July 2015

Algorithms to Compute Characteristic Classes

Martin Helmer
The University of Western Ontario

Supervisor
Eric Schost
The University of Western Ontario

Graduate Program in Applied Mathematics

A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

© Martin Helmer 2015

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Algebra Commons, Algebraic Geometry Commons, Geometry and Topology Commons, Other Applied Mathematics Commons, and the Theory and Algorithms Commons

Recommended Citation

https://ir.lib.uwo.ca/etd/2923

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact tadam@uwo.ca.
ALGORITHMS TO COMPUTE CHARACTERISTIC CLASSES

(Thesis Format: Integrated-Article)

by

Martin Helmer

Graduate Program in Applied Mathematics

A thesis submitted
in partial fulfilment of the requirements for
a Doctor of Philosophy

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

©Martin Helmer 2015
Abstract

In this thesis we develop several new algorithms to compute characteristic classes in a variety of settings. In addition to algorithms for the computation of the Euler characteristic, a classical topological invariant, we also give algorithms to compute the Segre class and Chern-Schwartz-MacPherson ($c_{SM}$) class. These invariants can in turn be used to compute other common invariants such as the Chern-Fulton class (or the Chern class in smooth cases).

We begin with subschemes of a projective space $\mathbb{P}^n$ over an algebraically closed field of characteristic zero. In this setting we give effective algorithms to compute the $c_{SM}$ class, Segre class and the Euler characteristic. The algorithms can be implemented using either symbolic or numeric methods. The algorithms are based on a new method for calculating the projective degrees of a rational map defined by a homogeneous ideal. Running time bounds are given for these algorithms and the algorithms are found to perform favourably compared to other applicable algorithms. Relations between our algorithms and other existing algorithms are explored. In the special case of a complete intersection scheme we develop a second algorithm to compute $c_{SM}$ classes and Euler characteristics in a more direct and efficient manner.

Each of these algorithms are generalized to subschemes of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$. Running time bounds for the generalized algorithms to compute the $c_{SM}$ class, Segre class and the Euler characteristic are given. Our Segre class algorithm is tested in comparison to another applicable algorithm and is found to perform favourably. To the best of our knowledge there are no other algorithms in the literature which compute the $c_{SM}$ class and Euler characteristic in the multi-projective setting.

For complete simplicial toric varieties defined by a fan we give a strictly combinatorial algorithm to compute the $c_{SM}$ class and Euler characteristic and a second combinatorial algorithm with reduced running time to compute only the Euler characteristic.

We also prove several Bézout type bounds in multi-projective space. An application
of these bounds to obtain a sharper degree bound on a certain system with a natural bi-projective structure is demonstrated.

**Keywords:** Euler characteristic; Chern-Schwartz-MacPherson class; Segre class; Characteristic class; Computer algebra; Computational intersection theory; Algebraic geometry
Acknowledgements

The author of this thesis would like thank Michael Stillman for suggesting to us the problem considered in Chapter 5 of this thesis; that is the problem of computing Chern-Schwartz-MacPherson classes of toric varieties. The author would also like to thank Éric Schost for numerous helpful comments and discussions regarding the content of this thesis.
# Contents

Abstract i

Acknowledgements iii

List of Tables vii

List of Appendices viii

1 Introduction 1

1.1 Overview of Contributions 2

1.1.1 Computing Characteristics Classes 2

1.1.2 Bézout-like Results in Multi-projective Space 7

1.2 Review and Previous Work 8

1.2.1 The Setting 8

1.2.2 The Segre Class 11

1.2.3 The Chern-Schwartz-MacPherson Class and the Euler Characteristic 12

1.3 Results 15

1.3.1 Characteristic Class Computations in $\mathbb{P}^n$ 15

1.3.2 Characteristics Class Computations in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ 24

1.3.3 Computing the Chern-Schwartz-MacPherson Class and Euler Characteristic of Complete Simplicial Toric Varieties 28

1.3.4 Bézout Type Results in Multi-projective Space 31

2 Computing Characteristics Classes in Projective Space 38

2.1 Problem and Setting 41

2.1.1 Chow Groups and Chow Rings 41

2.1.2 Chern and Chern-Schwartz-MacPherson Classes 45

2.2 Review 48

2.2.1 Projective Degrees 49

2.2.2 Segre classes 50
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.3 Chern-Schwartz-MacPherson Classes</td>
<td>54</td>
</tr>
<tr>
<td>2.3 Main Results and Algorithms</td>
<td>55</td>
</tr>
<tr>
<td>2.3.1 Results</td>
<td>56</td>
</tr>
<tr>
<td>2.3.2 Algorithms</td>
<td>61</td>
</tr>
<tr>
<td>2.4 Performance</td>
<td>64</td>
</tr>
<tr>
<td>2.4.1 Timings for the Computation of Segre Classes</td>
<td>67</td>
</tr>
<tr>
<td>2.4.2 Timings for the Computation of $c_{SM}$ Classes and Euler Characteristics</td>
<td>68</td>
</tr>
<tr>
<td>2.4.3 Running Time Bounds</td>
<td>70</td>
</tr>
<tr>
<td>3 An Improved Algorithm for Complete Intersections</td>
<td>77</td>
</tr>
<tr>
<td>3.1 Background</td>
<td>79</td>
</tr>
<tr>
<td>3.2 Main Results and Algorithms</td>
<td>82</td>
</tr>
<tr>
<td>3.2.1 The Main Result</td>
<td>83</td>
</tr>
<tr>
<td>3.2.2 Algorithms</td>
<td>87</td>
</tr>
<tr>
<td>3.3 Performance</td>
<td>89</td>
</tr>
<tr>
<td>4 Algorithms to Compute the Topological Euler Characteristic and the Chern-Schwartz-MacPherson Class of Arbitrary Subschemes of Products of Projective Space</td>
<td>97</td>
</tr>
<tr>
<td>4.1 Review</td>
<td>99</td>
</tr>
<tr>
<td>4.1.1 $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ and its Chow Ring</td>
<td>99</td>
</tr>
<tr>
<td>4.1.2 Previous Segre Class Algorithms in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$</td>
<td>101</td>
</tr>
<tr>
<td>4.1.3 $c_{SM}$ Class of a Hypersurface For a Subscheme of any Smooth Variety</td>
<td>102</td>
</tr>
<tr>
<td>4.2 Main Results</td>
<td>102</td>
</tr>
<tr>
<td>4.2.1 The Segre Class of Subvarieties of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$</td>
<td>103</td>
</tr>
<tr>
<td>4.2.2 Computing the Projective Multi-degrees</td>
<td>105</td>
</tr>
<tr>
<td>4.2.3 The $c_{SM}$ Class of Complete Intersections</td>
<td>110</td>
</tr>
<tr>
<td>4.3 Algorithms for Subschemes of a Product of Projective Spaces</td>
<td>111</td>
</tr>
<tr>
<td>4.3.1 Segre Class</td>
<td>112</td>
</tr>
<tr>
<td>4.3.2 The $c_{SM}$ Class Via Inclusion/Exclusion</td>
<td>114</td>
</tr>
<tr>
<td>4.3.3 $c_{SM}$: The Complete Intersection Case</td>
<td>115</td>
</tr>
<tr>
<td>4.4 Performance</td>
<td>116</td>
</tr>
<tr>
<td>4.4.1 Run Time Tests</td>
<td>116</td>
</tr>
<tr>
<td>4.4.2 Running Time Bounds</td>
<td>120</td>
</tr>
<tr>
<td>5 $c_{SM}$ Classes of Complete Simplicial Toric Varieties</td>
<td>124</td>
</tr>
<tr>
<td>5.1 Review</td>
<td>125</td>
</tr>
<tr>
<td>5.1.1 Constructing the Toric Variety of a Fan</td>
<td>125</td>
</tr>
</tbody>
</table>
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Comparision of algorithms to compute the Segre class of a subscheme of $\mathbb{P}^n$</td>
<td>67</td>
</tr>
<tr>
<td>2.2</td>
<td>Comparison of algorithms to compute the $c_{SM}$ class of a subscheme of $\mathbb{P}^n$ using inclusion/exclusion</td>
<td>70</td>
</tr>
<tr>
<td>3.1</td>
<td>Run time results for Algorithm 3.2.2 working over $\mathbb{Q}$</td>
<td>90</td>
</tr>
<tr>
<td>3.2</td>
<td>Run time results for Algorithm 3.2.2 working over the finite field $\mathbb{F}(32749)$</td>
<td>93</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparision of algorithms to compute the Segre class of a subscheme of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$</td>
<td>118</td>
</tr>
<tr>
<td>4.2</td>
<td>The $c_{SM}$ class and Euler characteristic in multi-projective space</td>
<td>119</td>
</tr>
<tr>
<td>4.3</td>
<td>The $c_{SM}$ class and Euler characteristic of certain complete intersections in multi-projective space</td>
<td>119</td>
</tr>
<tr>
<td>5.1</td>
<td>The $c_{SM}$ class and Euler characteristic of complete simplicial toric varieties</td>
<td>143</td>
</tr>
</tbody>
</table>
List of Appendices

Appendix A ................................. 170
Chapter 1

Introduction

The subject matter of this thesis focuses on two separate but related areas of work. The first area, and that which makes up the bulk of the work, is the computation of characteristic classes. This is the focus of Chapters 2, 3, 4 and 5. Chapter 6 gives the proof of several Bézout-like bounds in multi-projective space with a focus on their application to obtaining refined running time bounds for solving systems of polynomial equations in multi-projective space.

The thesis focuses on the use of intersection theory in computer algebra and on the use of computer algebra to perform computations in intersection theory and algebraic geometry. In Chapters 2, 3, 4 and 5 we use this interplay to construct algorithms for use on a computer algebra system that will allow us to compute important invariants in algebraic geometry by solving zero dimensional polynomial systems in the case of Chapters 2, 3 and 4 and by exploiting the combinatorics of certain algebraic structures in the case of Chapter 5. In Chapter 6 we use this interplay to give us refined degree bounds for affine and projective varieties which can be applied to bound the degrees of the polynomial systems arising in problems in computer algebra.

Macaulay2 [19] implementations of all algorithms for computing characteristics classes described in this thesis can be found at https://github.com/Martin-Helmer/char-class-calc. The implementations of the algorithms from Chap-
ters 2 and 3 are given as part of the “CharClassCalc” package, package syntax is discussed in Appendix A.1. The Macaulay2 [19] implementations for the algorithms from Chapter 4 are given in the “MultiProjChar” package, package syntax is discussed in Appendix A.4. A Macaulay2 [19] implementation of the algorithms presented in Chapter 5 is given in the “CharToric” package, package syntax is discussed in Appendix A.6.

1.1 Overview of Contributions

We now give a short overview of the contributions presented in this thesis. We begin by discussing our contributions to algorithms which compute characteristic classes of algebraic varieties. Next we give an overview of our work on Bézout-like bounds in multi-projective space. In this section, we will use some terms not defined until later; we do this to allow us to give a simple summary of the main results of the thesis. Complete definitions and more details will be given in the following sections.

In this chapter and in the following chapters we shall frequently employ the language of schemes rather than varieties when working with algebraic geometric objects. In the statements of the results given the reader may freely mentally substitute the word “scheme” with the word “variety” and the word “subscheme” with “subvariety” and so on, if desired. An overview of the scheme theoretic terminology used here can be found, for example, in Gathmann [17] or in Eisenbud and Harris [11].

1.1.1 Computing Characteristics Classes

Beginning with Euler’s Polyhedral Formula (circa 1750) the Euler characteristic has developed into an important invariant for the study of topology and geometry in a wide variety of settings. In addition to providing a mechanism to enable the classi-
fication of orientable surfaces, the Euler characteristic is an important component in many results in geometry. More recently several authors have noted applications of the Euler characteristic of projective varieties to problems in statistics and physics. Specifically the Euler characteristic is used when studying problems of maximum likelihood estimation in algebraic statistics by Huh in [22] as well as in the study of problems in string theory by Aluffi and Esole in [6] and by Collinucci, Denef, and Esole in [9].

