Doing Algebraic Geometry with the RegularChains Library

Parisa Alvandi[1], Changbo Chen[2], Steffen Marcus[3], Marc Moreno Maza[1], Éric Schost[1], and Paul Vrbik[1]

[1] University of Western Ontario, Canada
{palvandi,moreno,eschost,pvrbik}@csd.uwo.ca
[2] Chongqing Institute of Green and Intelligent Technology, CAS, China
changbo.chen@hotmail.com
[3] The College of New Jersey, Ewing, USA
steffenmarcus@gmail.com

Abstract. Traditionally, Groebner bases and cylindrical algebraic decomposition are the fundamental tools of computational algebraic geometry. Recent progress in the theory of regular chains has exhibited efficient algorithms for doing local analysis on algebraic varieties. In this note, we present the implementation of these new ideas within the module `AlgebraicGeometryTools` of the `RegularChains` library. The functionalities of this new module include the computation of the (non-trivial) limit points of the quasi-component of a regular chain. This type of calculation has several applications like computing the Zarisky closure of a constructible set as well as computing tangent cones of space curves, thus providing an alternative to the standard approaches based on Groebner bases and standard bases, respectively. From there, we have derived an algorithm which, under genericity assumptions, computes the intersection multiplicity of a zero-dimensional variety at any of its points. This algorithm relies only on the manipulations of regular chains.

Keywords: Algebraic geometry, regular chains, local analysis

1 Overview

Today, regular chains are at the core of algorithms computing triangular decomposition of polynomial systems and which are available in several software packages [7, 9, 10]. Moreover, those algorithms provide back-engines for computer algebra system front-end solvers, such as MAPLE's `solve` command.

One of the algorithmic strengths of the theory of regular chains is its *regularity test* procedure. Given a polynomial p and a regular chain R, both in a polynomial ring $\mathbf{k}[X_1, \ldots, X_n]$ over a field $\mathbf{k}$, this procedure computes regular chains $R_1, \ldots, R_e$ such that $R_1, \ldots, R_e$ is a decomposition of R in some technical sense [4] and for each $1 \leq i \leq e$ the polynomial p is either null or regular

[4] The radical of the saturated ideal $\mathrm{sat}(R)$ of R is equal to the intersection of the radicals of the saturated ideals of $R_1, \ldots, R_e$.

modulo the saturated ideal of R_i. In algebraic terms, this procedure decides whether the hypersurface $V(p)$ contains at least one irreducible component of the variety $V(\mathrm{sat}(R))$. Thanks to the D5 Principle [4], this regularity test avoids factorization into irreducible polynomials and involves only polynomial GCD computations. This is a core routine in most operations on regular chains.

One of the technical difficulties of regular chain theory, however, is the fact that regular chains do not fit well in the "usual algebraic-geometric dictionary" (Chapter 4, [3]). Indeed, the "good" zero set encoded by a regular chain R is a constructible set $W(R)$, called the *quasi-component* of R, which does not correspond exactly to the "good" ideal encoded by R, namely $\mathrm{sat}(R)$, the *saturated ideal* of R. In fact, the affine variety defined by $\mathrm{sat}(R)$ equals $\overline{W(R)}$, that is, the Zariski closure of $W(R)$. This difficulty probably explains why the use of regular chains in computational algebraic geometry remains limited, despite their nice algorithmic properties such as the above mentioned regularity test.

In [1], three of the co-authors of this note have recently proposed a procedure for computing the *non-trivial limit points* of the quasi-component $W(R)$, that is, the set $\lim(W(R)) := \overline{W(R)} \setminus W(R)$ as a finite union of quasi-components of some other regular chains. This procedure, currently implemented in the case where $\mathrm{sat}(R)$ has dimension one, relies only on operations on regular chains, like the regularity test. As a byproduct, it becomes possible to compute $\overline{W(R)}$ without any Gröbner basis computations. We illustrate this feature in Section 2.

The regularity test for regular chains is a powerful tool which has been studied and applied within many situations including polynomial algebra [6], differential algebra [2, 5], and computing with algebraic numbers [4], etc. Broadly speaking, it allows to extend an algorithm working over a field into an algorithm working over a direct product of fields. Or, to phrase it in another way, it allows to extend an algorithm working at point into an algorithm working at a group of points.

