Complexity Bounds for the Generalization of Fulton's Intersection Multiplicity Algorithm

Ryan Sandford

Independent Researcher

Overview

We analyze the Generalization of Fulton's Intersection Multiplicity Algorithm, a symbolic partial algorithm for computing intersection multiplicities via recursive algebraic rewriting. We show the algorithm exhibits non-elementary worst-case complexity on dense systems and exponential complexity in favorable cases. Finally, we propose an optimization that reduces the depth of recursion across branches of computation, limiting the growth responsible for non-elementary complexity.

Introduction and Notation

The Generalization of Fulton's Intersection Multiplicity Algorithm (GFIMA) [1, 2] is a symbolic partial algorithm which computes intersection multiplicities by recursively applying algebraic rewrite rules to systems of polynomial equations. GFIMA, along with its extension based on regular chains [3], are implemented in the REGULAR CHAINS LIBRARY [4] distributed with MAPLE.

Let $\mathbb{K}$ be a field and let $\mathbb{A}^n$ denote the affine space of dimension n over $\mathbb{K}$. Take $f_1, \ldots, f_n \in \mathbb{K}[x_1, \ldots, x_n]$ and fix $p \in \mathbb{A}^n$. The intersection multiplicity of $f_1, \ldots, f_n$ at p, written $\mathrm{IM}(p; f_1, \ldots, f_n)$, is the dimension of $\mathcal{O}_{\mathbb{A}^n,p} / \langle f_1, \ldots, f_n \rangle$ as a $\mathbb{K}$ vector space, where $\mathcal{O}_{\mathbb{A}^n,p}$ is the localization of the polynomial ring at p. We define the degree of the zero polynomial to be $-\infty$ with respect to any variable. Fix an ordering $x_1 \succ \ldots \succ x_n$. We write $f^{(i)} := f(x_1, \ldots, x_i, p_{i+1}, \ldots, p_n)$ to denote the specialization of f at p in the last $n-i$ variables, and $p^{(i)} := (p_1, \ldots, p_i)$ to denote truncation to the first i coordinates. We assume specializations are computed in reverse order and intermediate values are cached unless invalidated by structural changes to f. We define the elimination degree of f at an index i, written $\mathrm{elimdeg}(f, i)$, to be $\deg(f^{(i)}; x_i)$. Let $\Phi^{[k]}$ denote the k-fold composition of the function Φ. The valuation of f at $x_i - p_i$ is defined as the largest m such that $f \equiv 0 \bmod (x_i - p_i)^m$.

Proposition 1

If $f_1, \ldots, f_n \in \mathbb{K}[x_1, \ldots, x_n]$ with maximum degree d, $n > 2$, and $p \in \mathbb{A}^n$, the worst-case time complexity of Algorithm 1 is $O\big(n^2 \left(\Phi^{[d^n]}(d)\right)^n\big)$ where $\Phi : d \mapsto 2^{n-3}3^{(n-2)(d-1)}(4d-1) - \sum_{j=1}^{n-3} 2^{j-1}3^{j(d-1)} - 1$.

Proposition 2

Take $f_1, \ldots, f_n \in \mathbb{K}[x_1, \ldots, x_n]$ and $p \in \mathbb{A}^n$. If for each i, $\mathrm{elimdeg}(f_{n-i+1}, i) > 0$ and $\mathrm{elimdeg}(f_j, i) < 0$ for any $j < n-i+1$, then $(x_i - p_i)^{v_i}$ divides $f_{n-i+1}^{(i)}$ where v_i is its valuation at $x_i - p_i$. If q_{n-i+1} is the corresponding quotient of each division, we have:

$$\mathrm{IM}(p; f_1, \ldots, f_n) = \sum_{i=1}^{n} \left(\left(\prod_{j=i+1}^{n} v_j \right) \cdot \mathrm{IM}(p^{(i)}; q_{n-i+1}^{(i)}, f_{n-i+2}^{(i)}, \ldots, f_n^{(i)}) \right).$$