Let $V$ be a subscheme of a projective space $\mathbb{P}^n$ (over $k$ an algebraically closed field of characteristic zero). One of the first computational approaches to calculate the Euler characteristic of $V$, $\chi(V)$, was to do so by computing Hodge numbers and using the fact that the Euler characteristic is an alternating sum of Hodge numbers. This approach is implemented in Macaulay2 [19] as the function euler, where the Hodge numbers are found by computing the ranks of the appropriate cohomology rings. This approach, however, has significant drawbacks in both applicability and performance. Specifically, this method is only applicable for smooth subschemes and the computation of the cohomology rings and their respective ranks required to determine the Hodge numbers is computationally expensive.

Alternatively, one may obtain the Euler characteristic of $V \subset \mathbb{P}^n$ directly from the Chern-Schwartz-MacPherson class of $V$, $c_{SM}(V)$. In particular, when we consider $c_{SM}(V)$ as an element of the Chow ring of $\mathbb{P}^n$, $A^*(\mathbb{P}^n)$, we have that $\chi(V)$ is equal to the degree of the zero dimensional component of $c_{SM}(V)$. This is the method we shall use to obtain the Euler characteristic. This technique has been used by several authors (e.g. [2], [23], [21]) to construct different algorithms which are capable of calculating Euler characteristics of complex projective varieties. These previous methods will be discussed below.

In addition to containing the Euler characteristic, $c_{SM}$ classes are an important invariant in algebraic geometry, providing a generalization of the Chern class to singular schemes. While there are several other generalizations of the Chern class to singular schemes (i.e. the Chern-Fulton and Chern-Fulton-Johnson classes, see [3] for a discussion of these), the $c_{SM}$ class is the only generalization which preserves
the relation between Chern classes and the Euler characteristic. Additionally the $c_{SM}$ class has unique functorial properties (see Definition 2.1.2) and relationships to other common invariants. The $c_{SM}$ class has also found direct applications to problems from string theory in physics, see for example Aluffi and Esole [5].

The existence of a functorial theory of Chern classes for singular varieties, in terms of a natural transformation from the functor of constructible functions to some nice homology theory, and its relation to the Euler characteristic, was conjectured by Deligne and Grothendieck in the 1960’s. In the 1974 article [26], MacPherson proved the existence of such a transformation, introducing a new notion of Chern classes for singular algebraic varieties. Independently in the 1960’s Schwartz [28] defined a theory of Chern classes for singular varieties in relative cohomology. It was later shown in a paper of Brasselet and Schwartz [8] that these two different notions were in fact equivalent.

The problem we consider in Chapters 2 and 3 is the following. Let $k$ be an algebraically closed field of characteristic zero. Given an ideal $I$ in $k[x_0, \ldots, x_n]$ which defines a subscheme $V = V(I)$ in the projective space $\mathbb{P}^n$, how does one compute the Segre class of $V$ in $\mathbb{P}^n$, $s(V, \mathbb{P}^n)$, the Chern-Schwartz-MacPherson class of $V$, $c_{SM}(V)$ (or Chern class $c_{SM}(V) = c(T_V) \cdot [V]$ if $V$ is smooth) and the Euler characteristic of $V$, $\chi(V)$? Further, how does one compute these invariants in a time efficient manner using a computer algebra system?

Our contributions to the resolution of these questions are described in Chapter 2 and Chapter 3. We give a new expression for the projective degrees of a rational map in Theorem 2.3.1. Applying this theorem we give a new algorithm to compute the projective degrees using a computer algebra system in Algorithm 2.3.1. In Chapter 2 we use Algorithm 2.3.1 to give a method to compute the Segre class of $V$ (Algorithm 2.3.2) and a method to compute the $c_{SM}$ class and/or Euler characteristic of $V$ (Algorithm 2.3.3). These algorithms are then tested on a wide variety of examples and are found to perform favourably in comparison to other known algorithms. The running time results of our algorithms for these examples are summarized in Tables 2.1 and 2.2. We also give running time bounds for Algorithms 2.3.1, 2.3.2 and 2.3.3.
In §2.4.3.

In Chapter 3 we give Algorithm 3.2.1, a new algorithm to compute the $c_{SM}$ class of a complete intersection subscheme of $\mathbb{P}^n$ with a specific structure. Similar to the algorithms in Chapter 2 this procedure will also use our method to compute projective degrees (Algorithm 2.3.1). Algorithm 3.2.1 offers a significant speed up on some examples. We generalize this method to any complete intersection subscheme of $\mathbb{P}^n$ in Algorithm 3.2.2. The new algorithms are tested on a wide selection of examples and found to offer considerable performance improvements for many complete intersection varieties, particularly those defined by an ideal $I$ having the property that the majority of the generators of $I$ defined a smooth hypersurface in $\mathbb{P}^n$ when considered separately.

The Macaulay2 [19] and Sage [29] implementations of our algorithm for computing $c_{SM}$ classes, Euler characteristics and Segre classes of subschemes of projective space can be found at https://github.com/Martin-Helmer/char-class-calc. The Macaulay2 [19] implementation is also available as part of the “CharacteristicClasses” package in Macaulay2 version 1.7 and above and can be accessed using the option “Algorithm=>ProjectiveDegree”, see the Macaulay2 documentation http://www.math.uiuc.edu/Macaulay2/doc/Macaulay2-1.7/share/doc/Macaulay2/CharacteristicClasses/html/ for further details.

In Chapter 4 we generalize all of the algorithms to compute characteristic classes for subschemes of projective space described in Chapters 2 and 3 to the multi-projective setting.

Let $k$ be an algebraically closed field of characteristic zero and let $R$ be the coordinate ring of $\mathcal{P} = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$. Similar to the problem considered for subschemes of $\mathbb{P}^n$, we work with an ideal $I$ in $R$ which defines a subscheme $V = V(I)$ of multi-projective space $\mathcal{P}$. In this setting we devise an algorithm to compute the Segre class of $V$ in $\mathcal{P}$, $s(V, \mathcal{P})$, the Chern-Schwartz-MacPherson class of $V$, $c_{SM}(V)$ (or Chern class $c_{SM}(V) = c(T_V) \cdot [V]$ if $V$ is smooth) and the Euler characteristic of $V$, $\chi(V)$ in a time efficient manner using a computer algebra system.
The main results of Chapter 4 are Theorem 4.2.1 and Theorem 4.2.2. Theorem 4.2.1 provides a new expression for the Segre class \( s(V, \mathbb{P}) \) in terms of certain Chow ring elements which may be computed directly from the projective multi-degrees (which generalize the projective degrees to the multi-projective setting). The result of Theorem 4.2.1 generalizes a previous result of Aluffi [2], given below as Proposition 2.2.1, which gives an expression for the Segre classes in \( \mathbb{P}^n \). In Theorem 4.2.1 we give a new method to compute the projective multi-degrees which can be easily implemented on a computer algebra system. These results allow us to construct algorithms which compute \( s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \), \( c_{SM}(V) \) and \( \chi(V) \) for \( V \) a subscheme of \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \).

In Chapter 5 we present Algorithm 5.3.1 which gives a combinatorial algorithm to compute the Chern-Schwartz-MacPherson class and Euler characteristic and Algorithm 5.3.2 which gives a combinatorial algorithm to compute only the Euler characteristic of a complete simplicial toric variety \( X_\Sigma \) specified by a fan \( \Sigma \).

Both Algorithm 5.3.1 and Algorithm 5.3.2 are strictly combinatorial, since they use only the structure of the fan \( \Sigma \) to compute the \( c_{SM} \) class and Euler characteristic. As such the running times of the algorithms are not dependent on the algebraic degrees of the defining equations of the variety \( X_\Sigma \). Additionally, unlike the algorithms presented in previous chapters, Algorithm 5.3.1 and Algorithm 5.3.2 do not require us to use Gröbner bases or other polynomial system solving tools to find \( c_{SM}(X_\Sigma) \).

The main ingredient in the construction of these algorithms is a result of Barthel, Brasselet and Fieseler [7] which we state in Proposition 5.3.1 below. This result gives an expression for \( c_{SM}(X_\Sigma) \) in terms of the Chow ring classes of the orbit closures. These Chow ring classes can be easily computed in the case of complete simplicial toric varieties using standard results such as Theorem 12.5.2. of Cox, Little, and Schenck [10] (given as Proposition 5.3.3 below).


1.1.2 Bézout-like Results in Multi-projective Space

The problem originally investigated by Bézout considered the number of intersection points of two algebraic curves in the plane. In 1916 Macaulay [25] published a more general result giving the number of intersection points of \( n \) hypersurfaces which intersect transversally in \( \mathbb{P}^n \) as the product of the degrees of the hypersurfaces. In the modern literature, the term Bézout theorem is used to refer to a wide class of theorems concerning the intersections of arbitrary varieties or schemes in a certain projective space, \( \mathbb{P}^n \), and in particular to bound, or give an expression for, the degree of the intersection scheme.

For two intersecting curves in the projective plane, Bézout’s theorem tells us that the number of points in the intersection counted with multiplicity is equal to the product of the degrees of the curves. Results of this type in projective space \( \mathbb{P}^n \) have been studied intensively both classically and in modern algebraic geometry and intersection theory. A typical statement of a Bézout bound for subvarieties \( V_1, \ldots, V_r \) of \( \mathbb{P}^n \) can be found, for example, in Fulton [16, §8.4.6]. Let \( W_1, \ldots, W_t \) be the irreducible components of \( \bigcap_{i=1}^r V_i \), then we have:

\[
\sum_{i=1}^t \deg(W_i) \leq \prod_{i=1}^r \deg(V_i). \tag{1.1}
\]

Bézout type bounds in bi-projective, \( \mathbb{P}^{n_1} \times \mathbb{P}^{n_2} \), and multi-projective space, \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \) have proved to be both more nuanced and more difficult, partially due to the more complicated structure of the Chow ring. While there are several results that one could call “a Bézout type bound” it is not clear that there is one specific result that one could call “the Bézout bound” in the multi-projective setting.

The problem we consider in Chapter 6 is the following. Given a collection of hypersurfaces \( V_1, \ldots, V_r \) in the multi-projective space \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \) defined by multi-homogeneous polynomials, how do we bound the degrees of the sum of the irreducible components of the intersection \( V = V_1 \cap \cdots \cap V_r \) (counted with multiplicity) by some expression involving only degrees of some selection of the hyper-
surfaces $V_1, \ldots, V_r$? We also investigate how such a bound can be constructed in such a way as to give a refined bound on the degree of the irreducible components (with multiplicity) of a given affine or projective variety with an inherent multi-projective structure. Further, we would like our result to be phrased in such a way as to be advantageous for use to obtain complexity bounds on algorithms in computer algebra, and hence we would like the terms in the upper bound to be easily computable and for the notion of multiplicity to be compatible with existing results giving complexity bounds for solving systems of polynomial equations.

The motivating example for the work in Chapter 6 comes from a problem considered by Safey El Din and Trebuchet in [13] when developing an algorithm to compute at least one point in each connected component of a smooth real algebraic set. The type of systems considered by the algorithm of [13] have a natural bi-projective structure, because of this using the bi-projective Bézout-like results of Chapter 6 will give a sharper degree bound than using the usual projective Bézout bound.

## 1.2 Review and Previous Work

In this section we establish the setting for this work and discuss several previous results we will employ in later sections as well as discuss some previous algorithms to compute characteristic classes in projective spaces.

