Towards a Generalization of Fulton's Intersection Multiplicity Algorithm

Marc Moreno Maza and Ryan Sandford

Introduction

Take $f_1, \ldots, f_n \in \mathbf{k}[x_1, \ldots, x_n]$ such that $\mathbf{V}(f_1, \ldots, f_n)$ is zero-dimensional and fix some $p \in \mathbf{V}(f_1, \ldots, f_n)$. The intersection multiplicity $\mathrm{Im}(p;\, f_1, \ldots, f_n)$ gives the weight at the point p of the weighted sum in Bézout's Theorem.

Magma supports the computation of intersection multiplicities for two projective curves while Singular provides support for the n-variate case, but only at the origin. In 2020, Maple introduced support for the computation of intersection multiplicities in the n-variate case at any point. By applying an algorithmic criterion, the Maple implementation seeks to reduce to the bivariate case, a case which has a well-known solution given by Fulton's algorithm. Unfortunately, this reduction is not always possible.

Following the design goals of Maple's intersection multiplicity algorithm, we seek to design an algorithm which can provide an alternative to intersection multiplicity algorithms which use standard bases (a Gröbner basis with a local term ordering). That is, we wish to compute intersection multiplicities in the n-variate case, without computing a standard basis of $f_1, \ldots, f_n$. Further, we aim to design an algorithm that can in practice, compute intersection multiplicities at any point, rational or not.

Rather than reducing to the bivariate case to apply Fulton's algorithm, we extend Fulton's algorithm to a partial intersection multiplicity algorithm in the n-variate case. We further extend our generalization of Fulton's algorithm to handle any point as input, rational or not, by encoding such points in a zero-dimensional regular chain.

Algorithm

Definition 1 (Local Ring)**.** Take $p \in \mathbb{A}^n$, we define the local ring at p as

$$\mathcal{O}_{\mathbb{A}^n,p} := \left\{ \tfrac{f}{g} \mid f, g \in \mathbf{k}[x_1, \ldots, x_n] \text{ where } g(p) \neq 0 \right\}.$$

Definition 2 (Intersection Multiplicity)**.** Let $f_1, \ldots, f_n \in \mathbf{k}[x_1, \ldots, x_n]$. We define the intersection multiplicity of $f_1, \ldots, f_n$ at p as the dimension of the local ring at p modulo the ideal generated by $f_1, \ldots, f_n$ in the local ring at p, as a vector space over $\mathbf{k}$. That is,

$$\mathrm{Im}(p;\, f_1, \ldots, f_n) := \dim_{\mathbf{k}}(\mathcal{O}_{\mathbb{A}^n,p} \,/\, \langle f_1, \ldots, f_n \rangle).$$

Definition 3 (Modular Degree)**.** Take $p \in \mathbb{A}^n$, $v \in \{x_1, \ldots, x_n\}$, and $f \in \mathbf{k}[x_1, \ldots, x_n]$ where $x_1 > \ldots > x_n$. We define the modular degree of f at p with respect to v as

$$\deg_v(f \bmod \langle V_{<v,p} \rangle),$$

where $V_{<v,p} = \{x_i - p_i | x_i < v\}$. If $V_{<v,p} = \varnothing$ then the modular degree of f at p with respect to v is simply the degree of f with respect to v. If p is the origin, we denote by $\mathrm{moddeg}(f, v)$ the modular degree of f at v.

Theorem 1 (Generalization of Fulton's Properties)**.** Let $p = (p_1, \ldots, p_n) \in \mathbb{A}^n$ and $f_1, \ldots, f_n \in \mathbf{k}[x_1, \ldots, x_n]$.

(n-1) $\mathrm{Im}(p;\, f_1, \ldots, f_n)$ is a non-negative integer iff $\mathbf{V}(f_1, \ldots, f_n)$ is zero-dimensional.

(n-2) $\mathrm{Im}(p;\, f_1, \ldots, f_n) = 0$ iff $p \notin \mathbf{V}(f_1, \ldots, f_n)$.

(n-3) $\mathrm{Im}(p;\, f_1, \ldots, f_n)$ is invariant under affine changes of coordinates on $\mathbb{A}^n$.

(n-4) $\mathrm{Im}(p;\, f_1, \ldots, f_n) = \mathrm{Im}(p;\, f_{\sigma(1)}, \ldots, f_{\sigma(n)})$.

(n-5) $\mathrm{Im}(p;\, (x_1 - p_1)^{m_1}, \ldots, (x_n - p_n)^{m_n}) = m_1 \cdot \ldots \cdot m_n$ for $m_1, \ldots, m_n \in \mathbb{N}$.

