数理解析研究所講究録 1456 巻 2005 年 58-64

Development of SyNRAC—A Cylindrical Algebraic Decomposition Procedure

屋並 仁史

YANAMI HITOSHI*

(株) 富士通研究所/(独) 科学技術振興機構

FUJITSU LABORATORIES LTD./CREST, JST†

穴井 宏和

HIROKAZU ANAI

(株) 富士通研究所/(独) 科学技術振興機構

FUJITSU LABORATORIES LTD./CREST, JST‡

Abstract

We have been developing SyNRAC, a toolbox on Maple for solving real algebraic constraints derived from various engineering problems. In this paper we present newly implemented procedures in our toolbox. The current version of SyNRAC has added quantifier elimination (QE) by cylindrical algebraic decomposition (CAD) as well as QE by virtual substitution for quadratic formulas. We also show an application of CAD-based QE to the common Lyapunov function problem.

1 Introduction

Recently symbolic computation methods have been gradually applied to solving engineering problems, which has been caused by the efficient symbolic algorithms introduced and improved for these few decades and by the advancement of computer technology that has hugely increased the CPU power and memory capacity.

We have been developing SyNRAC, a Maple toolbox, for solving real algebraic constraints. This tool has been presented in 2003 [1] and in 2004 [2]. SyNRAC stands for a Symbolic-Numeric toolbox for Real Algebraic Constraints and is aimed at being a comprehensive toolbox including a collection of symbolic, numerical, and symbolic-numeric solvers for real algebraic constraints derived from various engineering problems. When we say a real algebraic constraint, what we have in mind is a first-order formula over the reals. Our main method is quantifier elimination (QE), which removes the quantified variables in a given formula to return a quantifier-free equivalent.

In this paper we present newly implemented procedures in SyNRAC. The current version of SyNRAC has added the following:

---

*This work was partially supported by CREST, the Japan Science and Technology Agency, Japan.

†yanami@labs.fujitsu.com

‡anai@jp.fujitsu.com

- a special QE procedure by virtual substitution for quadratic formulas
- a general QE procedure by cylindrical algebraic decomposition (CAD)

A special QE method for quadratic formulas has widened the application areas of SyNRAC in actual problems (see [3]). CAD-based QE is regarded as a general QE in the sense that it can deal with any type of formula, if the efficiency is ignored. Historically, CAD-based approaches preceded the special QE methods we had already implemented in SyNRAC. We implemented special QE first because there was a good class of formulas to which many practical problems could be reduced and a much more efficient special QE method was applicable [4, 5].

This paper is organized as follows. In Section 2 we present special QE by virtual substitution for quadratic formulas. We briefly describe CAD and show the commands on CAD in SyNRAC in Section 3. In Section 4 we show an example problem to which SyNRAC's CAD command can apply. We end with a conclusion in Section 5.

2 Solving quadratic algebraic constraints over the reals

We briefly explain a special QE by virtual substitution of parametric test points that is applicable to formulas in which the quantified variables appear at most quadratically (see [6] for details). We call a formula whose atomic subformulas are at most quadratic with respect to its quantified variables a *quadratic formula.*

Let

$$\psi(p_1,\ldots,p_m) \equiv Q_1x_1\cdots Q_nx_n\varphi(p_1,\ldots,p_m,x_1,\ldots,x_n)$$

be a linear or quadratic formula, where $Q_i \in \{\forall,\exists\}$ and φ is a quantifier-free formula. By using the equivalence $\forall x\varphi(x)\Longleftrightarrow\neg(\exists x\neg\varphi(x))$, we can change the formula into an equivalent formula of the form $(\neg)\exists x_1\cdots(\neg)\exists x_n\ (\neg)\varphi$. The possible negation '$\neg$' that precedes a quantifier-free formula can be easily eliminated (use De Morgan's law and rewrite the atomic subformulas), which is not essential part of quantifier elimination. Therefore we may focus our attention on an *existential formula,* i.e., a formula of the form $\exists x_1\cdots\exists x_n\varphi(p_1,\ldots,p_m,x_1,\ldots,x_n)$. Furthermore, it is sufficient to show how to eliminate $\exists x$ in $\exists x\varphi$, since all the quantifiers in the formula can be eliminated by removing one by one from the innermost one.