### 1.2.1 The Setting

A locally ringed space $(X, O_X)$ is a pair consisting of a topological space $X$ and a sheaf of rings $O_X$ all of whose stalks are local rings. An affine scheme is a locally ringed space which is isomorphic to the spectrum of a commutative ring. By the spectrum of a commutative ring $R$ we mean the set of all prime ideals in $R$, this will be denoted Spec($R$). In this way we may consider the affine space
\( \mathbb{A}^n \equiv \text{Spec}(k[x_1, \ldots, x_n]) \) as an affine scheme for \( k \) some algebraically closed field of characteristic zero. A scheme is a locally ringed space \( X \) covered by open sets \( U_i \) such that the restriction of the structure sheaf \( O_X \) to each \( U_i \) is isomorphic to an affine scheme. Put another way a scheme is obtained by gluing together affine schemes in the Zariski topology. The example of a scheme obtained by gluing affine schemes which we will most frequently use is that of a projective space \( \mathbb{P}^n = \text{Proj}(k[x_0, \ldots, x_n]) \) (over \( k \) an algebraically closed field of characteristic zero) which we can think of as a scheme obtained by glueing the affine schemes \( \mathbb{A}^n \). For a more complete discussion see, for example, Gathmann [17] or Eisenbud and Harris [11].

Characteristics classes will be considered as elements of some Chow ring. The Chow ring of a smooth (irreducible) variety \( M \) will be denoted \( A^*(M) \). For a general definition see §2.1.1. When working with Chow groups and Chow rings by variety we will mean a reduced and irreducible scheme. A subvariety of a scheme will be taken to mean a reduced and irreducible subscheme.

In Chapters 2 and 3 we consider \( V = V(I) \) to be a subscheme of a projective space \( \mathbb{P}^n \) over an algebraically closed field of characteristic zero defined by a homogeneous ideal \( I \) in \( k[x_0, \ldots, x_n] \). The characteristics classes \( c_{SM}(V) \) and \( s(V, \mathbb{P}^n) \) will be represented as elements of the Chow ring of \( \mathbb{P}^n, A^*(\mathbb{P}^n) \).

The Chow ring of \( \mathbb{P}^n \) may be expressed as \( A^*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1}) \) where \( h \) is the rational equivalence class of a hyperplane in \( \mathbb{P}^n \), hence a hypersurface \( W \) of degree \( d \) in \( \mathbb{P}^n \) is represented as \( [W] = d \cdot h \) in \( A^*(\mathbb{P}^n) \). We will always use the presentation \( \mathbb{Z}[h]/(h^{n+1}) \) to represent the Chow ring \( A^*(\mathbb{P}^n) \), and hence \( s(V, \mathbb{P}^n) \) and \( c_{SM}(V) \) will be polynomials in \( h \) with the term containing \( h^n \) representing the dimension zero (codimension \( n \)) component, \( h^{n-1} \) representing the dimension one (codimension \( n-1 \)) component and so on.

The Euler characteristic will be given as an integer and is equal to the degree of zero dimensional component of \( c_{SM}(V) \), that is the coefficient of \( h^n \) in the polynomial
representation of \( c_{SM}(V) \). We will express this as

\[
\chi(V) = \int c_{SM}(V).
\]

In Chapter 4 and Chapter 6 we will frequently work in the Chow ring of multi-projective space \( \mathbb{P} = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \). The Chow ring of \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \) may be expressed as

\[
A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \cong \mathbb{Z}[h_1, \ldots, h_m]/(h_1^{n_1+1}, \ldots, h_m^{n_m+1}),
\]

where \( h_i \) is the rational equivalence class of a general hyperplane in \( \mathbb{P}^{n_i} \) (more precisely \( h_i \) is the rational equivalence class of the pullback under the projection map \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \to \mathbb{P}^{n_i} \) of a general hyperplane in \( \mathbb{P}^{n_i} \) for \( i = 1, \ldots, m \)).

Let \( V = V(I) \) be a subscheme of multi-projective space \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \) over an algebraically closed field of characteristic zero. The characteristics classes \( c_{SM}(V) \) and \( s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \) which we compute in Chapter 4 will be expressed as elements of the Chow ring \( A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \). As with the projective case we may immediately obtain the Euler characteristic from \( c_{SM}(V) \). Specifically we have that

\[
\chi(V) = \int c_{SM}(V),
\]

which means that \( \chi(V) \) will be equal to the integer coefficient of \( h_1^{n_1} \cdots h_m^{n_m} \) in \( c_{SM}(V) \), where \( c_{SM}(V) \) is considered as an element of the Chow ring \( A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \).

For a complete simplicial toric variety \( X_\Sigma \) defined by a fan \( \Sigma \) the class \( c_{SM}(X_\Sigma) \) will be considered as a class in the rational Chow ring \( A^*(X_\Sigma)_Q \) of \( X_\Sigma \). The structure of this Chow ring is determined by the structure of the fan \( \Sigma \). For a definition see §5.2.
1.2.2 The Segre Class

The Segre class is an important invariant in intersection theory in algebraic geometry, both because it contains important intersection theoretic information and because it can be used to construct other commonly studied structures and invariants. For example, for $V$ an irreducible subvariety of a variety $W$ the Segre class $s(V, W)$ contains the Samuel (or algebraic) multiplicity of $V$ in $W$ (see Fulton [16, §4.3]). Additionally the Segre class is important in Fulton’s construction of the intersection product ([16, §6]) in the Chow ring and important invariants such as the Chern-Fulton and the Chern-Fulton-Johnson class (in some contexts, see (3.3)) and the Chern-Schwartz-MacPherson class (see Proposition 4.1.2) may be defined in terms of Segre classes.

For $V$ a proper closed subscheme of a variety $W$, we may define the Segre class of $V$ in $W$ as

$$s(V, W) = \sum_{j \geq 1} (-1)^{j-1} \eta_*(\tilde{V}^j) = \eta_*\left(\frac{[\tilde{V}]}{1 + [\tilde{V}]}\right) \in A^*(V) \quad (1.3)$$

where $\tilde{V}$ is the exceptional divisor of the blow-up of $W$ along $V$, $Bl_V W$, $\eta : \tilde{V} \to V$ is the projection, the class $\tilde{V}^k$ is the $k$-th self intersection of $\tilde{V}$ and $[\tilde{V}]$ is the class of $\tilde{V}$ in the Chow ring of the blow-up, $A^*(Bl_V W)$. See Fulton [16, §4.2.2] for further details.

We note that any algorithm to compute the Segre class will immediately give us an algorithm to compute the Chern-Fulton class $c_F$ (referred to as the Canonical class by Fulton [16]) of a subscheme $V$ of a smooth variety $M$ over an algebraically closed field. Specifically we have that

$$c_F(V) = c(T_M) \cdot s(V, M) \in A^*(M). \quad (1.4)$$

The Chern-Fulton class $c_F$ is a generalization of the Chern class to singular schemes, see, for example, Fulton [16, Examples 4.2.6, 19.1.7]. In particular then, any method to compute the Segre class will also give the Chern class $c(V) = c(T_V) \cdot [V]$ in the case where $V$ is a smooth subscheme of $M$, also see Eklund, Jost and Peterson.
Previously algorithms have been given by Allufi [2] and by Eklund, Jost, and Peterson [12] to compute the Segre class in $\mathbb{P}^n$. To compute $s(V, \mathbb{P}^n)$ the algorithm of Allufi [2] requires the computation of the blowup of $\mathbb{P}^n$ along $V$, i.e. requires the computation of the Rees algebra. This is an expensive operation in general. The algorithm of Eklund, Jost, and Peterson [12] works by computing certain residual sets via saturation and then computing their degrees. For a more detailed comparison of these methods with our method using projective degrees see Chapter 2.

In the multi-projective setting a previous algorithm of Moe and Qviller [27] which computes the Segre class of a subscheme of a smooth projective toric variety could be applied. This algorithm generalizes the algorithm of Eklund, Jost, and Peterson [12]. The algorithm of Moe and Qviller [27], however, does not make use of the special structure of the Chow ring of multi-projective space and hence performs extra, unnecessary, computations in the multi-projective case. A performance comparison with our algorithm to compute the Segre class in $\mathbb{P}^{n_1} \times \ldots \times \mathbb{P}^{n_m}$ can be found in Table 4.1.

### 1.2.3 The Chern-Schwartz-MacPherson Class and the Euler Characteristic

A general definition of the Chern-Schwartz-MacPherson class is given in Definition 2.1.2. Rather than giving the general definition here we instead focus on giving a more intuitive understanding of the geometric information contained in the $c_{SM}$ class and on some methods for its computation. Note that in this subsection, for simplicity, we will restrict our discussion to subschemes of $\mathbb{P}^n$.

We first recall a result of Aluffi [4] which states that when $V$ is a subscheme of $\mathbb{P}^n$ then $c_{SM}(V)$ contains the Euler characteristics of $V$ and those of general linear sections of $V$ for each codimension. In this way one may consider $c_{SM}(V)$ as a more refined version of the Euler characteristic. Specifically, if $\dim(V) = m$, starting from
c_{SM}(V) we may directly obtain the list of invariants

\[ \chi(V), \chi(V \cap L_1), \chi(V \cap L_1 \cap L_2), \ldots, \chi(V \cap L_1 \cap \cdots \cap L_m) \]

where \( L_1, \ldots, L_m \) are general hyperplanes. Conversely from the list of Euler characteristics above we could obtain \( c_{SM}(V) \), i.e. there exists an involution between the Euler characteristics of general linear sections and the \( c_{SM} \) class in this setting. This relationship is given explicitly in Theorem 1.1 of Aluffi [4]; we give an example of this below.

**Example 1.2.1.** Consider the subvariety of \( \mathbb{P}^4 \) given by \( V = V(4x_3x_2x_4x_1-x_0^3x_1, x_0x_1x_3x_4-x_3^2x_3) \). In Example 1.3.2 we will compute that \( c_{SM}(V) = 5h^4 + 8h^3 + 12h^2 \). To obtain the Euler characteristics of the general linear sections of \( V \) we may apply an involution formula given by Aluffi in [4, Theorem 1.1], specifically:

- First consider the polynomial \( p(t) = 5 + 8t + 12t^2 \in \mathbb{Z}[t]/(t^5) \) given by the coefficients of the \( c_{SM} \) class above.

- Next apply Aluffi’s involution

\[ p(t) \mapsto I(p) := \frac{t \cdot p(-t - 1) + p(0)}{t + 1} = 12t^2 + 4t + 5. \]

This gives \( \chi(V) = 5, \chi(V \cap L_1) = (-1) \cdot 4 = -4, \) and \( \chi(V \cap L_1 \cap L_2) = (-1)^3 \cdot 12 = 12 \) where \( L_1 \) and \( L_2 \) are general hyperplanes in \( \mathbb{P}^4 \).

The general result of Aluffi [4] relating the \( c_{SM} \) class and the Euler characteristic in \( \mathbb{P}^n \) can be found below in Theorem 2.1.5.

We now discuss the computation of \( c_{SM} \) classes and hence of Euler characteristics of subschemes of projective space, beginning with the case of a hypersurface.

Consider the hypersurface \( V(f) \subset \mathbb{P}^n \) defined by the homogeneous polynomial \( f \). To compute \( c_{SM}(V(f)) \) one may employ [2, Theorem 2.1], which may be expressed
as
\[
c_{SM}(V(f)) = (1 + h)^{n+1} - \sum_{j=0}^{n} g_j(-h)^j(1 + h)^{n-j} \text{ in } A^*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1}). \tag{1.5}
\]

This result has been used to yield several different computational methods to calculate the \(c_{SM}\) class. The differences between the methods lay in how the \(g_j\)'s are understood and computed. The first algorithm to compute \(c_{SM}(V(f))\) was that of Aluffi [2]. To compute the \(g_j\)'s this algorithm requires the computation of the blowup of \(\mathbb{P}^n\) along the singularity subscheme of \(V(f)\) (that is the scheme defined by the partial derivatives of \(f\)). Hence the cost of computing the \(c_{SM}\) class of a hypersurface using the method of Aluffi is that of computing the Ress algebra of the ideal defining the singularity subscheme of the hypersurface. This can be a quite expensive operation, making this algorithm impractical for many examples.

