

# Comment faire des preuves exactes sur des polynômes à l'aide de calculs en flottants

David Monniaux  
chargé de recherche au CNRS, laboratoire VERIMAG  
<http://www-verimag.imag.fr/~monniaux/>  
David.Monniaux@imag.fr

8 mars 2010

**Lieu du stage :** Laboratoire VERIMAG, 2 avenue de Vignate, 38610 Gières (banlieue de Grenoble)

**Sujet :** Comment prouver qu'un polynôme est positif ? Comment montrer qu'un système d'inégalités polynomiales n'a pas de solutions ? Comment borner un polynôme sur un compact ?

Ces questions ont une importance pratique :

- preuves d'invariants de programme
- preuves formelles à la Coq
- preuves de théorèmes mathématiques (p.ex. les lemmes de la preuve de la conjecture de Kepler<sup>1</sup> par Thomas Hales)

On se propose d'utiliser, via des théorèmes dits « Positivstellensatz », une réduction de ces problèmes vers la programmation semidéfinie,<sup>2</sup> c'est-à-dire la recherche d'une matrice semidéfinie positive (valeurs propres positives ou nulles) parmi un ensemble  $-F_0 + \sum_i \lambda_i F_i$ .

Il existe des algorithmes et des outils<sup>3</sup> qui permettent de résoudre ce problème numériquement. Malheureusement, ces algorithmes sont inadaptés aux cas où l'ensemble des solutions est géométriquement dégénéré (plat).

L'encadrant du stage a proposé en 2010 un algorithme permettant de « rectifier » des résultats numériques à l'aide de l'algorithme LLL.<sup>4</sup> L'implémentation actuelle, si elle donne des résultats intéressants, souffre encore de certaines inefficacités. L'objet du stage serait de réfléchir à des améliorations et des les programmer, notamment par usage judicieux de bibliothèques d'algèbre linéaire creuse, soit exacte,<sup>5</sup> soit flottante.

---

<sup>1</sup><http://code.google.com/p/flyspeck/>

<sup>2</sup><http://www.stanford.edu/~boyd/cvxbook/>

<sup>3</sup><http://www.mcs.anl.gov/DSDP/>

<sup>4</sup>Algorithme de réduction de bases de réseaux dû à Lenstra, Lenstra et Lovász, voir par exemple [http://en.wikipedia.org/wiki/LLL\\_algorithm](http://en.wikipedia.org/wiki/LLL_algorithm)

<sup>5</sup><http://www.linalg.org/>