Now our main purpose is to eliminate the quantified variable $\exists x$ in

$$\exists x\ \varphi(p_1,\ldots,p_m,x),$$

with $\varphi(p_1,\ldots,p_m,x)$ quantifier-free and quadratic to obtain an equivalent quantifier-free formula. For fixed real values $q_1,\ldots,q_m$ for the parameters $p_1,\ldots,p_m$, the set $M=\{r\in\mathbb{R}|\varphi(q_1,\ldots,q_m,r)\}$ of real values r for x satisfying φ is a finite union of closed, open, and half-open intervals over $\mathbb{R}$, since all polynomials appearing in $\varphi(x)$ are linear or quadratic. The endpoints of these intervals are among $\pm\infty$ and the real zeros of atomic formulas in φ. Then candidate terms, say, $t_1,\ldots,t_k$, for those zeros can be constructed by the solution formulas for linear or quadratic equations.

If φ does not contain any strict inequalities, all the intervals composing M are either unbounded or closed. In the closed case such an interval contains its real endpoint. So M is nonempty if and only if the substitution of $\pm\infty$ or of one of the candidate solutions t_j for x satisfies φ. Let S be the candidate set $S = \{t_1, \ldots, t_k, \pm\infty\}$. Such a set is called an *elimination set* for $\exists x\varphi$. We obtain a quantifier-free formula equivalent to $\exists x\varphi$ by substituting all candidates in S into φ *disjunctively:*

$$\exists x\varphi \Longleftrightarrow \bigvee_{t\in S} \varphi(x//t).$$

We note that there is a procedure assigning the expression $\varphi(x/t)$ obtained from φ by substituting t for x an equivalent formula [6]. We denote the resulting formula by $\varphi(x//t)$. If φ contains strict inequalities, we need to add to S other candidates of the form $s \pm \epsilon$, where s is a candidate solution for some left-hand polynomial in a strict inequality and ϵ is a positive infinitesimal. For improving the efficiency of this method, the following two points are crucial: (i) refining the elimination set S by a scrupulous selection of a smaller number of candidates in S; (ii) integrating with sophisticated simplifications of quantifier-free formulas. SyNRAC now employs three types of elimination sets proposed in [7].

Moreover, (heuristic) techniques for decreasing the degree during elimination are important for raising the applicability of quadratic QE, because after one quantifier is eliminated for a quadratic case the degree of other quantified variables may increase. Only simple degree-decreasing functions are implemented in the current version of SyNRAC.

3 Cylindrical Algebraic Decomposition

Cylindrical algebraic decomposition (CAD) was discovered by Collins in 1973; see [8] for his monumental work. Collins also proposed a general QE algorithm based on CAD, which provided a powerful method for solving real algebraic constraints.

Let A be a finite subset of $\mathbb{Z}[x_1, \ldots, x_n]$. An *algebraic decomposition* for A is a collection of mutually disjoint, semi-algebraic, A-invariant sets that partitions the Euclidean n-space E^n. To define the term *cylindrical,* we explain three parts of a CAD procedure—the projection phase, the base phase, and the lifting phase.

In the projection phase of a CAD, the PROJ function plays a central role. Let r be an integer greater than 1. PROJ maps a finite set of integral polynomials in r variables to a finite set of integral polynomials in $r-1$ variables: for $A_r \subset \mathbb{Z}[x_1, \ldots, x_r]$, $\mathrm{PROJ}(A_r) \subset \mathbb{Z}[x_1, \ldots, x_{r-1}]$. For a given $A \subset \mathbb{Z}[x_1, \ldots, x_n]$, we obtain a list

$$A = A_0 \overset{\mathrm{PROJ}}{\mapsto} A_1 \overset{\mathrm{PROJ}}{\mapsto} A_2 \overset{\mathrm{PROJ}}{\mapsto} \cdots \overset{\mathrm{PROJ}}{\mapsto} A_{n-1},$$

where $A_i \subset \mathbb{Z}[x_1, \ldots, x_{n-i}]$.

In the base phase we partition E^1 by using a set of univariate polynomials $A_{n-1} \subset \mathbb{Z}[x_1]$; we find all the real zeros of A_{n-1} and partition E^1 into A_{n-1}-invariant regions that consist of the zeros of A_{n-1} and the remaining open intervals. These points and intervals are called sections and sectors, respectively.