Following that strategy, three of the co-authors of this note have proposed, in another recent work [8], an extension of Fulton's algorithm for computing the intersection multiplicity of two plane curves at any of their intersection points. Indeed, as pointed out by Fulton in his *Intersection Theory*, the intersection multiplicity of two plane curves $V(f)$ and $V(g)$ satisfy a series of seven properties which *uniquely* define $I(p; f, g)$ at each point $p \in V(f, g)$. Moreover, the proof of this remarkable fact is constructive, which leads to an algorithm, that we call Fulton's algorithm. Unfortunately, this algorithm implicitly assumes that the coordinates of the point p are rational numbers.

Another limitation of Fulton's algorithm is that it does not generalize to n polynomials $f_1, \ldots, f_n$ in n variables $x_1, \ldots, x_n$. In [8], two extensions of Fulton's algorithm are proposed. First, thanks to the regularity test for regular chains, the construction is adapted such that it can work correctly at any point of $V(f, g)$, rational or not. Secondly, thanks to the above mentioned procedure for computing the limit points of a quasi-component, an algorithmic criterion is proposed for reducing the case of n variables to the bivariate one. These algorithmic tools are now implemented in the module `AlgebraicGeometryTools` and are illustrated in Sections 3 and 4.

2 Computation of limit points

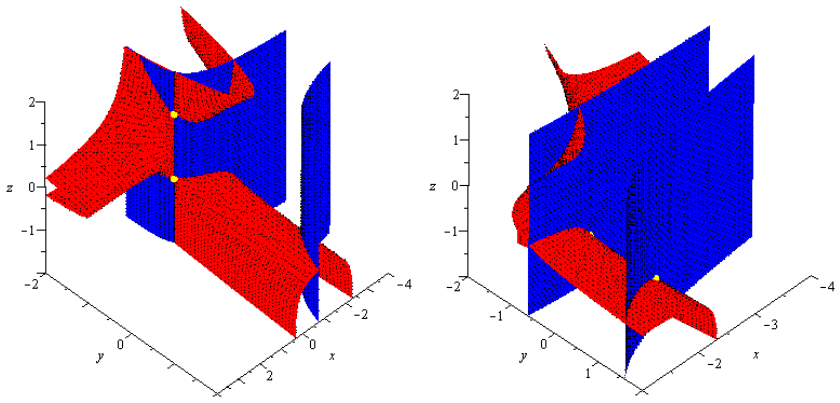

Fig. 1. Limit points of one-dimensional regular chain.

```
> with(AlgebraicGeometryTools):
> R := PolynomialRing([x, y, t]);
> F := [t*y^2 + y + 1, (t + 2)*t*x^2 + (y +1)* (x + 1)];
> C := Chain(F, Empty(R), R);
> lm := LimitPoints(C, R, false, true);
> Display(lm, R);
```

$$R := polynomial\_ring$$

$$F := \left[t\,y^2 + y + 1,\ (t+2)\,t\,x^2 + (y+1)\,(x+1)\right]$$

$$C := regular\_chain$$

$$lm := [regular\_chain,\ regular\_chain,\ regular\_chain,\ regular\_chain]$$

$$\left[\left\{\begin{array}{c} x+1=0 \\ y+\frac{1}{2}=0 \\ t+2=0 \end{array}\right\},\ \left\{\begin{array}{c} x+1=0 \\ y-1=0 \\ t+2=0 \end{array}\right\},\ \left\{\begin{array}{c} x+\frac{1}{2}=0 \\ y+1=0 \\ t=0 \end{array}\right\},\ \left\{\begin{array}{c} x-1=0 \\ y+1=0 \\ t=0 \end{array}\right\}\right]$$

Fig. 2. Limit point computation with `AlgebraicGeometryTools`.

For a regular chain [5] $R \subset \mathbf{k}[X_1, \ldots, X_n]$, recall that the *quasi-component* of R is $W(R) := V(R) \setminus V(h_R)$, that is, the common zeros of R that do not cancel the product h_R of the initials of R. As mentioned in Section 1, computing the non-trivial limit points, that is, the set $\lim(W(R)) := \overline{W(R)} \setminus W(R)$ has many applications. The algorithm proposed in [1] relies on the Puiseux series solutions of the quasi-component $W(R)$.

