Student Projects

In this file I sketch ideas for student projects with our ECG-project. For each project a short description is given. For each project, I give time estimates. In these estimates, I assume that the student works half-time on the project, i.e., the label one week corresponds to 20 hours of work.

Contents

1 The Diamond Operator for LEDA-Reals

LEDA reals support exact computations with algebraic numbers. At the moment, we can handle all numbers represented by expressions with integer operands and operators $+$, $-$, $*$, $/$, and $\sqrt[k]{\ }$ for integer k. In a recent paper [BFM$^+$01] we showed that the algorithmic approach taken in LEDA reals can also handle algebraic numbers defined as zeros of univariate polynomials. The task is to extend the implementation so as to provide the additional functionality.

The task involves several subtasks:

- understand the algorithmic approach taken in LEDA reals (1 week)
- implement an arbitrary precision algorithm for finding all roots of a univariate polynomial, e.g., simultaneous Newton iteration. (4 weeks).

- implement various methods for roots bounds of univariate polynomials (1 week)
- integrate the new algorithms into the LEDA real implementation (2 weeks)

As described, the task is a FOPRA. It can be extended to a Diplomarbeit; this would require the implementation and documentation of better methods for the two middle items. It can also be extended to a PhD thesis; this would require to either invent better methods for either item or to improve the analysis of the existing methods. For example, none of the existing practical methods guarantees global convergence. There are methods which come with a global convergence proof and are even efficient in the worst case. However, they do not seem to be efficient in practice.

Prerequisites: basic knowledge of algebra and numerical analysis

Here is small idea for the third item. The following root bound can be found in [**?**].

Lemma 1 *Let $p(x) = \sum_{0 \le i \le n} a_i x_i$ be a univariate polynomial with non-vanishing constant term and let $(\lambda_i)_{1 \le i \le n}$ be a set of non-negative real numbers whose sum is at most one, i.e., $\lambda_i \ge 0$ and $\sum_i \lambda_i \le 1$. Then*

$$|z| \ge \beta = \min \left\{ \left(\frac{|a_0|\lambda_i}{|a_i|} \right)^{1/i} ;\, a_i > 0 \text{ and } 1 \le i \le n \right\}$$

for any root z of p.

Proof: Observe that the definition of β implies $|a_i|\beta^i/|a_0| \le \lambda_i$ for all $i \ge 1$. Consider any z with $|z| < \beta$. Then

$$\begin{aligned} |p(z)| &= |\sum_i a_i z^i| = |a_0(1 + \sum_{1 \le i \le n} \frac{a_i z^i}{a_0})| \\ &> |a_0|(1 - \sum_{1 \le i \le n} \frac{|a_i|\beta^i}{|a_0|}) \\ &\ge |a_0|(1 - \sum_{1 \le i \le n} \lambda_i) \\ &\ge 0\ . \end{aligned}$$

∎

Specific roots bounds can now be obtained by special choices of the λ_i, for example $\lambda_i = 2^{-i}$ for $1 \le i \le n$ or $\lambda_i = 1/n$ for $1 \le i \le n$. In this way, one obtains root bounds which are easy to state, but are not necessarily the best possible.

An alternative would be to solve the non-linear optimization problem

$$\max \min \left(\frac{|a_0|\lambda_i}{|a_i|} \right)^{1/i} \qquad \text{subject to} \quad \lambda_i \ge 0 \quad \text{and} \quad \sum_i \lambda_i \le 1\ .$$

This is equivalent to

$$\max y \qquad \text{subject to} \quad y \le \left(\frac{|a_0|\lambda_i}{|a_i|}\right)^{1/i} \quad \text{and} \quad \lambda_i \ge 0 \quad \text{and} \quad \sum_i \lambda_i \le 1 \ .$$

