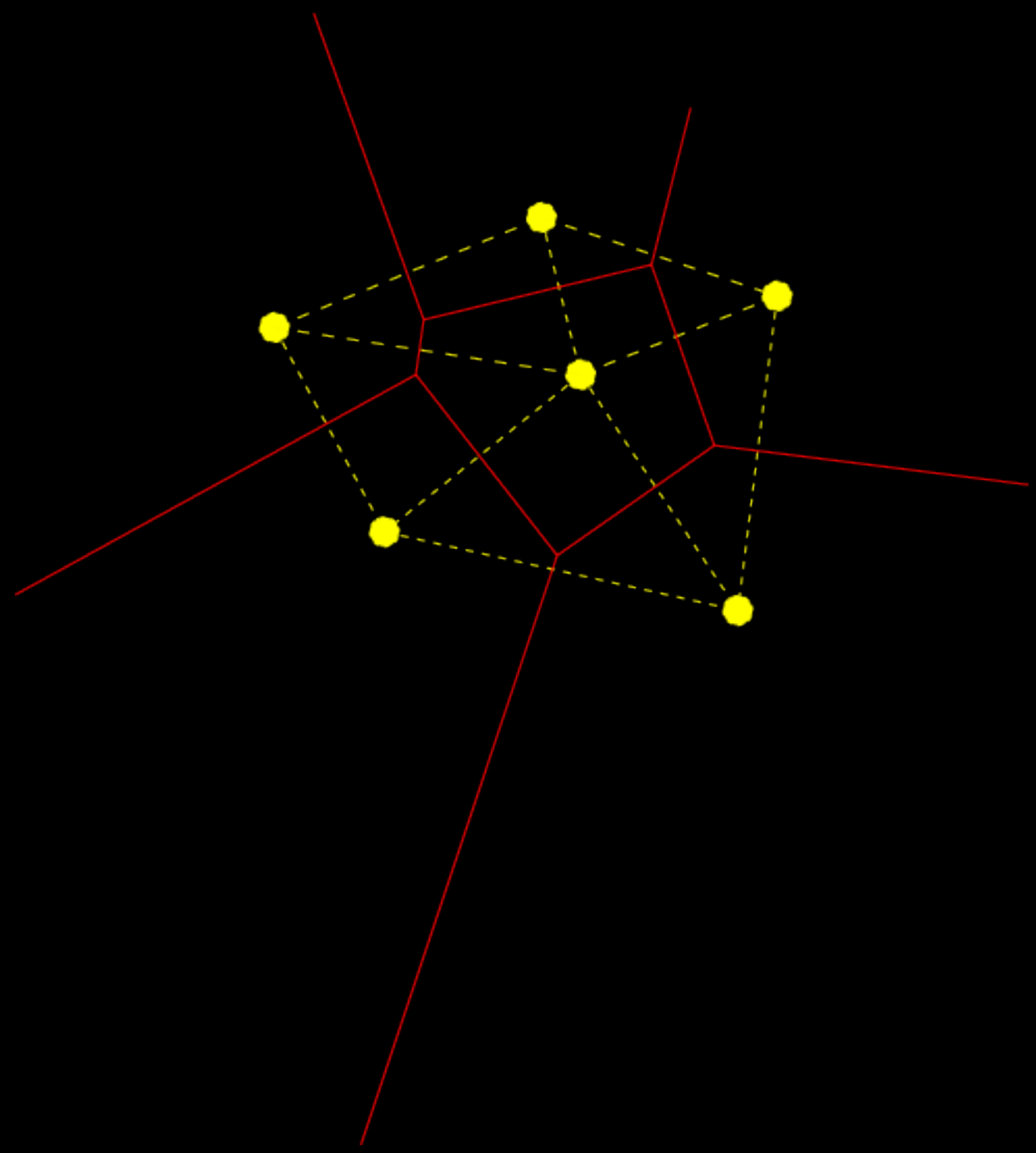
- Mechanical design: reachability, joint stress
- Control software: stability, robustness
- Sensing: visual, dead reckoning, SLAM
- Uncertainties: geometry, sensing, actuating
- Motion planning: collision avoidance
- Autonomous decision: interaction with humans


credits:krishnavedala, wikemedia
m
θ
l
y
x
M
F
Ω
E
Γ+(x1)
Γ+(x3)
Γ+(x2)

Chosen topics

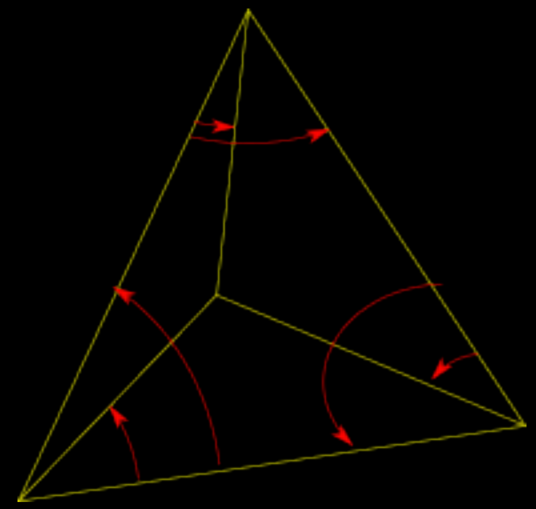
- Computational geometry: triangulations and Delaunay for motion planning
- Analysis and computer arithmetic