Another algorithm to compute the \(c_{SM}\) class of a hypersurface was given by Jost in [23]. This method makes use of Fulton’s residual intersection theorem (Theorem 9.2 of Fulton [16]) which allows Jost to consider the \(g_j\)'s in (1.5) as the degrees of Fulton’s residual scheme. Jost also shows that in the context of \(c_{SM}\) (and Segre) class computations these residual schemes can be computed by finding a particular saturation. Hence the computation of the saturation to find the residual scheme and the computation of its degree are the main costs of Jost’s algorithm. The algorithm of Jost is probabilistic and yields the correct result for a choice of objects lying in an open dense Zariski set of the corresponding parameter space, see Jost [23] or Eklund, Jost, and Peterson [12].

In Chapter 2 we present Algorithm 2.3.3, in which we consider the \(g_j\)'s as the projective degrees of a rational map defined by the partial derivatives of \(f\). As with the method of Jost [23] our method is probabilistic and yields the correct result for a choice of objects lying in an open dense Zariski set of the corresponding parameter space.

For \(V\) a possibly singular subscheme of \(\mathbb{P}^n\) all these methods require the use of the inclusion/exclusion property of \(c_{SM}\) classes when \(V\) has codimension higher than
one. Specifically for \( V_1, V_2 \) subschemes of \( \mathbb{P}^n \) the inclusion/exclusion property for \( c_{SM} \) classes states

\[
c_{SM}(V_1 \cap V_2) = c_{SM}(V_1) + c_{SM}(V_2) - c_{SM}(V_1 \cup V_2). \tag{1.6}
\]

This property may easily be extended to compute \( c_{SM} \) classes of any codimension, see Proposition 2.1.3.

While the use of this property allows for the computation of \( c_{SM}(V) \) for \( V \) of any codimension, it requires exponentially many \( c_{SM} \) computations relative to the number of generators of \( I \). Additionally some of the schemes considered while performing inclusion/exclusion may have significantly higher degree than the original scheme \( V \).

## 1.3 Results

In this section we provide a more detailed introduction to the results presented in this thesis. For the introduction we will focus on presenting examples where possible.

### 1.3.1 Characteristic Class Computations in \( \mathbb{P}^n \)

The main result of Chapter 2 is Theorem 2.3.1 which gives an expression for the projective degrees of a rational map associated to a homogeneous ideal. We use this result to construct Algorithm 2.3.1 which computes the projective degrees of a rational map defined by an ideal. We then use Algorithm 2.3.1 to construct Algorithms 2.3.2 and 2.3.3 which compute the Segre class and the \( c_{SM} \) class (and hence the Euler characteristic as well).
Projective Degrees of a Rational Map

Consider a rational map \( \phi : \mathbb{P}^n \to \mathbb{P}^m \). In the manner of Harris (Example 19.4 of [20]) we may define the projective degrees of the map \( \phi \) as a list of integers \((g_0, \ldots, g_n)\) where

\[
g_i = \text{card} \left( \phi^{-1} \left( \mathbb{P}^{m-i} \right) \cap \mathbb{P}^i \right),
\]

where \( \mathbb{P}^{m-i} \subset \mathbb{P}^m \) and \( \mathbb{P}^i \subset \mathbb{P}^n \) are general hyperplanes of dimension \( m-i \) and \( i \) respectively and card is the cardinality of a zero dimensional set.

We give a method to compute the projective degrees of a rational map in Theorem 2.3.1 below. This method will form the basis for our algorithms to compute characteristic classes for subschemes of \( \mathbb{P}^n \). Let \( I = (f_0, \ldots, f_m) \) be a homogeneous ideal in \( k[x_0, \ldots, x_n] \); if we consider a rational map \( \phi : \mathbb{P}^n \to \mathbb{P}^m \) associated to the ideal \( I \) which is defined by,

\[
\phi : p \mapsto (f_0(p) : \cdots : f_m(p)),
\]

then Theorem 2.3.1 tells us that \( g_0 = 1 \) and that

\[
g_i = \dim_k (k[x_0, \ldots, x_n, T]/(P_1 + \cdots + P_i + L_A + S)).
\]

Here \( P_\ell, L_\ell, L_A \) and \( S \) are ideals in \( k[x_0, \ldots, x_n, T] \); the ideals \( P_\ell \) are generated by a general linear combination of \( f_0, \ldots, f_m \), the ideals \( L_\ell \) are generated by general homogeneous linear forms in \( k[x_0, \ldots, x_n] \), the ideal \( L_A \) is generated by an affine linear form in \( k[x_0, \ldots, x_n] \) and the ideal \( S \) is given by

\[
S = \left( 1 - T \left( \sum_{j=0}^m \lambda_j f_j \right) \right),
\]

where \( \sum_{j=0}^m \lambda_j f_j \) is a general linear combination of \( f_0, \ldots, f_m \).

We use this result to construct a probabilistic algorithm to compute the projective degrees of a rational map specified by an ideal in Algorithm 2.3.1. The algorithm will give the correct result for a general choice of constants, i.e. for constants in \( k \) chosen from an open dense Zariski set.
Segre Classes

Assume that $V$ is a subscheme of $\mathbb{P}^n$ over $k$, an algebraically closed field of characteristic zero, and that $V$ is defined by a homogeneous ideal $I = (f_0, \ldots, f_m)$ in $k[x_0, \ldots, x_n]$. Adapting Proposition 3.1 of Aluffi [2] (given as Proposition 2.2.1 below) to this case we have that the Segre class $s(V, \mathbb{P}^n)$ can be written in terms of the projective degrees of the rational map associated to the ideal $I$.

In Algorithm 2.3.2 we give a method to compute the Segre class $s(V, \mathbb{P}^n)$ for $V$ a subscheme of $\mathbb{P}^n$ from the projective degrees of the rational map associated to $I$. Specifically our algorithm first computes the projective degrees by applying Algorithm 2.3.1 and then uses these to construct $s(V, \mathbb{P}^n)$ using Proposition 2.2.1.

We give an example illustrating the process used by Algorithm 2.3.2 in Example 1.3.1 below.

**Example 1.3.1.** Let $V = V(I)$ be the subvariety of $\mathbb{P}^4$ defined by the ideal $I = (4x_3x_2x_4x_1 - x_0^3x_1, x_0x_1x_3x_4 - x_2^3x_3) = (f_0, f_1)$ in $k[x_0, x_1, x_2, x_3, x_4]$. $V$ has dimension two and the singularity subscheme of $V$ also has dimension two (by singularity subscheme we mean the subscheme of $V$ defined by the $2 \times 2$ minors of the Jacobian matrix of $I$). Also set $d = \deg(f_0) = \deg(f_1) = 4$.

Recall that we may write the Chow ring of $\mathbb{P}^n$ as $A^*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1})$ where $h$ is the rational equivalence class of a hyperplane, meaning a hypersurface $W$ of degree $d$ in $\mathbb{P}^n$ is represented as $[W] = d \cdot h$ in $A^*(\mathbb{P}^n)$.

We first compute the Segre class $s(V, \mathbb{P}^4)$ of $V$ in $\mathbb{P}^4$ considered as an element of $A^*(\mathbb{P}^4) \cong \mathbb{Z}[h]/(h^5)$. We will follow the procedure of Algorithm 2.3.2. This algorithm is probabilistic in the same manner as Algorithm 2.3.1, our algorithm for computing projective degrees. Consider the rational map $\phi : \mathbb{P}^4 \dasharrow \mathbb{P}^1$ defined by the ideal $I$, that is

$$\phi : p \mapsto (f_0(p) : f_1(p)).$$

We may compute the projective degrees $(g_0, g_1, g_2, g_3, g_4)$ of this rational map (see (1.7)) using Theorem 2.3.1. Let $R = k[x_0, x_1, x_2, x_3, x_4, T]$. Theorem 2.3.1 gives us
\[ g_1 = \dim_k(R/(P_1 + L_1 + L_2 + L_3 + L_A + S)) \]

where \( P_1 = (7f_0 + 9f_1) \) is the ideal in \( R \) defined by a general linear combination of the generators of \( I \);

\[ L_1 = (-11x_0 + 21x_1 - 3x_2 - 18x_3 + 22x_4) \]
\[ L_2 = (31x_0 - 23x_1 + 2x_2 + 47x_3 - 43x_4) \]
\[ L_3 = (13x_0 - 52x_1 - 29x_2 + 71x_3 - 15x_4) \]

are ideals in \( R \) defined by general homogeneous linear forms in \( k[x_0, x_1, x_2, x_3, x_4] \).

\[ L_A = (17 - 14x_0 + 41x_1 + 12x_2 - 91x_3 - 3x_4) \]

is an ideal in \( R \) defined by an affine general linear form in \( k[x_0, x_1, x_2, x_3, x_4] \), and \( S \) is the ideal of \( R \) given by \( S = (1 - T(3f_0 - 5f_1)) \). The expression \( 3f_0 - 5f_1 \) in the definition of \( S \) is a general linear combination of the generators of \( I \). This gives \( g_1 = 4 \). In a similar manner we may compute the remaining projective degrees to obtain

\[ (g_0, g_1, g_2, g_3, g_4) = (1, 4, 0, 0, 0). \]

Applying the formula in (2.24) expressing the Segre class in terms of the projective degrees we obtain

\[ s(V, \mathbb{P}^n) = 1 - \sum_{i=0}^{n} \frac{g_i h^i}{(1 + dh)^{i+1}} \]
\[ = 1 - \frac{1}{1 + 4h} - \frac{4h}{(1 + 4h)^2} \]
\[ = 768h^4 - 128h^3 + 16h^2 \in A^*(\mathbb{P}^4). \]

In Table 2.1 we compare our algorithm to compute the Segre class \( s(V, \mathbb{P}^n) \) using
the projective degrees (Algorithm 2.3.2) to other known algorithms and find that in most cases Algorithm 2.3.2 performs favourably. In Corollary 2.4.2 we give a running time bound for Algorithm 2.3.2. The other known algorithms to compute Segre classes do not have known running time bounds.

The Chern-Schwartz-MacPherson Class and the Euler Characteristic

In Algorithm 2.3.3 we present an algorithm to compute \( c_{SM} \) classes using the projective degrees and inclusion/exclusion. Running time bounds for this algorithm are given in Corollary 2.4.3. The other known algorithms to compute \( c_{SM} \) classes do not have known running time bounds.

We now give an example of computing the \( c_{SM} \) class and Euler characteristic using Algorithm 2.3.3.