(n-6) $\mathrm{Im}(p;\, f_1, \ldots, gh) = \mathrm{Im}(p;\, f_1, \ldots, g) + \mathrm{Im}(p;\, f_1, \ldots, h)$ for any $g, h \in \mathbf{k}[x_1, \ldots, x_n]$ such that $f_1, \ldots, gh$ is a regular sequence in $\mathcal{O}_{\mathbb{A}^n,p}$.

(n-7) $\mathrm{Im}(p;\, f_1, \ldots, f_n) = \mathrm{Im}(p;\, f_1, \ldots, f_n + g)$ for any $g \in \langle f_1, \ldots, f_{n-1} \rangle$.

Algorithm 1: Generalized Fulton's Algorithm

Function $\mathrm{im_n}(p; f_1, \ldots, f_n)$
Input: Let: $x_1 > \ldots > x_n$.

1. $p \in \mathbb{A}^n$ the origin.
2. $f_1, \ldots, f_n \in \mathbb{K}[x_1, \ldots, x_n]$ such that $f_1, \ldots, f_n$ form a regular sequence in $\mathcal{O}_{\mathbb{A}^n,p}$ or at least one such f_i is a unit in $\mathcal{O}_{\mathbb{A}^n,p}$.

Output: $\mathrm{Im}(p;\, f_1, \ldots, f_n)$ or Fail

if $f_i(p) \neq 0$ for any $i = 1, \ldots, n$ **then**
  return 0
if $n = 1$ **then** /* Compute multiplicity */
  return $\max(m \in \mathbb{Z}^+ \mid f_n \equiv 0 \bmod \langle x_1^m \rangle)$
for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, n-1$ **do**
    $r_j^{(i)} \leftarrow \mathrm{moddeg}(f_i, x_j)$
for $j = 1, \ldots, n-1$ **do**
  Reorder $f_1, \ldots, f_{n-j+1}$ so that $r_j^{(1)} \leq \ldots \leq r_j^{(n-j+1)}$
  $m \leftarrow \min(i \mid r_j^{(i)} > 0)$ or $m \leftarrow \infty$ if no such i exists
  if $m \leq (n-j)$ **then**
    for $i = m+1, \ldots, n-j+1$ **do**
      $d \leftarrow r_j^{(i)} - r_j^{(m)}$
      $L_m \leftarrow \mathrm{lc}(f_m(x_1, \ldots, x_j, 0, \ldots, 0); x_j)$
      $L_i \leftarrow \mathrm{lc}(f_i(x_1, \ldots, x_j, 0, \ldots, 0); x_j)$
      if $L_m(p) \neq 0$ **then**
        $f_i' \leftarrow L_m f_i - x_j^d L_i f_m$
      else if $L_m \mid L_i$ **then**
        $f_i' \leftarrow f_i - x_j^d \frac{L_i}{L_m} f_m$
      else
        return Fail
    return $\mathrm{im_n}(p; f_1, \ldots, f_m, f'_{m+1}, \ldots, f'_{n-j+1}, \ldots, f_n)$
for $i = 1, \ldots, n-1$ **do**
  $q_i \leftarrow$ quo $(f_i(x_1, \ldots, x_{n-i+1}, 0, \ldots, 0), x_{n-i+1}; x_{n-i+1})$
return
$\mathrm{im_n}(p; q_1, f_2, \ldots, f_n)$
$+\, \mathrm{im_{n-1}}(p; q_2(x_1, \ldots, x_{n-1}, 0), \ldots, f_n(x_1, \ldots, x_{n-1}, 0))$
$+$
$\vdots$
$+\mathrm{im_2}(p; q_{n-1}(x_1, x_2, 0, \ldots, 0), f_n(x_1, x_2, 0, \ldots, 0))$
$+\mathrm{im_1}(p; f_n(x_1, 0, \ldots, 0))$

The generalization of Fulton's algorithm is best understood by examining the matrix of modular degrees corresponding to $f_1, \ldots, f_n$. The main loop of the algorithm terminates when all entries above the anti-diagonal are equal to $-\infty$. In fact, this condition is exactly what we must enforce in order to split the intersection multiplicity computation into a sum of smaller intersection multiplicity computations and make progress towards termination.

Definition 4 (Matrix of Modular Degrees)**.** The matrix of modular degrees of $f_1, \ldots, f_n \in \mathbf{k}[x_1, \ldots, x_n]$ is the matrix whose i-th, j-th entry is $\mathrm{moddeg}(f_i, x_j)$.