The lifting phase inductively constructs a decomposition of E^{i+1} from the decomposition of E^i, $i = 1, 2, \ldots, n-1$. Suppose D is a decomposition of E^i. A lifting of D is a decomposition $\bar{D}$ of E^{i+1} obtained by decomposing the space $R \times E^1$ by using A_{n-i-1} for each region $R \in D$ and putting all of them together. Let R be a region of a decomposition D of E^i. $R \times E^1$ is decomposed by the following; Take a point $(p_1, \ldots, p_i)$ in R and substitute it for $(x_1, \ldots, x_i)$ in each polynomial in A_{n-i-1} to obtain a set of univariate polynomials in x_{i+1}; Partition E^1 into, say, L_0, L_1, ..., L_{2k+1} by using the roots of the polynomials in x_{i+1}; Regard $R \times L_0$, $R \times L_1$, ..., $R \times L_{2k+1}$ as the resulting decomposition. The condition for this process to work is that A_{n-i-1} is *delineable* on R, in other words, every pair of polynomials in

A_{n-i-1} has no intersections on R. In such a case the decomposition is independent of the choice of a sample point. A decomposition D of E^r is *cylindrical* if it is constructed by iterating the above lifting method, i.e., $r = 1$ and E^1 is decomposed as in the base phase, or $r > 1$ and D is a lifting of some cylindrical decomposition D' of E^{r-1}.

Given a formula φ one can construct a CAD for the polynomials of the atomic formulas in φ. The point for CAD-based QE is that the value (truth/false) of φ is determined regionwise because each region in the CAD is A-invariant. See [8] for details. It is the PROJ function that is crucial in a CAD procedure. The fewer polynomials PROJ produces, the more efficient the CAD program becomes. But PROJ must be constructed to maintain the delineability and make the lifting phase possible. Some improvements in the projection phase of CAD are found in [9, 10, 11].

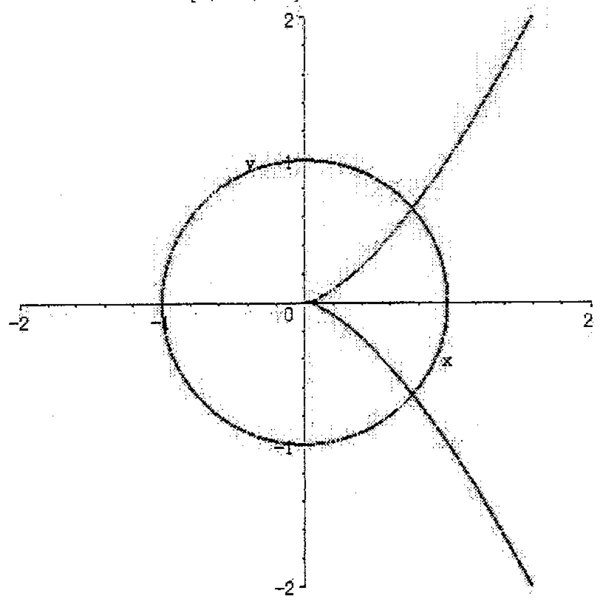

Figure 1: The graph of A

Here we show some examples of CAD commands in SyNRAC. We construct a CAD for $A := \{x^2 + y^2 - 1, x^3 - y^2\} \subset \mathbb{Z}[x, y]$. The graph of the two polynomials in A is shown in Fig. 1. The `Projection` command repeats PROJ and returns `P[1]`$= \mathrm{PROJ}^0(A) = A$ and `P[2]`$= \mathrm{PROJ}(A)$.

```
> read "synrac";
> A:=[ x^2 + y^2 - 1, x^3 - y^2 ]:
> P:=Projection(A, [y,x]):
> P[1]; P[2];

                [x^2 + y^2 - 1, x^3 - y^2]

               [x^2 - 1, x^2 - 1 + x^3, x]
```

Next the `Base` command partitions E^1 by using `P[2]` and returns a list of points that represent respective sections or sectors. A rational point is taken as a sample point for a sector, and a vanishing polynomial and an isolated interval are taken for a section. There are four real roots (sections) in `P[2]` and they make five open intervals (sectors).

```
> Base(P[2], x);

  [-2, [x + 1, [-1, -1]], -1/2, [x, [0, 0]], 3/8,
 [x^2 - 1 + x^3, [3/4, 7/8]], 15/16, [x - 1, [1, 1]], 2]
```

Lastly the `Lifting` command makes a stack for each section or sector. Out of nine regions, we have the third and the fifth ones displayed. The third region is a sector with a rational sample point `[-1/2]` and the stack on it is represented in a list of five sample points of sections/sectors. The data for the fifth sector are shown in a similar way. The data for It is similar for the fifth sector, which has `[3/8]` as a sample point.

```
> L:=Lifting(P, [y,x]):
> op(L[3]); op(L[5]);

  [3], [-1/2], [-2, [-3 + 4 y^2, [-1, 0]], 0, [-3 + 4 y^2, [0, 1]], 2]

  [5], [3/8], [-2, [-55 + 64 y^2, [-1, -1/2]], -1/2,
  [512 y^2 - 27, [-1/2, 0]], 0, [512 y^2 - 27, [0, 1/2]], 1/2,
  [-55 + 64 y^2, [1/2, 1]], 2]
```