Example 1.3.2. As in Example 1.3.1 we take \( V = V(I) \) be the subvariety of \( \mathbb{P}^4 \) defined by the ideal \( I = (4x_3x_2x_4x_1 - x_0^3x_1, x_0x_1x_3x_4 - x_0^3x_3) \) in \( k[x_0, x_1, x_2, x_3, x_4] \). By the inclusion/exclusion property of \( c_{SM} \) classes (1.6) we have that

\[
c_{SM}(V) = c_{SM}(V(f_0)) + c_{SM}(V(f_1)) - c_{SM}(V(f_0 \cdot f_1)). \tag{1.9}
\]

We first calculate \( c_{SM}(V(f_0)) \); we begin by finding the projective degrees of the map corresponding to the ideal \( J \) generated by the partial derivatives of \( f_0 \)

\[
J = (\nabla f_0) = (3x_0^2x_1, -x_0^3 + 4x_2x_3x_4, 4x_1x_3x_4, 4x_1x_2x_4, 4x_1x_2x_3),
\]

that is we must find the projective degrees \( (g_0, g_1, g_2, g_3, g_4) \) of the rational map \( \varphi : \mathbb{P}^4 \to \mathbb{P}^4 \) (sometimes referred to as the polar or gradient map (2.18)) given by

\[
\varphi : (p_0 : p_1 : p_2 : p_3 : p_4) \to (3p_0^2p_1 : -p_0^3 + 4p_2p_3p_4 : 4p_1p_3p_4 : 4p_1p_2p_4 : 4p_1p_2p_3).
\]

In this example we will show the computation of \( g_2 \). By Corollary 2.3.3, \( g_0 = 1 \) and
we computed $g_1 = 3$. Now compute

$$g_2 = \dim_k (k[x_0, x_1, x_2, x_3, x_4, T]/(P_1 + P_2 + L_1 + L_2 + L_A + S)),$$

where $P_1$ and $P_2$ are the ideals in $R$ generated by a general linear combination of the generators of $J$; $L_1, L_2$ are ideals of $R$ generated by homogeneous linear forms in $k[x_0, x_1, x_2, x_3, x_4]$ and $L_A$ is an ideal in $R$ given by a general affine form in $k[x_0, x_1, x_2, x_3, x_4]$ and finally $S$ is the ideal in $R$ given by

$$S = \left(1 - T \left(7(3x_0^2 x_1) + 15(-x_0^3 + 4x_2 x_3 x_4) - 13(4x_1 x_3 x_4) + 24(4x_1 x_2 x_4) - 3(4x_1 x_2 x_3)\right)\right).$$

This gives $g_2 = 6$. Again applying Corollary 2.3.3 we find the other projective degrees are $(g_0, g_1, g_2, g_3, g_4) = (1, 3, 6, 6, 2)$. By (1.5) this gives us that

$$c_{SM}(V(f_0)) = (1 + h)^{n+1} - \sum_{j=0}^{n} g_j (-h)^j (1 + h)^{n-j}$$

$$= (1 + h)^5 - \sum_{j=0}^{4} g_j (-h)^j (1 + h)^{4-j}$$

$$= 5h^4 + 9h^3 + 7h^2 + 4h \in A^*(\mathbb{P}^4).$$

Similarly we find that the projective degrees corresponding to $f_1$, and $f_0 f_1$ are $(1, 3, 6, 6, 2)$ and $(1, 7, 23, 29, 12)$ respectively. This gives the $c_{SM}$ classes:

$$c_{SM}(V(f_1)) = 5h^4 + 9h^3 + 7h^2 + 4h,$$

$$c_{SM}(V(f_0 f_1)) = 5h^4 + 10h^3 + 2h^2 + 8h.$$  

Combining these we obtain

$$c_{SM}(V) = 5h^4 + 8h^3 + 12h^2 \in A^*(\mathbb{P}^4) \cong \mathbb{Z}[h]/(h^5).$$

From this we may immediately obtain that the Euler characteristic of $V$ is $\chi(V) = 5$ since the Euler characteristic of $V$ is the degree of the zero dimensional component of
We now given an example of using this result in the manner presented in Algorithm 3.2.1 to compute the $c_{SM}$ class.

**Example 1.3.3.** Let $I = (3x_0^3 + 5x_1^3 + 2x_2^3 - 9x_3^3 + 7x_4^3, -x_2^2x_3^2 + x_0x_1x_4^2)$ and let $V = V(I)$, compute $c_{SM}(V)$ using Algorithm 3.2.1. Note that $V(3x_0^3 + 5x_1^3 + 2x_2^3 - 9x_3^3 + 7x_4^3)$ is smooth, hence Algorithm 3.2.1 can be used directly. First compute the singularity subscheme $Y$ of $V$, the Jacobian matrix of $I$ is:

$$\text{Jac}(I) = \begin{bmatrix} 9x_0^2 & 15x_1^2 & 6x_2^3 & -27x_3^2 & 21x_4^2 \\ x_1x_4^2 & x_0x_4^2 & -2x_2x_3 & -2x_3^2x_4 & 2x_0x_1x_4 \end{bmatrix}$$

let $\tilde{J}$ be the ideal generated by the $2 \times 2$ minors of $\text{Jac}(I)$ and compute

$$J = (\tilde{J} + I) : (x_0, x_1, x_2, x_3, x_4)^\infty$$

$$= (x_1x_4^2, x_0x_4^2, x_2x_3x_4, x_0x_1x_4, x_2x_3^2, x_2^2x_3, 3x_0^3 + 5x_1^3 + 2x_2^3 - 9x_3^3 + 7x_4^3)$$

Hence we have that $Y = V(J)$ is the singularity subscheme of $V$. Note that $\dim Y = 1$, hence $V$ is not smooth. Now compute the Segre class $s(Y, \mathbb{P}^n)$ using Algorithm 2.3.2. We consider the projective degrees $(g_0, g_1, g_2, g_3, g_4)$ of the rational map $\phi$:
\( \mathbb{P}^4 \rightarrow \mathbb{P}^6 \) defined by the ideal \( J \), we may compute these projective degrees using Theorem 2.3.1, for this example we will show the computation of \( g_3 \).

Let \( R = k[x_0, x_1, x_2, x_3, x_4, T] \); from Theorem 2.3.1 we have

\[
g_3 = \dim_k (R/(P_1 + P_2 + P_3 + L_1 + L_A + S)),
\]

where \( P_1, P_2, P_3 \) are ideals in \( R \) defined by general linear combinations of the generators of \( J \), \( L_1 \) is an ideal in \( R \) defined by a general homogeneous linear form in \( k[x_0, x_1, x_2, x_3, x_4] \), \( L_A \) is an ideal in \( R \) defined by a general affine linear form in \( k[x_0, x_1, x_2, x_3, x_4] \) and \( S \) is the ideal in \( R \) defined by \( S = (1 - T(\lambda_0 J_0 + \cdots + \lambda_6 J_6)) \) where \( J_0, \ldots, J_6 \) are the generators of \( J \) and \( \lambda_0 J_0 + \cdots + \lambda_6 J_6 \) is a general linear combination. This gives \( g_3 = 21 \); the remaining projective degrees may be obtained in a similar fashion, giving \((g_0, g_1, g_2, g_3, g_4) = (1, 3, 9, 21, 24) \). Now using (2.24) as in Algorithm 3.2.1 we obtain

\[
s(Y, \mathbb{P}^4) = 1 - \sum_{i=0}^{4} \frac{g_i h^i}{(1 + 3h)^{i+1}}
= -15h^4 + 6h^3 \in A^*(\mathbb{P}^4).
\]

In the notation of Theorem 3.2.1 this gives \((s_0, s_1, s_2, s_3, s_4) = (0, 0, 6, -15) \).

Again using the notation of Theorem 3.2.1 we note that \( \prod_{i=0}^{4}(1 + \deg(f_i)h) = (1 + 3h)(1 + 4h) = 12h^2 + 7h + 1 \), hence we have that \( \bar{c}_0 = 1, \bar{c}_1 = 7 \) and \( \bar{c}_2 = 12 \).

We may now calculate \( c_{SM}(V) \) by applying Theorem 3.2.1, this gives

\[
c_{SM}(V) = (1 + h)^5 \cdot \frac{3h}{1 + 3h} \cdot \frac{4h}{1 + 4h} + \frac{(1 + h)^5}{(1 + 3h)(1 + 4h)} \left( \sum_{p=0}^{2} h^p \sum_{i=0}^{p} \binom{2 + 1 - i}{p - i} (-1)^i 4^{p-i} \cdot \bar{c}_i \right) \cdot \left( \sum_{i=0}^{4} (-1)^i \bar{c}_i \right).
\]

Simplifying we obtain

\[
c_{SM}(V) = 81h^4 - 18h^3 + 12h^2.
\]
The Euler characteristic \( \chi(V) = 81 \) is given by the degree of the zero dimensional component of \( c_{SM}(V) \).

In Proposition 3.2.2 we give a modified version of the inclusion/exclusion property which considers only the singular generators of an ideal, specifically we show the following. Let \( Z \subset \mathbb{P}^n \) be smooth (scheme-theoretically) and let \( X_1 = V(f_1) \), \( X_2 = V(f_2) \) be singular hypersurfaces in \( \mathbb{P}^n \). If \( V = Z \cap X_1 \cap X_2 \), then we have

\[
c_{SM}(V) = c_{SM}(Z \cap X_1) + c_{SM}(Z \cap X_2) - c_{SM}(Z \cap (X_1 \cup X_2)),
\]

(1.11)

here \( X_1 \cup X_2 \) is the scheme generated by \( f_1 \cdot f_2 \). Additionally, when \( V \) is a complete intersection each of the terms in (3.9) can be computed using Theorem 3.2.1.

Using this result and Algorithm 3.2.1 we devise Algorithm 3.2.2 which is applicable for any globally complete intersection subscheme of \( \mathbb{P}^n \). Algorithm 3.2.2 uses the specialized version of inclusion/exclusion (1.11) to break up the \( c_{SM} \) class computation into a sum of \( c_{SM} \) classes of objects which satisfy the assumptions of Theorem 3.2.1, i.e. where there exists a smooth scheme defined by all but one of the generators. Each of these \( c_{SM} \) classes can then be computed with Algorithm 3.2.1.

In Table 3.1 and Table 3.2 we test Algorithms 3.2.1 and 3.2.2 on a wide selection of complete intersection subschemes of \( \mathbb{P}^n \). We find that Algorithms 3.2.1 and 3.2.2 perform favourably in comparison to other algorithms which compute \( c_{SM}(V) \) class on many applicable examples, with the largest speed up happening when the majority of the generators of the ideal defining the scheme \( V \) are smooth. We also note that the speed up over our inclusion/exclusion based algorithm is quite significant in some cases. If, however, many of the generators define a singular scheme then Algorithms 3.2.1 does not necessarily offer improved performance in comparison to inclusion/exclusion as the cost of computing the singularity subschemes and their Segre classes can become too large. All things considered we believe that Algorithms 3.2.1 and 3.2.2 effectively complement existing algorithms by making \( c_{SM} \) calculation for certain classes of examples much more computationally accessible then it would otherwise be.
1.3.2 Characteristics Class Computations in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$

The main results in Chapter 4 are Theorem 4.2.1 and Theorem 4.2.2. Theorem 4.2.2 gives a method to compute the so-called projective multi-degrees, i.e. the analogue of the projective degrees of (1.7) in multi-projective space (see (4.9)). Theorem 4.2.1 generalizes a result of Aluffi [2] and gives an expression for the Segre class in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ in terms of Chow ring classes which can be computed directly from the projective multi-degrees of (4.9), these can in turn be found using Theorem 4.2.2.

**Projective Multi-degrees**

Theorem 4.2.2 generalizes the result of Theorem 2.3.1; we summerize this result here. Recall that the Chow ring of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ may be expressed as

$$A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \cong \mathbb{Z}[h_1, \ldots, h_m]/(h_1^{n_1+1}, \ldots, h_m^{n_m+1}).$$

Let $R$ be the coordinate ring of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$, let $I = (f_0, \ldots, f_r)$ be a multi-homogeneous ideal in $R$ defining a subscheme $V = V(I)$ of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ and let $n = n_1 + \cdots + n_m$. Assume, without loss of generality, that all generators of $I$ have the same multidegree, that is assume that $\text{deg}(f_i) = (d_1, \ldots, d_m)$ for all $i$. Define a rational map $\phi : \mathbb{P} \to \mathbb{P}^r$ given by

$$\phi : p \mapsto (f_0(p) : \cdots : f_r(p)). \quad (1.12)$$

Let

$$G = \sum_{i=0}^{\text{codim}(V)-1} (d_1 h_1 + \cdots + d_m h_m)^i + \sum_{i=\text{codim}(V)}^{n} [Y_i] \in A^*(\mathbb{P}), \quad (1.13)$$

where

$$[Y_i] = [V(P_1 + \cdots + P_i) - V(I)] \quad (1.14)$$
with the $P_i$ being general linear combinations of $(f_0, \ldots, f_r)$. Note that $[Y_i]$ has pure codimension $\iota$, hence the class $[Y_i] \in A^\iota(\mathbb{P})$ will have the form

$$[Y_i] = \sum_{i_1 = 0}^{i_1 + \cdots + i_m = \iota} \gamma_{(i_1, \ldots, i_m)} h_1^{i_1} \cdots h_m^{i_m}. \quad (1.15)$$

We will refer to the $\gamma_{(i_1, \ldots, i_m)}$ as the *projective multi-degrees* of the rational map $\phi$.

