

Sujet 1

Root solver using subdivision and Fast Fourier Transform

Proposé par : Guillaume Moroz and Marc Pouget

General information

Encadrants	Guillaume Moroz	Marc Pouget
Adresse	LORIA, Campus Scientifique - BP 239, 54506 Vandœuvre-lès-Nancy	
Email	guillaume.moroz@inria.fr	marc.pouget@inria.fr

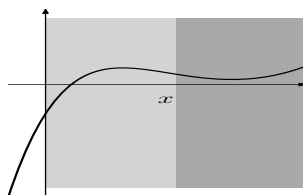
Motivations

In scientific computation, finding the roots of polynomial equations is a fundamental problem arising in robotics or visualization for example. Bisection is a standard numerical method to solve this problem [1, §2.1]. On the other hand, it requires to evaluate the input polynomials at each bisection step, and for a high degree polynomial p , evaluating p can be costly. Approaches based on Fast Fourier Transform can be used to evaluate a polynomial on several point for the cost of one evaluation [2, §8.2, Cor.10.8]. These approaches have never been used in numerical root-finding methods. The goal of this internship is to use multipoint evaluation approaches to improve significantly root-finding methods for univariate and bivariate polynomial equations.

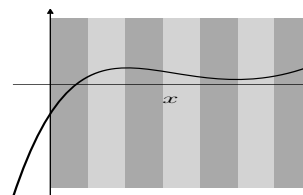
Subject

Let $p(x)$ be a polynomial of degree d with regular roots. For a interval I small enough, it is possible to decide if p has a root or not in I using evaluations. The bisection method consists in splitting the initial domain in two, recursively, until the intervals are small enough to decide if they contain a root of p or not.

Instead of splitting the initial domain in two parts (left figure), we would like to split it in d parts (right figure). Indeed, using Fast Fourier Transform, evaluating p in one point costs almost the same as evaluating p in d points [2, Cor. 10.8].



Bisecting the x -domain in two parts



Splitting the x -domain in d parts

During the internship, the candidate will first need to become familiar with interval arithmetic, subdivision and Fast Fourier Transform methods. Then, using existing interval arithmetic and Fast Fourier Transform library, the candidate will design and experiment new subdivision methods, and compare them with state-of-the-art approaches [Arb library¹ or Sage²]. This work will be first focused on univariate polynomials, and it may be extended to bivariate polynomial systems.

Profile of the candidate

The candidate should have a taste for both mathematics numerical analysis and computer science. Programming skills would be appreciated.

Références

- [1] Burden, R., and J. Faires. 2010. *Numerical Analysis*. Cengage Learning. <http://ins.sjtu.edu.cn/people/mtang/textbook.pdf>.
- [2] Joachim von zur Gathen and Jürgen Gerhard. 2013. *Modern Computer Algebra*. 3rd ed. New York, USA : Cambridge University Press.

1. <http://fredrikj.net/arb>

2. <https://hal.archives-ouvertes.fr/hal-00846961v3>

Sujet 2

Point removal in 3D Delaunay triangulation

Informations générales

Encadrants Olivier Devillers
Adresse LORIA, Campus Scientifique
 BP 239, 54506 Vandœuvre-lès-Nancy
Téléphone 03 54 95 84 97
Email olivier.devillers@inria.fr
Bureau B 168

Context and motivation

Geometric problems are central in many areas of science and engineering. Computational geometry, the study of combinatorial and algorithmic problems in a geometric setting, and in particular triangulations have tremendous practical applications in areas such as computer graphics, computer vision and imaging, scientific visualization, geographic information systems, astronomy, computational biology... Traditionally, the complexity of computational geometry algorithms is studied in the worst case setting. This kind of analysis is often quite pessimistic compared to real life data. Due to the emergence of standardized software libraries, in particular the Computational Geometry Algorithms Library CGAL, developed in the framework of an Open Source Project, the so-far mostly theoretical results developed in computational geometry are being used and extended for practical use like never before. CGAL has been proposing efficient and robust packages computing Delaunay triangulations in the 2D and 3D Euclidean spaces for years.

Objectives

Currently, inserting a point in a triangulation is 10 times cheaper than removing it. The goal of this research is to obtain a better algorithm in practice to remove a point. A fast deletion operation is crucial in several contexts. For example in mesh optimization, or in simulation of moving objects.

Knowledge involved :

A good level in C++ (templates, etc) is necessary.

Working program

When removing a point and all incident tetrahedra in a Delaunay triangulation a hole is created. This hole must be retriangulated without the deleted point.

Even if this is not the best theoretical solution, the most efficient algorithm (among several benchmarked possibilities) currently implemented in CGAL is the following :

- compute a small triangulation with only the hole vertices,
- recognize the hole boundary in that triangulation,
- throw away the tetrahedra of the small triangulation outside the hole, and
- put relevant pointers to glue the remaining part of the small triangulation in the hole of the big one.

This algorithm has the drawback of computing stuff that is soon thrown away. Actually, using an elegant definition for the boundary of the triangulation it is possible to compute only the relevant tetrahedra. This new definition should also facilitate the gluing operation [1].

The candidate will finalize and code the new deletion algorithm inside CGAL, conducts benchmarks about its efficiency.

Références

- [1] Kevin Buchin, Olivier Devillers, Wolfgang Mulzer, Okke Schrijvers, and Jonathan Shewchuk. Vertex Deletion for 3D Delaunay Triangulations. In *Proc. 21th European Symposium on Algorithms*, volume 8125 of LNCS, pages 253–264, 2013. <https://hal.inria.fr/hal-00832992>

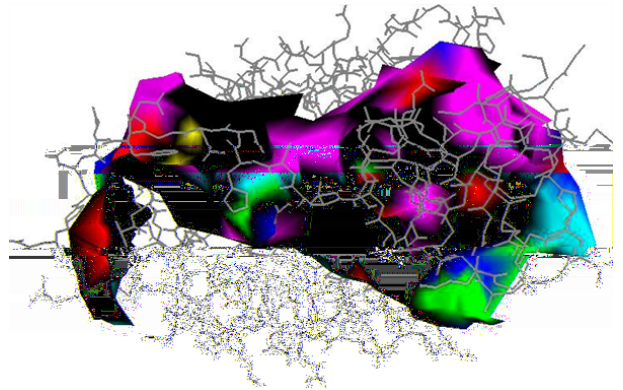
Sujet 1

Intersurf++

Proposé par : Dave Ritchie

Informations générales

Encadrants Dave Ritchie
O. Devillers, B. Maigret, Monique Teillaud
Adresse LORIA, Campus Scientifique
BP 239, 54506 Vandœuvre-lès-Nancy
Téléphone 03 83 59 30 45
Email Dave.Ritchie@inria.fr
Bureau B 166



Motivations

Studying protein-protein complexes in 3D can help to understand many fundamental biological processes.

Topic

Intersurf++ : Calculating 3D interfaces between protein molecules

Context