4 Application of CAD-based QE

In this section we show an application of CAD to a practical problem. The common Lyapunov function problem is a problem that studies the existence of a common Lyapunov function for a set of linear time-invariant systems. The problem often arises in stability analysis and control design of various types of control systems such as uncertain systems, fuzzy systems, switched systems, etc. We focus on one of the main types of common Lyapunov functions, a *common quadratic Lyapunov function* (CQLF). See [12, 13, 14] for details.

One important issue of a CQLF problem is to find an existence condition of CQLF. For a given set of stable constant linear systems, we can verify whether the systems share a CQLF and construct the CQLF if they do with some numerical semidefinite programming (SDP) package. We consider here the problem of finding symbolic existence conditions on system matrices such that these systems share a common Lyapunov function. An existence condition provides us stability regions of parameters for control systems. Although there are some attempts to resolve this so far, only partial results are obtained. The CQLF problems to compute an existence condition can be solved by using QE systematically; see [15].

Common Lyapunov Function Problem: We consider a set of continuous-time linear time-invariant systems

$$\dot{\mathbf{x}} = \mathbf{A}_{ci}\mathbf{x},\ \mathbf{x} \in \mathbb{R}^n,\ \mathbf{A}_{ci} \in \mathbb{R}^{n\times n},\ i = 1,\ldots,q. \tag{1}$$

The set of systems (1) is said to have a CQLF if there exists a symmetrical positive definite matrix $\mathbf{P} = \mathbf{P}^T > 0$ such that the following Lyapunov inequalities

$$\mathbf{P}\mathbf{A}_{ci} + \mathbf{A}_{ci}^T\mathbf{P} < 0,\ \forall i = 1,\ldots,q \tag{2}$$

are satisfied. Then the CQLF is $\mathbf{V}(\mathbf{x}) = \mathbf{x}^T\mathbf{P}\mathbf{x}$.

Solving Common Lyapunov Function Problem by QE: We consider two Hurwitz stable continuous-time linear time-invariant systems, i.e., the case $n = 2$ and $q = 2$ in (1):

$$\dot{\mathbf{x}} \ = \ \mathbf{A}_{ci}\mathbf{x},\ \mathbf{x} \in \mathbb{R}^2,\ \mathbf{A}_{ci} \in \mathbb{R}^{2\times 2},\ i = 1,2. \tag{3}$$

Moreover let $\mathbf{A}_{c1}, \mathbf{A}_{c2}$ be

$$\mathbf{A}_{c1} = \begin{bmatrix} x & 0 \\ y & -1 \end{bmatrix},\ \mathbf{A}_{c2} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix}, \tag{4}$$

respectively, where $x, y \in \mathbb{R}$ are system parameters. We have the following theorem.

Theorem 1
(R.N.Shorten et.al. [13]) A necessary and sufficient condition for the two second-order systems (3) to have a CQLF is

$$Re(\lambda(co(\mathbf{A}_{c1}, \mathbf{A}_{c2}))) < 0, \quad (5)$$

$$Re(\lambda(co(\mathbf{A}_{c1}, \mathbf{A}_{c2}^{-1}))) < 0, \quad (6)$$

where co(.) denotes the convex hull (polytope) of matrices: $co(\mathbf{X}, \mathbf{Y}) = \{\alpha\mathbf{X} + (1-\alpha)\mathbf{Y} : \alpha \in [0,1]\}$, $\lambda(\mathbf{X})$ *denotes the eigenvalues of matrix* $\mathbf{X}$ *and Re(.) denotes the real part of a complex number.*

This implies that our desired condition is that all roots of characteristic polynomials of $\mathrm{co}(\mathbf{A}_{c1}, \mathbf{A}_{c2})$ and $\mathrm{co}(\mathbf{A}_{c1}, \mathbf{A}_{c2}^{-1})$ locate within a left half of the Gaussian plane for $\alpha \in [0,1]$. The conditions can be reduced to a set of polynomial inequalities by using the well-known Liènard-Chipart criterion. Then we can apply QE for the polynomial inequalities.