In Theorem 4.2.2 we show that we may compute the projective multi-degrees $\gamma_{(i_1, \ldots, i_m)}$ by computing the vector space dimensions

$$\gamma_{(i_1, \ldots, i_m)} = \dim_k \left( R[T]/(P_1 + \cdots + P_i + \sum_{l=0}^r \omega_l f_l + L_{(a_1, \ldots, a_m)} + L_A + S) \right), \quad (1.16)$$

for $\iota = \text{codim}(V), \ldots, n$ where:

- $P_1, \ldots, P_i$ are ideals defined by general linear combinations of the generators of $I$, i.e.

$$P_j = \left( \sum_{l=0}^r \lambda_{j,f_l} \right).$$

- $S$ is an ideal given by

$$S = \left( 1 - T \sum_{l=0}^r \theta_l f_l \right),$$

where $\sum_{l=0}^r \theta_l f_l$ is a general linear combination of $f_0, \ldots, f_r$.

- $L_{(a_1, \ldots, a_m)}$ is an ideal generated by $a_1$ general homogeneous linear forms of multi-degree $(1, 0, 0, \ldots, 0)$, $a_2$ general homogeneous linear forms of multi-degree $(0, 1, 0, \ldots, 0)$, and so on.

- $L_A$ is the ideal generated by the $m$ affine linear forms

$$L_A = (1 - \ell_{(1,0,0,\ldots,0)}, 1 - \ell_{(0,1,0,\ldots,0)}, \ldots, 1 - \ell_{(0,0,0,\ldots,1)}),$$

where $\ell_{(0,0,\ldots,1,\ldots,0)}$ is a homogeneous linear form having multi-degree $(0, 0, \ldots, 1, \ldots, 0)$. 
Segre Classes

Let $V = V(f_0, \ldots, f_r)$ be a subscheme of $\mathbb{P} = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$. In Theorem 4.2.1 we prove a result which gives an expression for the Segre class $s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})$ in terms of the projective multi-degrees (1.15). Using this result and the result of Theorem 4.2.2 we construct Algorithm 4.3.1 which computes the Segre class $s(V, \mathbb{P})$ by constructing the classes $[Y_i]$ as in (1.15). The main computational steps of Algorithm 4.3.1 are the calculations of the vector space dimensions to find the projective multi-degrees $\gamma(i_1, \ldots, i_m)$ in (1.16).

In Table 4.1 we compare the run time of our new algorithm to compute the Segre class of a subscheme of multi-projective space to the algorithm of Moe and Qviller [27], which is also capable of computing Segre classes in this setting, for a variety of examples. We find that in all cases our method of Algorithm 4.3.1 offers superior run time performance and that for the majority of the examples the difference in performance is considerable. We note that the algorithm of Moe and Qviller [27] works in a more general setting (subschemes of a smooth projective toric variety) and does not attempt to take advantage of the special structures of the Chow rings associated to any particular case.

We give a running time bound for our algorithm to compute Segre classes of subschemes of multi-projective space (Algorithm 4.3.1) in Proposition 4.4.1.

Chern-Schwartz-MacPherson Classes

We give two algorithms to compute the class $c_{SM}(V)$ for $V$ a subscheme of multi-projective space, Algorithm 4.3.2 and Algorithm 4.3.3. Algorithm 4.3.2 generalizes Algorithm 2.3.3 and computes the $c_{SM}$ class using inclusion/exclusion and Aluffi’s [1, Theorem I.4] formula (see Proposition 4.1.2 below) which expresses the $c_{SM}$ class of a hypersurface in terms of the Segre class of the singularity subscheme (i.e. the subscheme of $V$ defined by the vanishing of the partial derivatives of the equation defining the hypersurface). This will allow us to construct Algorithm 4.3.2.
using Algorithm 4.3.1 to compute the Segre class of the singularity subscheme and using the inclusion/exclusion property of $c_{SM}$ classes in higher codimension.

We give running time bounds for Algorithm 4.3.2 in Corollary 4.4.2. In Table 4.2 we give the running time of our algorithm on several examples. At present there are no other existing algorithms known to us for computing Chern-Schwartz-MacPherson classes in the multi-projective setting, hence we are unable to compare these running times to those of another existing algorithm.

In Algorithm 4.3.3 we generalize Algorithm 3.2.1 to the multi-projective setting. We proceed similarly to the construction for projective space given in Chapter 3, namely we prove Theorem 4.2.3 which gives an expression for the $c_{SM}$ class of a complete intersection $V = V(f_0,\ldots,f_r) \subset \mathbb{P}^{m_1} \times \cdots \times \mathbb{P}^{m_n}$ where $V(f_0,\ldots,f_{r-1})$ is a smooth scheme (for some ordering) in terms of the Segre class $s(Y,\mathbb{P}^{m_1} \times \cdots \times \mathbb{P}^{m_n})$ of the singularity subscheme $Y$ of $V$. As in the projective case to prove Theorem 4.2.3 we apply a result of Fullwood [14] which gives an expression for the Milnor class in this setting. Theorem 4.2.3 generalizes the result of Theorem 3.2.1 to the multi-projective setting.

Hence Algorithm 4.3.3 computes the $c_{SM}$ class of a complete intersection satisfying the assumptions of Theorem 4.2.3 without the need for inclusion/exclusion. This algorithm can also be extended to any complete intersection by doing a partial inclusion/exclusion in a manner similar to that of Algorithm 3.2.2. Specifically for $Z$ a smooth subscheme of $\mathbb{P}^{m_1} \times \cdots \times \mathbb{P}^{m_n}$ and for $V_1, V_2$ arbitrary subschemes of $\mathbb{P}^{m_1} \times \cdots \times \mathbb{P}^{m_n}$ we have

$$c_{SM}(Z \cap V_1 \cap V_2) = c_{SM}(Z \cap V_1) + c_{SM}(Z \cap V_2) - c_{SM}(Z \cap (V_1 \cup V_2))$$

note that all expressions on the left hand side of the equation above satisfy the assumptions of Theorem 4.2.3.

In Table 4.3 we compare the run times of Algorithm 4.3.3 to those of Algorithm 4.3.2, our algorithm to compute the $c_{SM}$ class using inclusion/exclusion. We find that, at least for the applicable examples considered in the table, the direct method
of Algorithm 4.3.3 does indeed offer a performance improvement over inclusion/exclusion. While the set of applicable examples is slightly restricted we believe that Algorithm 4.3.3 still provides a useful complement to the more general inclusion/exclusion method of Algorithm 4.3.2.

Note that both these methods to compute $c_{SM}(V)$ for $V$ a subscheme of $[\mathbb{P}^m_1 \times \cdots \times \mathbb{P}^m_n]$ also allow us to immediately obtain the Euler characteristic $\chi(V)$ from the class $c_{SM}(V)$ since

$$\chi(V) = \int c_{SM}(V).$$

1.3.3 Computing the Chern-Schwartz-MacPherson Class and Euler Characteristic of Complete Simplicial Toric Varieties

For a complete simplicial toric variety $X_\Sigma$ defined by a fan $\Sigma$, the class $c_{SM}(X_\Sigma)$ will be considered as a class in the rational Chow ring $A^*(X_\Sigma)_\mathbb{Q}$ of $X_\Sigma$. A definition of this Chow ring (which is well suited to computation) is given in §5.2.

We now give an example using Algorithm 5.3.1 and Algorithm 5.3.2 to compute $c_{SM}(\mathbb{P}^3) = c(T_{\mathbb{P}^3}) \cdot [\mathbb{P}^3]$ in the Chow ring of $\mathbb{P}^3$ and to compute $\chi(\mathbb{P}^3)$. We note that this Chern class and Euler characteristic are, of course, well known and these algorithms are not required for this computation. Rather this example is chosen to illustrate the algorithms in a simple way. An example with a singular toric variety is given as Example 5.3.4 in Chapter 5. For definitions of terms used in this example see §5.1.

**Example 1.3.4.** To define $\mathbb{P}^3 = X_\Sigma$ as the toric variety of a fan we let $\Sigma$ be the fan
defined by the cones

\[ \sigma_0 = \rho_0 + \rho_1 + \rho_2 \]
\[ \sigma_1 = \rho_0 + \rho_1 + \rho_3 \]
\[ \sigma_2 = \rho_0 + \rho_2 + \rho_3 \]
\[ \sigma_3 = \rho_1 + \rho_2 + \rho_3 \]

and their faces where \( \rho_0 = \langle (1,0,0) \rangle, \rho_1 = \langle (0,1,0) \rangle, \rho_2 = \langle (0,0,1) \rangle, \rho_3 = \langle (-1,-1,-1) \rangle \). We may refer to \( \rho_0, \rho_1, \rho_2, \rho_3 \) as the generating rays \( \Sigma(1) \) of \( \Sigma \).

The Chow ring of \( \mathbb{P}^3 \) as a toric variety has presentation

\[ A^*(\mathbb{P}^3) \cong \mathbb{Z}[x_0, x_1, x_2, x_3]/(x_0 x_1 x_2 x_3, x_1 - x_0, x_2 - x_0, x_3 - x_0), \]

note that the above toric presentation is isomorphic to the usual presentation \( A^*(\mathbb{P}^3) \cong \mathbb{Z}[h]/(h^4) \), however for this example we will use the toric presentation as in Algorithm 5.3.1. Note that since \( \mathbb{P}^3 \) is smooth we will have \( \text{mult}(\sigma) = 1 \) for all cones \( \sigma \in \Sigma \), see Lemma 5.3.2.

