Read Me

Tom Baumbach

February 24, 2023

This is a prototype for solving sparse polynomial systems. We present the code in order to solve such systems. In order to run this code the user also needs to install SeDuMi [1].

The code contains two main functions. Both functions are capable of solving polynomial systems, if the number of solutions defined by these polynomial equations is finite. Otherwise the algorithms may not terminate.

But before we go into more detail about the two main functions, we want to talk about the unit *tol*. The unit *tol*, standing for tolerance, can be chosen freely and allows the user to chose how accurate the solutions should be. The smaller *tol* is, the more accurate is the solution. But there is a tradeoff, namely if *tol* is chosen too small, the software SeDuMi might not find a solution or runs into numerical errors. From the experiments we think that a good choice for *tol* is between 10^{-3} and 10^{-10} and we suggest that it is best to start with a bigger tolerance and decrease *tol* overtime.

1 Moment_Method_A

This function computes the real roots contained in $(\mathbb{R} \setminus \{0\})^n$ of a given sparse polynomial system $f_1 = \cdots = f_m = 0$, where $f_i \in \mathbb{R}[\underline{x}^{\pm 1}]$, taking the structure of the polynomials into account.

The function needs four different inputs:

1) Polynomials $f_1, \ldots, f_m$ written in the form $\{f_1, ..., f_m\}$, and each polynomial $f_i = \sum_j^k c_j x^{\alpha_j}$,$\alpha_i \in \mathbb{Z}^n$ is represented as $f_i = \{\{c_1, [\alpha_1]\}, ..., \{c_k, [\alpha_k]\}\}$.

2) The set $\mathcal{A} \subset \mathbb{Z}^n$ such that the semi-group generated by $\mathcal{A}$ is equal to $\mathbb{Z}^n$. (The zero vector must always be the first vector).

3) The number of variables x_i.

4) The tolerance, denoted by tol, for the accuracy of the solutions.

Example 1.1. *In Example1.m we consider the polynomials*

$$f_1 = x^{-1}y - 1 \text{ and } f_2 = x^2 - 2.$$

We can see in Example1.m how to represent these two polynomials the other units. The output is a matrix whose row vectors are the solutions of the system $f_1 = f_2 = 0$.

2 Moment_Method_A_standard

This function computes the real roots contained in $\mathbb{R}^n$ of a given sparse polynomial system $f_1 = \cdots = f_m = 0$, where $f_i \in \mathbb{R}[x_1, \ldots, x_n]$, taking the structure of the polynomials into account.

The function needs four different inputs:

1) Polynomials $f_1, \ldots, f_m$ written in the form $\{f_1, ..., f_m\}$, and each polynomial $f_i = \sum_j^k c_j x^{\alpha_j}, \alpha_i \in \mathbb{N}^n$ is represented as $f_i = \{\{c_1, [\alpha_1]\}, ..., \{c_k, [\alpha_k]\}\}$.

2) The set $\mathcal{A} \subset \mathbb{N}^n$, represented as row vectors, such that the semi-group generated by $\mathcal{A}$ is equal to $\mathbb{N}^n$. (The zero vector must always be the first vector).

3) The number of variables x_i.

4) The tolerance, denoted by tol, for the accuracy of the solutions.

Example 2.1. *In Example2.m we consider the polynomials*

$$f_1 = x^2y^{11} - x, f_2 = x^3 - y^4 \text{ and } f_3 = xy^2 + y^2 - 2.$$

We can see in Example2.m how to represent all the necessary inputs in order to solve the system $f_1 = f_2 = f_3 = 0$. *Again, the output is a matrix whose row vectors represent all the solutions of the polynomial system.*

References

[1] Jos F. Sturm. Using sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software, volume 11, pages 625-65, https://doi.org/10.1080/10556789908805766 3, doi = 10.1080/10556789908805766*, 1999.