Algorithm 1 The Generalization of Fulton's Intersection Multiplicity Algorithm

```
Input: Variable order $x_1 \succ \ldots \succ x_n$; point $p = (p_1, \ldots, p_n) \in \mathbb{A}^n$; polynomials $f_1, \ldots, f_n \in \mathbb{K}[x_1, \ldots, x_n]$ such that $\mathbf{V}(f_1, \ldots, f_n)$ is zero-dimensional at $p$.
Output: $\mathrm{IM}(p; f_1, \ldots, f_n)$ or FAIL
function $\mathrm{gfima}(p; f_1, \ldots, f_n)$
    if $f_i(p) \neq 0$ for any $i = 1, \ldots, n$ then
        return 0
    if $n = 1$ then
        return $\max\{m > 0 \mid f_n \equiv 0 \bmod (x_1 - p_1)^m\}$
    for $j = 1, \ldots, n-1$ do                              ▷ Rewrite Loop
        while TRUE do
            for $i = 1, \ldots, n-j+1$ do
                $r_i \leftarrow \mathrm{elimdeg}(f_i, j)$
            Sort $f_1, \ldots, f_{n-j+1}$ so that $r_1 \leq \ldots \leq r_{n-j+1}$
            if $r_{n-j} < 0$ then
                break
            $m \leftarrow \min\{k \mid r_k > 0\}$
            for $i = m+1, \ldots, n-j+1$ do
                $L_m \leftarrow \mathrm{lc}(f_m^{(j)}; x_j)$
                $L_i \leftarrow \mathrm{lc}(f_i^{(j)}; x_j)$
                $L \leftarrow \mathrm{lcm}(L_i, L_m)$
                if $\frac{L}{L_i}(p) = 0$ then
                    return FAIL
                $f_i \leftarrow \frac{L}{L_i} f_i - (x_j - p_j)^{r_i - r_m} \frac{L}{L_m} f_m$
    for $i = 1, \ldots, n$ do                                ▷ Recursive Split
        $q_i \leftarrow \mathrm{quo}(f_i^{(n-i+1)}, x_{n-i+1} - p_{n-i+1}; x_{n-i+1})$
    return $\sum_{k=1}^{n} \mathrm{gfima}\left(p^{(k)}; q_{n-k+1}^{(k)}, f_{n-k+2}^{(k)}, \ldots, f_n^{(k)}\right)$
```

Worst Case Time Complexity

For dense $f_1, \ldots, f_n$ with $n > 2$, the time complexity of the GFIMA is non-elementary in the worst case. Due to the expression swell experienced by the input system during the rewriting process, operations performed against the final system of polynomials dominate the cost of the algorithm.

Let d be the maximum degree of $f_1, \ldots, f_n$. Each iteration of the Rewrite Loop can increase the degree. The first iteration can increase the degree to at most $2d-1$. Subsequent iterations increase the degree by at most $h : \delta \mapsto 3^{d-1}(2\delta + 1) - 1$. Each completion of the Rewrite Loop therefore increases the degree by at most

$$\begin{aligned}\Phi(d) &:= h^{[n-2]}(2d-1) \\ &= 2^{n-3}3^{(n-2)(d-1)}(4d-1) - \sum_{j=1}^{n-3} 2^{j-1}3^{j(d-1)} - 1.\end{aligned}$$

Since the sum of the heights of each branch in the recursion tree with positive multiplicity is at most d^n and each Recursive Split strictly decreases the remaining multiplicity [2], the worst case occurs when $d^n - 1$ recursive calls which return positive multiplicity are made to GFIMA on systems of size n. In this case the degree of the final rewritten system will be at most $\Phi^{[d^n]}(d)$, and the total complexity reduces to the cost of computing $O(n)$ specializations in $n-1$ variables in the final step. **Proposition 1** gives the final cost of **Algorithm 1** in the worst case.

Efficient Splitting Optimization

Because the overall cost exhibits non-elementary growth, any optimization that limits the number of Recursive Split steps can significantly improve average-case performance. **Proposition 2** achieves this by refining the standard decomposition: rather than decreasing the remaining multiplicity by the sum of sub-multiplicities, it decomposes the remaining multiplicity into a sum of sub-multiplicities weighted by products of valuations. This more aggressive reduction decreases the depth of the recursion tree, hence collapsing the nested compositions responsible for non-elementary growth.

Best Case Time Complexity

In the best case, the input system does not require rewriting, reducing the cost of GFIMA to the cost of computing $O(n)$ specializations in $n-1$ variables. This yields a time complexity of $\Omega(n^2 d^n)$ if only a single invocation is required, or $\Omega(n^2 d^{2n})$ if up to d^n Recursive Split steps are still performed.

Proposition 3

Let $f_1, \ldots, f_n \in \mathbb{K}[x_1, \ldots, x_n]$ be polynomials of degree at most d, where $n > 1$ and $p \in \mathbf{V}(f_1, \ldots, f_n)$. Then the best-case time complexity of Algorithm 1 is $\Omega(n^2 d^n)$.

References

[1] W. Fulton, *Algebraic Curves: An Introduction to Algebraic Geometry*, Addison–Wesley, 1989.

[2] M. Moreno Maza and R. Sandford, "Towards Extending Fulton's Algorithm for Computing Intersection Multiplicities Beyond the Bivariate Case," in *CASC 2021*, LNCS 12865, Springer, 2021, pp. 232–251. https://doi.org/10.1007/978-3-030-85165-1_14

[3] R. Sandford, J. Gerhard, and M. Moreno Maza, "Computing Intersection Multiplicities with Regular Chains," *Maple Trans.*, vol. 2, no. 1, 2022. https://doi.org/10.5206/mt.v2i1.14463

[4] F. Lemaire, M. Moreno Maza, and Y. Xie, "The RegularChains Library in Maple," *ACM SIGSAM Bull.*, vol. 39, no. 3, pp. 96–97, 2005. https://doi.org/10.1145/1113439.1113456