Using Algorithm 5.3.1 we have that the codimension one part of \( c_{SM}(\mathbb{P}^3) \) is

\[ (c_{SM}(\mathbb{P}^3))^{(1)} = \text{mult}(\rho_0)[V(\rho_0)] + \text{mult}(\rho_1)[V(\rho_1)] + \text{mult}(\rho_2)[V(\rho_2)] + \text{mult}(\rho_3)[V(\rho_3)] \in A^*(\mathbb{P}^3) \]
\[ = x_0 + x_1 + x_2 + x_3 \]
\[ = 4x_3. \]
The codimension two part of $c_{SM}(\mathbb{P}^3)$ is

$$(c_{SM}(\mathbb{P}^3))^{(2)} = \text{mult}(\rho_0 + \rho_1)[V(\rho_0 + \rho_1)] + \text{mult}(\rho_0 + \rho_2)[V(\rho_0 + \rho_2)]$$
$$+ \text{mult}(\rho_0 + \rho_3)[V(\rho_0 + \rho_3)] + \text{mult}(\rho_1 + \rho_2)[V(\rho_1 + \rho_2)]$$
$$+ \text{mult}(\rho_1 + \rho_3)[V(\rho_1 + \rho_3)] + \text{mult}(\rho_2 + \rho_3)[V(\rho_2 + \rho_3)] \in A^*(\mathbb{P}^3)$$

$$= x_0x_1 + x_0x_2 + x_0x_3 + x_1x_2 + x_1x_3 + x_2x_3$$
$$= 6x_3^2.$$

The codimension three part of $c_{SM}(\mathbb{P}^3)$ is

$$(c_{SM}(\mathbb{P}^3))^{(3)} = \text{mult}(\rho_0 + \rho_1 + \rho_2)[V(\rho_0 + \rho_1 + \rho_2)]$$
$$+ \text{mult}(\rho_0 + \rho_1 + \rho_3)[V(\rho_0 + \rho_1 + \rho_3)]$$
$$+ \text{mult}(\rho_0 + \rho_2 + \rho_3)[V(\rho_0 + \rho_2 + \rho_3)]$$
$$+ \text{mult}(\rho_1 + \rho_2 + \rho_3)[V(\rho_1 + \rho_2 + \rho_3)] \in A^*(\mathbb{P}^3)$$

$$= x_0x_1x_2 + x_0x_1x_3 + x_0x_2x_3 + x_1x_2x_3$$
$$= 4x_3^3.$$

Finally we note the the codimension zero part of $c_{SM}(\mathbb{P}^3)$ is $1 \in A^*(\mathbb{P}^3)$, i.e. the class of the orbit closure of the zero cone $\langle (0,0,0) \rangle$ is $[V(\langle (0,0,0) \rangle)] = 1$. Hence Algorithm 5.3.1 gives us

$$c_{SM}(\mathbb{P}^3) = 4x_3^3 + 6x_3^2 + 4x_3 + 1 \in A^*(\mathbb{P}^3) \cong \mathbb{Z}[x_0,x_1,x_2,x_3] / \langle x_0x_1x_2x_3, x_1 - x_0, x_2 - x_0, x_3 - x_0 \rangle,$$

the last step of the algorithm is to find a basis for the dimension zero Chow group $A_0(\mathbb{P}^3)$ and compute the Euler characteristic. In this case $\{x_3^3\}$ forms a basis of $A_0(\mathbb{P}^3)$, hence the Euler characteristic is the coefficient of $x_3^3$ in $c_{SM}(\mathbb{P}^3)$, that is

$$\chi(\mathbb{P}^3) = \int c_{SM}(\mathbb{P}^3) = 4.$$
would perform only the computation of the codimension three piece of the $c_{SM}$ class, that is the computation of $(c_{SM}(\mathbb{P}^3))^{(3)}$ above. From this, Algorithm 5.3.2 obtains the Euler characteristic directly by summing the coefficients of the monomials in the polynomial expression of $(c_{SM}(\mathbb{P}^3))^{(3)}$ above, this gives $\chi(\mathbb{P}^3) = 4$.

### 1.3.4 Bézout Type Results in Multi-projective Space

In this subsection we will focus only on motivating the Bézout-like bounds of Chapter 6 by considering an example application of the results proved in Chapter 6 to a problem considered by Safey El Din and Trebuchet in [13].

In the following we will frequently make use of the notion of geometric multiplicity in the manner of Fulton [16, §1.5] and Fulton [15, §2.1]; we briefly describe this notion here. Let $k$ be an algebraically closed field of characteristic zero. Let $V$ be a subvariety (or subscheme) of $k^n = \text{Spec}(k[x_1, \ldots, x_n])$, i.e. a subvariety of dimension $n$ affine space with coordinate ring $k[x_1, \ldots, x_n]$. Let $W$ be an irreducible component of $V$. In this case the local ring $O_{W,V}$ is given by the localization of the coordinate ring of $V$ at the prime ideal $I(W)$, that is

$$O_{W,V} = (k[x_1, \ldots, x_n]/I(V))_{I(W)},$$

see [15, §2.1] or the proof of Lemma 6.3.3, and see §6.1 for the definition of $O_{W,V}$ in a more general setting.

We will write $\ell(O_{W,V}) = \ell_{O_{W,V}}(O_{W,V})$ for the geometric multiplicity of $W$ in $V$ where $\ell(O_{W,V})$ is the length of $O_{W,V}$ as an $O_{W,V}$-module. Recall that a module $M$ has length $n$ if there is a composition series $M_0 = M \supsetneq M_1 \supsetneq \cdots \supsetneq M_n = \{0\}$ and this is the shortest such series. For a set of points in affine space the notion of geometric multiplicity defined above reduces to the usual notion of the multiplicity of a point as we will see in the Example 1.3.5.

#### Example 1.3.5

By Fulton [15, §2.1] when we consider a dimension zero variety
\[ V = V(f_1, \ldots, f_n) \text{ in } k^n \text{ and } W = (a_1, \ldots, a_n) \text{ an isolated point in } V \text{ we have} \]

\[ \ell(O_{W,V}) = \dim_k ((k[x_1, \ldots, x_n]/(f_1, \ldots, f_n))_P), \quad (1.17) \]

where \( P = (x_1 - a_1, \ldots, x_n - a_n) \) is the prime ideal of the point \( W \).

Consider the intersection of two curves \( f, g \) in \( \mathbb{C}^2 = \text{Spec}(\mathbb{C}[x,y]) \), if \( p = (a, b) \) is an isolated point in the intersection \( V = V(f, g) \) then

\[ \ell(O_{p,V}) = \dim_{\mathbb{C}} \left( (\mathbb{C}[x,y]/(f,g))_{(x-a,y-b)} \right). \]

If we take \( V \) to be the intersection of the curves \( y = x^2 \) and the x-axis \( y = 0 \) we have one isolated point at \( p = (0, 0) \) with geometric multiplicity

\[ \ell(O_{p,V}) = \dim_{\mathbb{C}} \left( (\mathbb{C}[x,y]/(x^2 - y,y))_{(x,y)} \right) \]
\[ = \dim_{\mathbb{C}} \left( \mathbb{C}[x,y]_{(x,y)}/(x^2 - y,y) \right) \]
\[ = \dim_{\mathbb{C}} \left( \mathbb{C}[[x,y]]/(x^2 - y,y) \right) \]
\[ = 2. \]

Here the basis of \( \mathbb{C}[[x,y]]/(x^2 - y,y) \) given by \([1, x]\) where \( \mathbb{C}[[x,y]] \) denotes the ring of formal power series. Note that we may replace the localization \( \mathbb{C}[x,y]_{(x,y)} \) by its completion \( \mathbb{C}[[x,y]] \) in this case, see Fulton [15, §1.6].

The motivating example for the work in Chapter 6 we consider here comes from a problem considered by Safey El Din and Trebuchet in [13] when developing an algorithm to compute at least one point in each connected component of a smooth real algebraic set.

Suppose we have an arbitrary collection of polynomials \( f_1, \ldots, f_m \) in \( k[x_1, \ldots, x_n] \) with \( m < n \) and with \( \deg(f_i) \leq D \) for all \( i \) defining an affine variety in \( \mathbb{A}^n \). Further suppose we wish to find the critical locus of \( V(f_1, \ldots, f_m) \) using the method of Lagrange multipliers. To do this, the system we wish to consider is the following
collection of polynomials in $k[x_1, \ldots, x_n, l_1, \ldots, l_m]$

\[
F_j = \begin{cases} 
  f_j & \text{if } j \leq m \\
  l_1 \frac{\partial f_j}{\partial x_{jm}} + \cdots + l_m \frac{\partial f_j}{\partial x_{jm}} - 1 & \text{if } j = m + 1 \\
  l_1 \frac{\partial f_j}{\partial x_{jm}} + \cdots + l_m \frac{\partial f_j}{\partial x_{jm}} & \text{if } m + 2 \leq j \leq m + n
\end{cases} \tag{1.18}
\]

We may then calculate the critical locus by using an algorithm such as Giusti, Lecerf and Salvy [18] (or Lecerf [24]) to compute the variety $V = V(F_1, \ldots, F_{n+m})$. Let $W_1, \ldots, W_t$ be the irreducible components of $V$. The algorithms of Giusti, Lecerf and Salvy [18] and of Lecerf [24] have known running time bounds that depend on the sum of the degrees of the $W_i$ weighted by multiplicity, that is the running time bounds depend on the quantity $\delta$ given by

\[
\delta = \sum \ell(O_{W_i,V}) \deg(W_i).
\]

Hence to give refined running time bounds on the time to compute the critical locus of $V(f_1, \ldots, f_m)$ using the method of Lagrange multipliers the problem we wish to consider is the following. Letting $V \subset A^{n+m}$ be the affine variety defined by $V = V(F_1, \ldots, F_{n+m})$, how do we provide a refined bound on the degrees of the irreducible components $W_1, \ldots, W_j$ of $V$ with multiplicity, i.e. a bound which is sharper than the usual Bézout bound in this case? More specifically, if we homogenize to obtain the projective closure $\overline{V} \subset P^{n+m}$ we could then apply the usual Bézout bound in $P^{n+m}$ to obtain

\[
\delta = \sum_{i=1}^{j} \ell(O_{W_i,V}) \deg(W_i) \leq D^{n+m}. \tag{1.19}
\]

Our goal is to obtain a sharper bound than this by making use of the natural bi-projective structure of the variety associated to the system of polynomials in (1.18). In fact from Corollary 6.3.8 we have the following bound

\[
\delta = \sum_{i=1}^{j} \ell(O_{W_i,V}) \deg(W_i) \leq \binom{n + m - 1}{n - 1} D^p,
\]
Corollary 6.3.8 follows from Theorem 6.2.1 which we prove in Chapter 6. Additionally if $V(F_1, \ldots, F_m)$ is a complete intersection, Corollary 6.3.9 gives us the slightly sharper bound

$$\delta = \sum_{i=1}^{j} \ell(O_{W_i,V})\deg(W_i) \leq \binom{n}{n-m} D^m(D-1)^{n-m}.$$

We note that the bounds obtained from Corollary 6.3.8 and Corollary 6.3.9 are sharper (at least for large degree) than the bound obtained from the standard projective Bézout bound given in (1.19).
Bibliography


[24] Grégoire Lecerf. Computing the equidimensional decomposition of an alge-


Chapter 2

Computing Characteristics Classes in Projective Space

The method to compute Chern-Schwartz-MacPherson classes described here is based on several known formulas due to Aluffi [1, 2], and on the notion of the projective degrees of a rational map as expressed in Harris [14]. The main result of this chapter is Theorem 2.3.1 which gives a method to compute projective degrees.

In particular, in this chapter, given the ideal \( I \) defining a projective variety \( V \) in \( \mathbb{P}^n \) we will compute the pushforward to \( \mathbb{P}^n \) of both the Segre class of \( V \) in \( \mathbb{P}^n \) and the Chern-Schwartz-MacPherson class of \( V \) (we abuse notation and denote the pushforwards to \( \mathbb{P}^n \) as \( s(V, \mathbb{P}^n) \) and \( c_{SM}(V) \) respectively). From \( c_{SM}(V) \) we may immediately obtain the Euler characteristic of \( V \), \( \chi(V) \) using the well-known relation which states that \( \chi(V) \) is equal to the degree of the zero dimensional component of \( c_{SM}(V) \). The algorithm described may be implemented either symbolically, with the computations relying on Gröbner bases calculations, or numerically using homotopy continuation.

We now give an example of the computation of the Segre class, the \( c_{SM} \) class and the Euler characteristic for a singular projective variety. Note that since the variety \( V \) considered in the example is singular the results \( c_{SM}(V) \) and \( \chi(V) \) could not be obtained with standard Chern class computations.
Example 2.0.6. Let $V = V(I)$ be the subvariety of $\mathbb{P}^4$ defined by the ideal $I = (4x_3x_2x_4x_1 - x_0^3x_1, x_0x_1x_3x_4 - x_2^3x_3)$ in $k[x_0, x_1, x_2, x_3, x_4]$. Also let $A_*(\mathbb{P}^4) \cong \mathbb{Z}[h]/(h^5)$ be the Chow ring of $\mathbb{P}^4$.