We do not change the problem if we require equality in the last inequality, i.e., $\sum_i \lambda_i = 1$. Also, in an optimal solution we will have $\lambda_i = 0$ if $a_i = 0$. The remaining λ_i are chosen such that all bounds for y are identical (assume there are unequal bounds; by changing the λ_i slightly, we can get a better bound. Thus, we have $(|a_0|\lambda_i/|a_i|)^{1/i} = (|a_0|\lambda_j/|a_j|)^{1/j}$ whenever $a_i \ne 0 \ne a_j$. Fix a particular, i with $a_i \ne 0$ and express every other λ_j in terms of λ_i, i.e., $\lambda_j = \sqrt[i]{\frac{|a_0|^{j-i}\lambda_i^j |a_j|^i}{|a_i|^j}}$. This is particularly simple, if $i = 1$. I guess, in general, one should take i as small as possible. We now substitute these values into the equation $\sum_l \lambda_l$ and obtain a single equation for λ_i. This equation is easily solved by Newton method. Observe that all powers of λ_i have positive coefficient.

2 Restricted Diamond Operators

We extend LEDA reals by additional operators which give the real roots of equations of degree two, three and four respectively. For degree two there is nothing to do. For degrees three and four two approaches can be taked.

One either uses the known formulae for the roots of an equation of degree three or four. I looked at this for degree three. A linear transformation brings the equation into the form $x^2 + 3px + 2q = 0$. The discrimant $D = q^2 + p^3$ determines the number of real roots. If $D \le 0$ there are three real roots and if $D > 0$, there is a real root and a pair of conjugate complex roots. In the latter case the real root is $u + v$ where $u = \sqrt[3]{q + \sqrt{q^2 + p^3}}$ and $v = \sqrt[3]{q - \sqrt{q^2 + p^3}}$. The other two roots are $u\epsilon_1 + v\epsilon_2$ and $u\epsilon_2 + v\epsilon_1$, where $\epsilon_{1,2} = -1/2 \pm i\sqrt{3}/2$. The same expressions are valid if the discrimiant is negative. Then all real roots are expressed as sums of complex numbers. There seems to be no way around this.

I have not looked at the equation of degree four.

The other approach is purely numerical. We only need to provide a numerical method for computing the real roots and a rules which computes the separation bound. We can follow the approach outlined in the preceding section. It might be simpler to do it for degree three and degree four equation than for general equations.

3 Trapezoidal Decomposition

We have a sweep line algorithm to compute the arrangement defined by a set of lines and/or circular arcs. The sweep line paradigm can also be used to compute the trapezoidal decomposition. The task is to extend the implementation such that it alternatively computes the trapezoidal decomposition. The task involves several subtasks:

- understand the sweep algorithm (2 weeks)
- precisely define the trapezoidal decomposition in the presence of degeneracies (2 weeks)
- extend the implementation (3 weeks) for straight-line segments

The task can be extended in several directions.

- do the same for circular arcs
- extend the sweep algorithm so that it can handle conics (this is a master thesis)
- explore alternative algorithms, e.g., Mulmuley's method.

4 Nef Polyhedra

A Nef Polyhedron is any set that can be obtained from halfspaces by boolean operations (complement, intersection, union). The theory of Nef polyhedra was developed by Nef [Nef78] and later refined by Bieri and Nef and a sequence of papers [BN88]. Michael Seel developed data structures and algorithms for two-dimensional Nef polyhedra in his Ph-D thesis. The thesis contains further references. The task is to tackle three-dimensional Nef-polyhedra. This is a master thesis, if an inefficient solution is acceptable (i.e., the functionality must be provided, but efficiency is of no concern) and this is a PhD-thesis if an efficient solution is aimed for. Michael Seel already wrote the specification of a data structure for three-dimensional Nef polyhedra. The paper [DMY93] describes an efficient intersection algorithm if one the polyhedra is convex.

5 Nef Polyhedra with Curved Boundaries

The task is an in the previous section, but now we do not start with halfspaces but with more complex surfaces, e.g., all surfaces of degree two (circles, ellipsoids, parabolas, hyperbola) or Bezier surface patches. The task is to develop a theory of such polyhedra (what are the local properties of the objects that can be obtained in this way), to develop data structures for representing them, and algorithms for operating on these representations. As a starting point, one can take Nef's theory [Nef78] and the work done in the CAD-community [Hof89, HR96, Ros97, RO90, RR91].

This is a PhD project.