Lemma 1. Let $f_1, \ldots, f_n \in \mathbf{k}[x_1, \ldots, x_n]$ forming a regular sequence in $\mathcal{O}_{\mathbb{A}^n,p}$ where p is the origin. Let $V = \{x_1, \ldots, x_n\}$ and let $V_{>v} = \{x_i \in V \mid x_i > v\}$. Define $J : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that $J(i) = n - i + 1$. Assume $\mathrm{moddeg}(f_i, v) < 0$ holds for all $i = 1, \ldots, n-1$ and all $v \in V_{>x_{J(i)}}$. Then, the variable $x_{J(i)}$ divides $f_i(x_1, \ldots, x_{J(i)}, 0, \ldots, 0)$. Moreover, if q_i denotes the quotient of $f_i(x_1, \ldots, x_{J(i)}, 0, \ldots, 0)$ by $x_{J(i)}$ then we have:

$$\begin{aligned}\mathrm{Im}(p;\, f_1, \ldots, f_n) &= \mathrm{Im}(p;\, q_1, f_2, \ldots, f_n) + \mathrm{Im}(p;\, x_n, q_2, \ldots, f_n) \\ &\quad + \ldots + \mathrm{Im}(p;\, x_n, \ldots, x_{J(i)+1}, q_i, f_{i+1}, \ldots, f_n) + \ldots \\ &\quad + \mathrm{Im}(p;\, x_n, x_{n-1}, \ldots, q_{n-1}, f_n) + m_n\end{aligned}$$

where $m_n = max(m \in \mathbb{Z}^+ \mid f_n(x_1, 0, \ldots, 0) \equiv 0 \bmod \langle x_1^m \rangle)$.

In order to apply the lemma, we must first rewrite $f_1, \ldots, f_n$ so that the matrix of modular degrees has all entries above the anti-diagonal equal to $-\infty$. We iterate column-wise, and hence the j-th iteration corresponds to the variable x_j. In the j-th iteration we choose a pivot element f_m, with minimal modular degree in x_j, and use f_m to reduce the modular degree in x_j of the other polynomials using only operations permissible by (n-7).

The algorithm is partial, meaning it doesn't always succeed. This is because (n-7) cannot generically be used to reduce modular degrees when $n > 2$. When (n-7) is applicable, the algorithm proceeds as expected. When (n-7) is not applicable, we return Fail to the user. In particular, suppose we wish to replace f_i with

$$f_i' := \mathrm{lc}(f_m(x_1, \ldots, x_k, 0, \ldots, 0); x_k) f_i - x_k^d \mathrm{lc}(f_i(x_1, \ldots, x_k, 0, \ldots, 0); x_k) f_m,$$

for some $i, m, k, d \in \mathbb{N}, i \neq m$. Unlike the bivariate case, $\mathrm{lc}(f_m(x_1, \ldots, x_k, 0, \ldots, 0); x_k)$ is not always invertible in $\mathcal{O}_{\mathbb{A}^n,p}$, hence property ($n$-7) does not always apply. Hence, it is not generically true that $\langle f_1, \ldots, f_i, \ldots, f_n \rangle = \langle f_1, \ldots, f_i', \ldots, f_n \rangle$. That is, substituting f_i' for f_i does not necessarily preserve intersection multiplicity and thus, return Fail.

Example 1. Let $f_1, f_2, f_3 \in \mathbf{k}[x, y, z]$ be given by $f_1 = x^2, f_2 = (x+1)y + x^3, f_3 = y^2 + z + x^3$. The algorithm first computes the below matrix r of modular degrees:

$$r = \begin{bmatrix} 2 & 0 \\ 3 & 1 \\ 3 & 2 \end{bmatrix},$$

where the i-th row corresponds to the polynomial f_i and the j-th column corresponds to the variable x_j. Hence, (i, j)-th entry is the modular degree of f_i w.r.t. x_j. Write:

$$f_2' := f_2 - x f_1 = (x+1)y + x^3 - x^3 = (x+1)y, \quad \text{and} \quad f_3' := f_3 - x f_1 = y^2 + z + x^3 - x^3 = y^2 + z.$$

Redefine $f_2 := f_2'$ and $f_3 := f_3'$. Hence, we consider $f_1 = x^2, f_2 = (x+1)y, f_3 = y^2 + z$. The matrix r computed in the first section is now:

$$r = \begin{bmatrix} 2 & 0 \\ -\infty & 1 \\ -\infty & 2 \end{bmatrix},$$

and after reordering f_1, f_2, f_3 by modular degree we have $f_1 = (x+1)y, f_2 = y^2 + z, f_3 = x^2$, with matrix of modular degrees:

$$r = \begin{bmatrix} -\infty & 1 \\ -\infty & 2 \\ 2 & 0 \end{bmatrix}.$$

Consider $f_1 = (x+1)y, f_2 = y^2 + z, f_3 = x^2$. Write

$$f_2' := (x+1)f_2 - y f_1 = (x+1)y^2 + (x+1)z - (x+1)y^2 = (x+1)z.$$

Redefining $f_2 := f_2'$ and reordering by modular degree in y, we have $f_1 = (x+1)z, f_2 = (x+1)y, f_3 = x^2$, and the matrix of modular degrees is now:

$$r = \begin{bmatrix} -\infty & -\infty \\ -\infty & 1 \\ 2 & 0 \end{bmatrix}.$$

Applying the Lemma on $f_1 = (x+1)z, f_2 = (x+1)y, f_3 = x^2$ gives:

$$\begin{aligned}\mathrm{Im}(p;\, f_1, f_2, f_3) &= \mathrm{Im}(p;\, x+1, f_2, f_3) + \mathrm{Im}(p;\, z, x+1, f_3) + \mathrm{Im}(p;\, z, y, f_3) \\ &= 0 + 0 + 2.\end{aligned}$$

Implementation and Experimentation

We extend our algorithm to handle a zero-dimensional regular chain as input, rather than just a point, allowing it to compute intersection multiplicities at any point, rational or not. Additionally, we reimplement Maple's IntersectionMultiplicity command to combine the generalization of Fulton's algorithm with the partial intersection multiplicity algorithm already in Maple, forming a hybrid intersection multiplicity algorithm. Lastly, we modify the TriangularizeWithMultiplicity command to support this new implementation; a command which first solves the system of polynomial equations and then maps the IntersectionMultiplicity command to each solution.

The first (respectively second) table below compares the generalization of Fulton's algorithm to the existing intersection multiplicity algorithm in Maple using the IntersectionMultiplicity command (respectively TriangularizeWithMultiplicity). For the TriangularizeWithMultiplicity command, Success Ratio denotes the number of intersection multiplicities successfully computed over the total number of regular chains returned. All tests are un in serial using Maple 2021.2.

	System Specifications			Fulton		Maple	
System	n	Ordering	Point	Im	CPU Time	Im	CPU Time
cbms1	3	$x > y > z$	$(0,0,0)$	FAIL	32ms	ERROR	16ms
cbms2	3	$x > y > z$	$(0,0,0)$	FAIL	31ms	ERROR	31ms
mth191	3	$x > y > z$	$(0,1,0)$	4	31ms	ERROR	1.47s
Ojika2	3	$x > y > z$	$(0,0,1)$	2	63ms	2	1.53s
Ojika2	3	$x > y > z$	$(1,0,0)$	2	62ms	2	1.48s
Ojika3	3	$x > y > z$	$(0,0,1)$	4	31ms	4	1.62s
Ojika3	3	$x > y > z$	$(-\frac{5}{2}, \frac{5}{2}, 1)$	2	31ms	2	1.02s
Ojika4	3	$x > y > z$	$(0,0,10)$	FAIL	16ms	3	2.00s
Ojika4	3	$x > z > y$	$(0,10,0)$	3	63ms	3	4.02s
DZ1	4	$x_1 > \ldots > x_4$	$(0,0,0,0)$	FAIL	31ms	ERROR	16ms
DZ2	3	$x > z > y$	$(0,0,-1)$	16	78ms	ERROR	16ms
Solotarev	4	$x_1 > \ldots > x_4$	$(\frac{5}{3}, -1, 5, -\frac{47}{27})$	2	110ms	2	1.36s

	System Specifications		Fulton		Maple	
System	n	Ordering	Success Ratio	CPU Time	Succeess Ratio	CPU Time
cbms1	3	$x > y > z$	10/11	641ms	ERROR	218ms
cbms2	3	$x > y > z$	1/2	11.55s	ERROR	422ms
mth191	3	$x > y > z$	8/8	609ms	ERROR	2.45s
Ojika2	3	$x > y > z$	4/4	219ms	4/4	5.38s
Ojika3	3	$x > y > z$	2/2	62ms	2/2	5.02s
Ojika4	3	$x > y > z$	3/5	438ms	ERROR	906ms
Ojika4	3	$x > z > y$	5/5	500ms	ERROR	3.41s
Caprasse	4	$x_1 > x_2 > \ldots$	4/15	1.58s	NR	>2000s
Caprasse	4	$x_4 > x_2 > x_1 > x_3$	12/15	4.48s	ERROR	14.09s
DZ1	4	$x_1 > x_2 > \ldots$	NR	>2000s	ERROR	313ms
DZ2	3	$x > z > y$	2/2	172ms	ERROR	47ms
Solotarev	4	$x_1 > x_2 > \ldots$	4/4	453ms	4/4	3.47s