Now we compute feasible regions of x, y so that the systems (3) have a CQLF by QE. As mentioned above, by applying the Liènard-Chipart criterion to characteristic polynomials of $\mathrm{co}(\mathbf{A}_{c1}, \mathbf{A}_{c2})$ and $\mathrm{co}(\mathbf{A}_{c1}, \mathbf{A}_{c2}^{-1})$, the CQLF existence condition can be described by the following formula:

$$\begin{aligned} \forall\alpha \ \ ((0 \le \alpha \le 1) \Rightarrow \ \ & (2 - \alpha - x\alpha > 0 \ \wedge \ 1 + y\alpha - \alpha^2 - x\alpha^2 - y\alpha^2 > 0 \ \wedge \\ & 1 - 2\alpha - 2x\alpha - y\alpha + \alpha^2 + x\alpha^2 + y\alpha^2 > 0)). \end{aligned} \quad (7)$$

Applying CAD-based QE in SyNRAC to (7), the output turns out to be

$$x < 0 \ \wedge \ -2 - 2\sqrt{-x} < y \ \wedge \ y < 2\sqrt{-x} - 2x. \quad (8)$$

5 Conclusion

We have presented a newly developed procedure in Maple-package SyNRAC. The current version of SyNRAC provides quantifier elimination (QE) by virtual substitution up to quadratic formulas and a CAD-based QE procedure, as well as some standard simplifiers. The new added CAD command greatly extends the applicability and tractability of SyNRAC for solving real algebraic constraints in engineering. As an application of SyNRAC, we have treated an example of the common Lyapunov function problem.

We proceed to implement other known QE algorithms and improve them, and are setting about developing symbolic-numeric algorithms. We also plan to develop a toolbox for parametric robust control design on MATLAB using SyNRAC as a core engine.

References

[1] Anai, H., Yanami, H.: SyNRAC: a Maple-package for solving real algebraic constraints. In: Proceedings of International Conferences on Computational Science. Volume 2657 of LNCS., Springer (2003) 828–837

[2] Yanami, H., Anai, H.: Development of SyNRAC—Formula description and new functions. In: Proceedings of International Conferences on Computational Science. Volume 3039 of LNCS., Springer (2004) 286–294

[3] Dolzmann, A., Sturm, T., Weispfenning, V.: Real quantifier elimination in practice. In Matzat, B.H., Greuel, G.M., Hiss, G., eds.: Algorithmic Algebra and Number Theory. Springer, Berlin (1998) 221–247

[4] Anai, H., Hara, S.: A parameter space approach for fixed-order robust controller synthesis by symbolic computation. In: Proceedings of IFAC World Congress on Automatic Control b'02. (2002)

[5] Anai, H., Yanami, H., Hara, S.: SyNRAC: a Maple-package for solving real algebraic constraints toward a robust parametric control toolbox. In: Proceedings of SICE Annual Conference 2003 (Fukui, Japan). (2003) 1716–1721

[6] Weispfenning, V.: Quantifier elimination for real algebra—the quadratic case and beyond. Applicable Algebra in Engineering Communication and Computing **8** (1997) 85–101

[7] Loos, R., Weispfenning, V.: Applying linear quantifier elimination. The Computer Journal **36** (1993) 450–462 Special issue on computational quantifier elimination.

[8] Collins, G.E.: Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In Caviness, B., Johnson, J., eds.: Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts and Monographs in Symbolic Computation. Springer, Wien, New York (1998) 85–121

[9] Hong, H.: An improvement of the projection operator in cylindrical algebraic decomposition. In Caviness, B., Johnson, J., eds.: Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts and Monographs in Symbolic Computation. Springer, Wien, New York (1998) 166–173

[10] McCallum, S.: An improved projection operation for cylindrical algebraic decomposition. In Caviness, B., Johnson, J., eds.: Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts and Monographs in Symbolic Computation. Springer, Wien, New York (1998) 242–268

[11] Brown, C.W.: Improved projection for cylindrical algebraic decomposition. Journal of Symbolic Computation **32** (2001) 447–465

[12] Shorten, R.N., Narendra, K.S.: Necessary and sufficient conditions for the existence of a common quadratic Lyapunov function for m stable second order linear time-invariant systems. Proc. of ACC (2000) 359–363

[13] Shorten, R.N., Narendra, K.S.: Necessary and sufficient conditions for the existence of a common quadratic Lyapunov function for two stable second order linear time-invariant systems. Proc. of ACC (1999) 1410–1414

[14] Mori, Y.: Investigation on common Lyapunov function: Toward complete analysis. Kyoto Institute of Technology, Doctoral Thesis, (in Japanese) (2002)

[15] Nguyen, T.V.: Common Lyapunov function problem: Quadratic and infinity-norm functions. Kyoto Institute of Technology, Master Thesis (2003)