[5] We refer to [1] for the basic concepts and properties of regular chain theory.

The MAPLE session displayed on Figure 2 illustrates a limit point computation with $R = \{ty^2 + y + 1, (t+2)tx^2 + (y+1)(x+1)\}$ for the variable ordering $t < y < x$. Thus we have $h_R = t(t-2)$. As shown in Figure 1, there are four limit points - see the yellow dots - which are returned by the command `LimitPoints` in the form of regular chains. Other formats are possible; they are controlled by the last two arguments of the `LimitPoints` command.

3 Intersection multiplicity in the plane

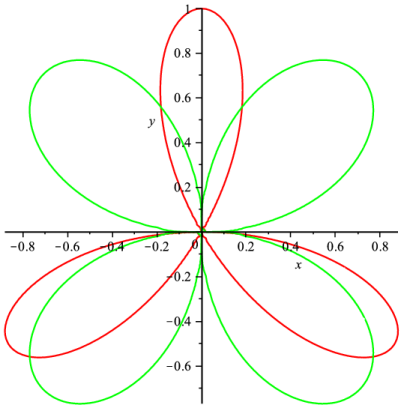

Fig. 3. Two plane curves with an intersection multiplicity of 14 at the origin.

```
> R := PolynomialRing([x,y]);
> F := [(x^2+y^2)^2+3*x^2 * y-y^3, (x^2+y^2)^3-4 *x^2 *y^2];
> dec := TriangularizeWithMultiplicity(F, R);
> Display(dec,R);
>
```

$$R := polynomial\_ring$$

$$F := \left[\left(x^2 + y^2\right)^2 + 3\,x^2\,y - y^3,\ \left(x^2 + y^2\right)^3 - 4\,x^2\,y^2 \right]$$

$$dec := [[1, regular\_chain], [14, regular\_chain]]$$

$$\left[\left[1, \left\{ \begin{array}{c} (64\,y + 80)\,x^2 + 20\,y - 15 = 0 \\ 16\,y^2 - 5 = 0 \\ 64\,y + 80 \neq 0 \end{array} \right\} \right], \left[14, \left\{ \begin{array}{c} x = 0 \\ y = 0 \end{array} \right\} \right] \right]$$

Fig. 4. Intersection multiplicity computation with `AlgebraicGeometryTools`.

Let us now turn to intersection multiplicity computation. We start with the bivariate case, illustrating it with the two plane curves depicted on Figure 3. The command `TriangularizeWithMultiplicity`, see Figure 4, computes the five intersection points of these curves and, for each of them, returns the corresponding intersection multiplicity. Observe that four of these five points are

described by a single regular chain; moreover, at each of those, the intersection multiplicity is 1. In fact, the computation of their intersection multiplicity is performed as a single computation since there is no need to write explicitly the coordinates of each of these points.

4 Intersection multiplicity in higher dimension and computation of tangent cones

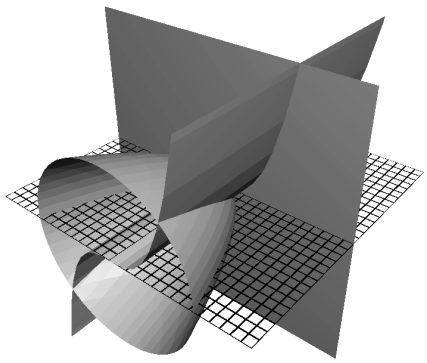

Fig. 5. Three surfaces: case where reduction 3D to 2D is straightforward.

```
> R   := PolynomialRing( [x,y,z] ):
> F  := [x, x-y^2-z^2,y-z^3];
> dec := Triangularize(F, R); Display(dec, R);
>
```

$$F := [x,\ -y^2 - z^2 + x,\ -z^3 + y]$$

$$dec := [\mathit{regular\_chain},\ \mathit{regular\_chain}]$$

$$\left[\left\{\begin{array}{c} x = 0 \\ y - z^3 = 0 \\ z^4 + 1 = 0 \end{array}\right\},\ \left\{\begin{array}{c} x = 0 \\ y = 0 \\ z = 0 \end{array}\right\}\right]$$

```
> IsTransverse(dec[1], F[3], F[1..2], R); IsTransverse(dec[2], F[3], F[1..2], R);
```

$$\mathit{true}$$

$$\mathit{true}$$

```
> dec := TriangularizeWithMultiplicity(F, R); Display(dec, R);
```

$$dec := [[1,\ \mathit{regular\_chain}],\ [2,\ \mathit{regular\_chain}]]$$

$$\left[\left[1,\ \left\{\begin{array}{c} x = 0 \\ y - z^3 = 0 \\ z^4 + 1 = 0 \end{array}\right\}\right],\ \left[2,\ \left\{\begin{array}{c} x = 0 \\ y = 0 \\ z = 0 \end{array}\right\}\right]\right]$$