6 Polynomials

For our work we will need the class univariate polynomial. It has to support a large class of operations and should work with coefficients from different field types. Some desirable operations are: construction of polynomials from a sequence of coefficients, evaluation of polynomials, construction of polynomials from argument-value-pairs (interpolation), determine the number of real roots (Sturm sequences), determine the number of complex roots in the upper half plane, compute all roots with prescribed precision, separate the roots, determine a particular root with prescribed precision, compute initial approximations for the roots, compute various kinds of resultants, ...).

We will also have to study systems of polynomials: we need resultants, deciding whether there are only isolated roots, determining all roots, curve following if there is a continuous set of roots, ...

This task is almost unbounded and needs to be specified further. A good starting point are univariate polynomials and their roots.

7 Separation Bounds and Zero Tests

Separation bounds are the basis for numerical methods for determining signs of algebraic numbers. Algebraic numbers can be defined in different ways: as the values of algebraic expressions as in LEDA reals, as roots of univariate polynomials, as roots of systems of polynomials. Different separation bounds are available for different cases, e.g., [BFM+01, LY01] for algebraic expressions, the bound mentioned in Section 1 for univariate polynomials (there are many others), and Canny's bound [Can87] for solutions of systems of polynomials. All these bounds can be improved. Observe that the work of the last 10 years concentrates on algebraic expressions and has ignored the more general situations. We have to back to them.

This is a PhD thesis. A nice master thesis would be to survey the existing work and to give a more readable account of Canny's work.

8 Zero Test of Algebraic Expressions

For non-zero algebraic numbers the sign can be determined by numerical computation. The separation bound is needed to conclude that a number is zero.

An alternative to separation bound would be an efficient test whether a number is zero or non-zero. For sums of radicals, Blömer [Blo98] has given such a test. The challenge is to extend it to a more general class of expressions. Two candidates are

- sums of roots of bounded degree univariate polynomials. This is probably doable and would be a nice master thesis.
- arbitrary algebraic expressions. This is a PhD topic.

It is also worthwhile to investigate special cases. For example, if we have two equal circle points and the defining lines used to define the circle points are distinct, the circle point is actually a rational point (namely the intersection of two rational lines). This could simplify matters.

9 Voronoi Diagrams of Line Segments

Michael Seel developed and implemented an algorithm for Voronoi diagrams of line segments in his master's thesis [See94]. The implementation makes use of LEDA reals. In fact, it was the first program that used LEDA reals and it is still an impressive application of reals. The program is by now seven years old.

Given today's knowledge, the program uses the naive way and is therefore quite slow. The task is to redesign the program and to make it hundred times faster. To carry out the task requires

- understand the algorithm for randomized incremental constructions of Voronoi diagrams (2 weeks)
- understand Michael's implementation (2 weeks)
- understand the concept of floating point filters and study the optimizations used in Susan Hert's sweep line algorithm for circle arcs (4 weeks)
- design a new representation of Voronoi diagrams for line segments and design improved algorithms for the basic primitives (4 weeks)
- implement the improved algorithms (8 weeks)
- modify Michael's implementation so as to make use of the new primitives (4 weeks)

The task requires significant theoretical work. It is suitable as a FOPRA or a Diplomarbeit.

10 High-Level Filtering and Analysis of Algorithms

Stefan Funke and I proposed to apply filtering techniques to high-level predicates [FMN99]. A high-level predicate determines, for example, the triangle of a triangulation containing a particular query point. See the slides for my presentation in Leiden on my home page for more details.

In our paper we assumed naively that the time bounds for searches in various geometric data structures stay valid in the presence of incorrect primitives. It was pointed out to us at the Leiden meeting that this is not true. The analysis of randomized incremental construction uses geometric reasoning and this reasoning is no longer valid if imprecise primitives are used. Is there a way to remedy the situation in general, in particular cases?

As a starting point, I would look at the following examples:

- incremental two-dimensional convex hull as for example described in the LEDA book
- incremental 3d hull and dd hull as described by Clarkson and Shor with either conflict lists or history graphs [BY98]
- incremental hull algorithms maintaining a triangulation [CMS93].
- incremental Delaunay and Voronoi diagram algorithms

This is probably not suitable for a student project as several groups will attempt to solve problem given the reaction at the Leiden meeting.