Delaunay triangulations



Triangles

- Basic concept : counterclockwise triplet
  - Comes with a few properties [Knuth92]
- Derived notion: inside triangle



Triangulation

- Input: a set of points *d*
- output: a set of triangles
  - Union of triangle vertices is *d*
  - No point in *d* is inside a triangle
  - Triangles and their edges cover the convex hull of *d*
  - other conditions

Operations on triangulations

- Add a point (inside a triangle, on an edge, outside the hull)
- Flip an edge
- Show that properties of triangulations are preserved
- Optimisations: find triangles, edges

Towards Delaunay

- Repeatedly flip irregular edges
- Prove termination [Dufourd&B10]
- Find efficient data structure, e.g. Hypermaps
- Higher dimensions
- Implementations as refinements of abstract presentation

Concrete CCW

- Positivity of a determinant
- Logical conditions : determinant properties (e.g. Cramer's law) in general positions
- Difficulty with degenerate positions [B&Pichardie01]

CCW and in-circle algebraically

$$\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \qquad \begin{vmatrix} 1 & x_1 & y_1 & x_1^2 + y_1^2 \\ 1 & x_2 & y_2 & x_2^2 + y_2^2 \\ 1 & x_3 & y_3 & x_3^2 + y_3^2 \\ 1 & x_4 & y_4 & x_4^2 + y_4^2 \end{vmatrix}$$

- Required properties: n linear alternating form, Laplace, Cramer, Jensen’s inequality
- Many properties in mathematical components

Combinatorics

- Take points in any numerical type
- Use only a finite subset of the points
- Make a triangle from 3 points: degenerate cases
- Plan to explore efficient data-structures

Analysis example

- Algorithm for computing many decimals of π
- Based on arithmetic geometric means

Arithmetic geometric means

$$a_{n+1} = \frac{a_n + b_n}{2} \qquad b_{n+1} = \sqrt{a_n b_n}$$

$$\text{often} \quad a_0 = 1 \qquad b_0 = x$$

$$y_n = \frac{a_n}{b_n} \qquad z_n = \frac{b'_n}{a'_n}$$

$$y_{n+1} = \frac{1 + y_n}{2\sqrt{y_n}} \qquad z_{n+1} = \frac{1 + y_n z_n}{(1 + z_n)\sqrt{y_n}}$$

Computing **π**

$$\text{for} \quad a_0 = 1 \qquad b0 = \frac{1}{\sqrt{2}}$$

$$\pi_n = (2+\sqrt{2}) \prod_{i=1}^{n} \frac{1+y_n}{1+z_n}$$

$$\pi'_n = \frac{4a_n^2}{1 - \sum_{i}^{n-1} 2^i (a_i - b_i)^2}$$

$$\pi_n - \pi = O\left(500^{-2^{n-1}}\right) \quad \pi'_n - \pi = \tilde{O}\left(500^{-2^{n-1}}\right)$$

Elliptic integrals

$$I(a,b) = \int_{-\infty}^{+\infty} \frac{\mathrm{d}t}{\sqrt{(a^2+t^2)(b^2+t^2)}}$$

$$I(a_n,b_n) = I(a_{n+1},b_{n+1}) = \frac{\pi}{M(a,b)}$$

$$I(1,x) = 4\int_0^{\sqrt{x}} \frac{\mathrm{d}t}{\sqrt{(1+t^2)(x^2+t^2)}}$$

Three level proof

- Abstract level: pure real number computation
- Refinement step: reason about rounding errors in real arithmetic
- Implementation level: compute with large integers

Mathematical level

- Undergraduate mathematics (cf. T. Gowers)
- Requires improper Riemann integrals, uniform continuity
- No numeric integration (cf. A. Mahboubi)
- Automatic proofs: derivatives, algebraic formulas including square roots, beyond linear arithmetic
- Benefit from existing libraries (cf. M. Kohlhase)

Error evaluation

- Not interval computation: computing once
- Use directed rounding
- Interesting discovery: errors stay low in y_n
- Only 4 extra digits for 1 million digits with $\boldsymbol{\pi}_n$
- No error compensation for $\boldsymbol{\pi}_n'$
- Tools: interval (Melquiond) and psatzl (Besson)

Integer computation

- Using library bigZ (L. Théry et al.)
- Efficient square roots and division (B&Magaud)
- Systematic correspondence with Real operations, satisfying error model
- Computations in Coq, different levels of trust
- Faster computations with π'_n

Related work

- Initial work on Delaunay by Dufourd (hypermaps, Coq) also related to Gonthier's 4 colour theorem
- Meikle Delaunay formally verified (Isabelle)
- Voronoi Diagrams in Flyspeck (Hol-Light)
- Improper integrals based on filters cf. Harrison (Hol-Light), Hölzl, Immler, Huffmann (Isabelle)
- Laurence Rideau and Laurent Théry: Bailey-Borwein-Plouffe formula (Coq)
  - Also proved some time ago by J. Harrison (HOL-Light)
- Krebbers uses a Machin-like formula (Coq)

Thanks

- Coq infrastructure (cf. Coq acknowledgments)
- Libraries matter a lot
    - Social aspects of development and maintenance
- Mathematical components (Gonthier, Cohen, Mahboubi, Rideau, Théry, Tassi), Finite maps and sets (Cohen), Coquelicot (Boldo, Lelay, Melquiond, Sibut-Pinote), Interval (Melquiond), bigZ (Théry), psatzl (Besson)