Fig. 6. Intersection multiplicity computation with `AlgebraicGeometryTools`.

Fig. 7. Three surfaces: case where reduction 3D to 2D requires cylindrification.

```
> R   := PolynomialRing( [z, y, x] ):
> F := [x^2+y+z-1, x+y^2+z-1,z+y+z^2-1];
> dec := Triangularize(F, R); Display(dec, R);
>
```

$$F := \left[x^2+y+z-1,\ y^2+x+z-1,\ z^2+y+z-1\right]$$

$$dec := [\mathit{regular\_chain}, \mathit{regular\_chain}, \mathit{regular\_chain}, \mathit{regular\_chain}, \mathit{regular\_chain}]$$

$$\left[\left\{\begin{array}{l} z-x=0 \\ y-x=0 \\ x^2+2x-1=0 \end{array}\right\}, \left\{\begin{array}{l} z+2=0 \\ y+1=0 \\ x-2=0 \end{array}\right\}, \left\{\begin{array}{l} z=0 \\ y-1=0 \\ x=0 \end{array}\right\}, \left\{\begin{array}{l} z-1=0 \\ y+1=0 \\ x+1=0 \end{array}\right\}, \left\{\begin{array}{l} z+1=0 \\ y-1=0 \\ x-1=0 \end{array}\right\}\right]$$

```
> Display(dec[3], R); IsTransverse(dec[3], F[3], F[1..2], R);
```

$$\left\{\begin{array}{l} z=0 \\ y-1=0 \\ x=0 \end{array}\right\}$$

$$\mathit{false}$$

```
> Cylindrify(dec[3], F, R, true);
```

$$\left[x^2+z+y-1,\ -x^2+y^2+x-y,\ x^4+2x^2y-3x^2+y^2-2y+1\right]$$

Fig. 8. Intersection multiplicity computation with `AlgebraicGeometryTools`.

Fulton's algorithm does not apply directly to n polynomials $f_1, \ldots, f_n$ in n variables $x_1, \ldots, x_n$. However, the criterion proved in [8] allows to reduce the computation of the intersection multiplicity of $f_1, \ldots, f_n$ at a point p to an intersection multiplicity calculation in a lower dimension space. We recall this criterion. Assume that $h_n = V(f_n)$ is non-singular at p. Let v_n be tangent hyperplane of h_n at p. Assume that h_n meets each component (through p) of the curve $\mathcal{C} = V(f_1, \ldots, f_{n-1})$ transversally (that is, the tangent cone $TC_p(\mathcal{C})$ intersects v_n only at the point p). Let $g \in k[x_1, \ldots, x_n]$ be the degree 1 polynomial defining v_n. Then, we have

$$I(p; f_1, \ldots, f_n) = I(p; f_1, \ldots, f_{n-1}, g).$$

Assume[6] that the coefficient of x_n in g is non-zero, thus $g = x_n - g'$, where $g' \in k[x_1, \ldots, x_{n-1}]$. Hence, we can rewrite the ideal $\langle f_1, \ldots, f_{n-1}, g\rangle$ as $\langle g_1, \ldots, g_{n-1}, g'\rangle$ where g_i is obtained from f_i by substituting x_n with g'. Then, we have

$$I_n(p; f_1, \ldots, f_n) = I_{n-1}(p|_{x_1,\ldots,x_{n-1}}; g_1, \ldots, g_{n-1}).$$

[6] One can always reduce to this case by means of a linear change of coordinates.

In the example from Figure 5, the tangent hyperplane $y = 0$ of $V(y-z^3)$ at the origin and each component (through the origin) of the curve $\mathcal{C} := V(x, x+y^2-z^2) = V(x, (y-z)(y+z))$ meet transversally. Therefore, we have $I_3((0,0,0); x, x+y^2-z^2, y-z^3) = I_2((0,0); x, x-z^2) = 2$, as shown by the calculation on Figure 6.