11 Fast Integer Multiplication and Division

Multiplication and division of arbitrary length integers is at the heart of any system for exact computation. The LEDA routines are among the best available. We use the method of Karazuba-Ofman with asymptotic running time $O(n^{\log 3})$.

The applications in ECG will require us to handle even larger integers than in the past and hence asymptotically faster multiplication and division algorithms could gain importance. The task is to implement an FFT-based multiplication algorithm.

I suggest this to Arno Eigenwillig as a FOPRA.

References

[BFM+01] C. Burnikel, S. Funke, K. Mehlhorn, S. Schirra, and S. Schmitt. A separation bound for real algebraic expressions. Technical report, 2001. ESA 2001, to appear, `www.mpi-sb.mpg.de/\~{}mehlhorn/ftp/ImprovedSepBounds.ps.gz`, companion pages.

[Blo98] J. Bloemer. A probabilistic zero-test for expressions involving roots of rational numbers. *Lecture Notes in Computer Science*, 1461:151–??, 1998.

[BN88] H. Bieri and W. Nef. Elementary set operations with d-dimensional polyhedra. In Hartmut Noltemeier, editor, *Proceedings of the International Workshop on Computational Geometry and its Applications (CG'88)*, volume 333 of *Lecture Notes in Computer Science*, pages 97–112, 1988.

[BY98] J.-D. Boissonnat and M. Yvinec. *Algorithmic Geometry*. Cambridge University Press, Cambridge, 1998.

[Can87] J.F. Canny. *The Complexity of Robot Motion Planning*. The MIT Press, 1987.

[CMS93] K. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. *Computational Geometry: Theory and Applications*, 3:185–212, 1993.

[DMY93] K. Dobrindt, K. Mehlhorn, and M. Yvinec. A complete and efficient algorithm for the intersection of a general and a convex polyhedron. In Frank Dehne, Jörg-Rüdiger Sack, Nicola Santoro, and Sue Whitesides, editors, *Proceedings of the 3rd International Workshop on Algorithms and Data Structures (WADS'93)*, volume 709 of *Lecture Notes in Computer Science*, pages 314–324, Montréal, Canada, 11–13 August 1993. Springer.

[FMN99] Stefan Funke, Kurt Mehlhorn, and Stefan Naeher. Structural filtering: A paradigm for efficient and exact geometric algorithms. In *Proceedings of the 11th Canadian Conference on Computational Geometry*, pages 39–42, 1999. `www.mpi-sb.mpg.de/~funke/Struct\_long.ps.gz`.

[Hof89] C.M. Hoffmann. *Geometric and Solid Modeling : An Introduction. 2nd pr. 1993*. Series in computer graphics and geometric modeling. Morgan Kaufmann, 1989.

[HR96] C.M. Hoffmann and J.R. Rossignac. A road map to solid modeling. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):3–10, March 1996.

[LY01] C. Li and C. Yap. A new constructive root bound for algebraic expressions. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'01)*, pages 496–505, 2001.

[Nef78] W. Nef. *Beitraege Zur Theorie Der Polyeder Mit Anwendungen in der Computergraphik (Contributions to the Theory of Polyhedra, with Applications in Computer Graphics)*. Verlag H. Lang & Cie. AG, Bern, 1978.

[RO90] J.R. Rossignac and M.A. O'Connor. SGC: A dimensional-independent model for pointsets with internal structures and incomplete boundaries. In M.J. Wozny, J.U. Turner, and K. Preiss, editors, *Geometric Modeling for Product Engineering*, pages 145–180. Elsevier Science Publishers B.V., North Holland, 1990.

[Ros97] J. Rossignac. Structured topological complexes: A feature-based API for non-manifold topologies. In Christoph Hoffmann and Wim Bronsvort, editors, *Proceedings of the 4th Symposium on Solid Modeling and Applications*, pages 1–9, New York, May14–16 1997. ACM Press.

[RR91] J.R. Rossignac and A.A.G. Requicha. Constructive non-regularized geometry. *Computer Aided Geometry*, 23(1):21–32, 1991.

[See94] Michael Seel. Eine Implementierung abstrakter Voronoidiagramme. Master's thesis, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, 1994.