Verifying this transversality condition requires the computation of tangent cones of and tangent planes. The module `AlgebraicGeometryTools` provides commands `TangentCone` (of space curves) and `TangentPlane` for that purpose. Each tangent cone is computed as a limit of secants and reduces to compute limit points of quasi-components of one-dimensional regular chains.

In Figure 8, the transversality condition does not hold. This is detected using the `TangentCone` and `TangentPlane` commands. Another strategy is then attempted, we call it *cylindrification*. We explain its principle under simplifying assumptions. Assume that among $f_1, \ldots, f_n$ one polynomial, say f_n has degree one in x_n and assume that its coefficient in x_n is invertible in the local ring at p. Then, one replaces $f_1, \ldots, f_{n-1}$ by $g_1, \ldots, g_{n-1}$ where g_i is the pseudo-remainder of f_i by f_n w.r.t x_n. It is not hard to see that $I(p; f_1, \ldots, f_n) = I(p; g_1, \ldots, g_{n-1}, f_n)$ holds. Moreover, the transversality condition clearly holds for $g_1, \ldots, g_{n-1}, f_n$. Therefore, we are back in the above case where we could reduce computations from n to $n-1$ variables. Returning to the example of Figure 8, cylindrification replaces the three surfaces on the left of Figure 7 with the surfaces on the right.

5 Concluding remarks

The new module `AlgebraicGeometryTools` of the `RegularChains` library performs various operations on one-dimensional objects (computation of limit points of constructible sets, tangent cones of space curves) as well as intersection multiplicity computation for zero-dimensional varieties. All these operations rely only on regular chain techniques, that is, no calculations of Gröbner bases or standard bases are performed.

Extending limit point computations to higher dimension is work in progress. The cylindrification strategy (for reducing intersection multiplicity calculation from n to $n-1$ variables) may fail in some situations; improving this situation is also work in progress.

Benchmarks reported in the PhD dissertation of the last author shows that the command `TriangularizeWithMultiplicity` is computationally efficient in the sense that it can process almost all examples which can be processed by the `Triangularize`[7]

The `RegularChains` library is available at `www.regularchains.org`.

Acknowledgments

This work was supported by the NSFC (11301524) and the CSTC (cstc2013jjys0002).

[7] The `Triangularize` command decomposes a polynomial system into regular chains but discards any multiplicity information.

References

1. P. Alvandi, C. Chen, and M. Moreno Maza. Computing the limit points of the quasi-component of a regular chain in dimension one. In Vladimir P. Gerdt, Wolfram Koepf, Ernst W. Mayr, and Evgenii V. Vorozhtsov, editors, *CASC*, volume 8136 of *Lecture Notes in Computer Science*, pages 30–45. Springer, 2013.
2. F. Boulier, D. Lazard, F. Ollivier, and M. Petitot. Computing representations for radicals of finitely generated differential ideals. *Appl. Algebra Eng. Commun. Comput.*, 20(1):73–121, 2009.
3. D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Spinger-Verlag, 2nd edition, 1996.
4. J. Della Dora, C. Dicrescenzo, and D. Duval. About a new method for computing in algebraic number fields. In *Proc. of EUROCAL' 85*, pages 289–290, 1985.
5. E. Hubert. Factorization-free decomposition algorithms in differential algebra. *J. Symb. Comput.*, 29(4-5):641–662, 2000.
6. M. Kalkbrener. Algorithmic properties of polynomial rings. *J. Symb. Comput.*, 26(5):525–581, 1998.
7. F. Lemaire, M. Moreno Maza, and Y. Xie. The `RegularChains` library. In *Maple 10*, Maplesoft, Canada, 2005. `www.regularchains.org`
8. S. Marcus, M. Moreno Maza, and P. Vrbik. On Fulton's algorithm for computing intersection multiplicities. In *Proc. of CASC'12*, pages 198–211, 2012.
9. D. K. Wang. The `Wsolve` package. http://www.mmrc.iss.ac.cn/∼dwang/wsolve.html.
10. D. M. Wang. Epsilon 0.618. http://www-calfor.lip6.fr/∼wang/epsilon.