Using Algorithm 2.3.2 with input $I$ we obtain the Segre class

$$s(V, \mathbb{P}^4) = 768h^4 - 128h^3 + 16h^2 \in A_*(\mathbb{P}^4).$$

Using Algorithm 2.3.3 with input $I$ we obtain the Chern-Schwartz-MacPherson class

$$c_{SM}(V) = 5h^4 + 8h^3 + 12h^2 \in A_*(\mathbb{P}^4)$$

and/or the Euler characteristic $\chi(V) = 5$.

The organization of the remainder of chapter is as follows. In Section 2.1 we state the problem we wish to consider and review the general definitions of the $c_{SM}$ class and the Chow ring. We also give several important results relating to the computation of $c_{SM}$ classes.

In Section 2.2 we briefly review relevant background on the projective degrees of a rational map and state known formulas which expresses the Segre class and $c_{SM}$ class in terms of these projective degrees. Also in Section 2.2 we review previous algorithms for the computation of Segre and $c_{SM}$ classes. Specifically we review algorithms of Aluffi [2] and Eklund, Jost and Peterson [8] for the computation of Segre classes and we review algorithms of Aluffi [2] and Jost [17] for the computation of $c_{SM}$ classes. Additionally we explain the relationship between the residual degrees computed by Eklund, Jost and Peterson in [8] and the projective degrees in (2.16). In light of this relationship one could see Algorithm 2.3.2 and Algorithm 2.3.3 as refinements of the algorithms of [8] and [17] respectively; however we note that these methods are developed using very different theoretical tools, and a priori there is no obvious relationship between them.

In Section 2.3 we state and prove Theorem 2.3.1 which is the main result of this chapter and which gives a new formula for calculating the projective degrees of a rational map defined by a homogeneous ideal. In Algorithm 2.3.1 we show how
the result of Theorem 2.3.1 can be used to compute the projective degrees of a rational map using a computer algebra system. We then apply Algorithm 2.3.1 to give Algorithm 2.3.2 which computes the Segre class and Algorithm 2.3.3 which computes the $c_{SM}$ class.

In Section 2.4 we discuss the performance of our algorithm to compute Segre classes (Algorithm 2.3.2) and our algorithm to compute the $c_{SM}$ class (Algorithm 2.3.3). Run time performance for Algorithm 2.3.2 is compared with previous algorithms of Aluffi [2] and of Eklund, Jost and Peterson [8] which also compute Segre classes. The results of the running time comparisons for Segre classes are summarized in Table 2.1; we see that our algorithm performs favourably in most cases. Run time performance for Algorithm 2.3.3 is compared with previous algorithms of Aluffi [2] and of Jost [17] which also compute the $c_{SM}$ class and/or the Euler characteristic. We also compare the Macaulay2 [13] implementation of Algorithm 2.3.3 to the Macaulay2 built in routine “euler” which calculates Hodge numbers to compute the Euler characteristic. In all cases Algorithm 2.3.3 performs favourably in comparison to other known algorithms. The results are summarized in Table 2.2.

The Macaulay2 [13] and Sage [24] implementations of our algorithm for computing $c_{SM}$ classes, Euler characteristics and Segre classes of projective varieties can be found at https://github.com/Martin-Helmer/char-class-calc, see Appendix A.1 for a description of the syntax used by our “CharClassCalc” Macaulay2 package. The Macaulay2 [13] implementation is also available as part of the “CharacteristicClasses” package in Macaulay2 version 1.7 and above and can be accessed using the option “Algorithm=>ProjectiveDegree”, see the Macaulay2 documentation http://www.math.uiuc.edu/Macaulay2/doc/Macaulay2-1.7/share/doc/Macaulay2/CharacteristicClasses/html/ for further details.
2.1 Problem and Setting

Suppose $V$ is an arbitrary subscheme of a projective space $\mathbb{P}^n$ over an algebraically closed field of characteristic zero. The problem we wish to consider is that of devising an effective and practical algorithmic method to compute the Segre class $s(V, \mathbb{P}^n)$ and the Chern-Schwartz-MacPherson class $c_{SM}(V)$ as elements of the Chow ring of $\mathbb{P}^n$. An algorithm which computes $c_{SM}(V)$ automatically gives us the Euler characteristic $\chi(V)$, since this information is contained directly in $c_{SM}(V)$.

In this section we review the definition of Chow groups and Chow rings; this is important as we will represent characteristic classes as elements of some Chow ring. We also define the Chern-Schwartz-MacPherson class in a general setting and discuss some important results relating to the computation of the $c_{SM}$ class. A general definition of the Segre class is given in §1.2.2 in the introduction, specifically see (1.3) above.

2.1.1 Chow Groups and Chow Rings

When working with Chow groups and Chow rings by variety we will mean a reduced and irreducible scheme. A subvariety of a scheme will be taken to mean a reduced and irreducible subscheme.

Let $Y$ be a scheme of finite type over a ground field, (for example $Y$ could be a variety), we may define the group of cycles on $Y$, $Z(Y)$, as the free abelian group generated by set of irreducible subvarieties of $Y$. This group is graded by dimension with $Z_j(Y)$ denoting the group of $j$-cycles, that is the group of cycles which are finite formal linear combinations of varieties of dimension $j$, we can write this as

$$Z_j(Y) = \left\{ \sum n_i[V_i] \mid n_i \in \mathbb{Z}, \text{ } V_i \text{ is a } j \text{ dimensional subvariety of } Y \right\}.$$
So we have that

\[ Z(Y) = \bigoplus_{j} Z_j(Y). \]

**Chow Groups**

The Chow group will be given by the group of cycles modulo rational equivalence. Informally we say two cycles \( \alpha, \beta \in Z(Y) \) are rationally equivalent if there exists a “family” of cycles specified by a rational parametrization which interpolates between \( \alpha \) and \( \beta \). More explicitly we define a map \( \delta_Y : Z(Y \times \mathbb{P}^1) \to Z(Y) \) on free generators as follows. Let \( W \) be a subvariety of \( Y \times \mathbb{P}^1 \). If the projection onto the second factor \( \pi : W \to \mathbb{P}^1 \) is not dominant, i.e. if \( W \subset Y \times \{t\} \) for some \( t \in \mathbb{P}^1 \), then we set \( \delta_Y(W) = 0 \). If, on the other hand, the projection \( W \to \mathbb{P}^1 \) is dominant then we let \( W_0 = \pi^{-1}(0) \subset Y \times \{0\} = Y \) and \( W_\infty = \pi^{-1}(\infty) \subset Y \times \{\infty\} = Y \), where \( 0 = (0 : 1) \) and \( \infty = (1 : 0) \) are the usual zero and infinity points of \( \mathbb{P}^1 \). In this case we define \( \delta_Y(W) = [W_0] - [W_\infty] \).

We write \( \text{Rat}(Y) \subset Z(Y) \) for the image \( \delta_Y(Z(Y \times \mathbb{P}^1)) \), that is the subgroup generated by all cycles of the form \([W_0] - [W_\infty]\). Two cycles \( \alpha, \beta \in Z(Y) \) are defined to be rationally equivalent if \( \alpha - \beta \in \text{Rat}(Y) \), and \( \text{Rat}_j(Y) \) is the group of \( j \)-cycles rationally equivalent to zero. The Chow group \( A(Y) \) is the group of rational equivalence classes,

\[ A(Y) := Z(Y)/\text{Rat}(Y) \]

\[ = \text{coker}(\delta_Y). \]

See Chapter 4 of Eisenbid and Harris [7] and Section 1.6 of Fulton [10] for further details.

The quotient group \( A_j(X) = Z_j(X)/B_j(X) \) is the Chow group of dimension \( k \). The quotient group \( A^j(X) = Z^j(X)/B^j(X) \) is the Chow group of codimension \( j \) where \( Z^j(X) \) is the group of codimension \( j \) cycles and \( B^j(X) \) is the associated group of codimension \( j \) cycles rationally equivalent to zero. See Chapter 1 of Fulton [10].
Chapter 9 of Gathmann [11], or §1.1 of Eisenbud and Harris [7] for a complete description.

Chow Rings

In the case where \( Y \) is a smooth variety of dimension \( n \) the Chow groups of \( Y \), \( A_j(Y) \), form a graded ring

\[
A_*(Y) = \bigoplus_{j=0}^{n} A_j(Y),
\]

(2.1)

this ring is graded by dimension. We may also form a ring \( A^*(Y) = A_*(Y) \) graded by codimension from the Chow groups, that is

\[
A^*(Y) = \bigoplus_{j=0}^{n} A^j(Y).
\]

(2.2)

Multiplication on the Chow ring \( A^*(Y) = A_*(Y) \) is given by the intersection product (2.3), we describe this multiplication below.

Let \( V \) be a subscheme of \( Y \) having pure codimension \( d \), and let \( W \) be a purely \( j \) dimensional subscheme of \( Y \). Also let \( T \) denote the tangent bundle of \( Y \), \( T_Y \), restricted to \( V \cap W \), and let \( c(T) \) denote the total Chern class of the vector bundle \( T \). We may define the intersection product as

\[
[V] \cdot [W] = \{c(T) \cdot s(V \cap W, V \times W)\}_{j-d} \in A_{j-d}(V \cap W) \subset A_{j-d}(Y).
\]

(2.3)

Here we consider \( V \cap W \) as a subvariety of \( V \times W \) via the diagonal embedding of \( Y \) in \( Y \times Y \). Note that the expression \( c(T) \cdot s(V \cap W, V \times W) \) denotes the homomorphism specified by the Chern class \( c(T) \) acting on \( s(V \cap W, V \times W) \) in the manner of Fulton [10, Chapter 3]. This product makes \( A_*(Y) \) (and \( A^*(Y) \)) into a commutative graded ring with unit \( [Y] \). In what follows we will most frequently use the notation \( A^*(Y) \) for the Chow ring, i.e. we will use the codimension grading.

**Example 2.1.1 (Ex. 8.1.11 [10]).** Let \( V, W \) be subvarieties of a non-singular variety \( Y \). If \( V \) and \( W \) are non-singular varieties which meet transversely at generic points
of $V \cap W$ we have
\[
[V] \cdot [W] = [V \cap W]. \tag{2.4}
\]
More generally the equality (2.4) holds if the diagonal embedding of the intersection scheme $V \cap W$ in $V \times W$ is a regular embedding of codimension $\dim Y$.

Degree of a Chow Ring Element

Take $M$ to be a smooth (irreducible) variety over an algebraically closed field and let $\alpha = \sum_{V} n_{V}[V]$ be an arbitrary element of $A^*(M)$. We will refer to $\int \alpha$ as the degree of the zero dimensional part of $\alpha$, that is
\[
\int \alpha = \sum_{[V] \in A_0(M)} n_{V} = \sum_{\dim(V) = 0} n_{V}. \tag{2.5}
\]
Put another way, $\int \alpha$ denotes the sum of the integer coefficients of the classes of dimension zero irreducible varieties in $\alpha$, that is the coefficients of the pieces of $\alpha$ which are in the dimension zero Chow group $A_0(M)$.

Chow Ring of $\mathbb{P}^n$

In this chapter (and in Chapter 3) we will work only in the Chow ring of $\mathbb{P}^n$, $A^*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1})$, where $h = c_1(O_{\mathbb{P}^n}(1))$ is the equivalence class of a hyperplane in $\mathbb{P}^n$, hence a hypersurface $W$ of degree $d$ in $\mathbb{P}^n$ is represented as $[W] = d \cdot h$ in $A^*(\mathbb{P}^n)$ (for more details see Fulton [10]). Here $c_1$ denotes the first Chern class of a line bundle, see Fulton [10, §2.5].

For an element $\alpha \in A^*(\mathbb{P}^n)$ we have that $\int \alpha$ will be the integer coefficient of $h^n$ in $\alpha$ (which can be zero). For $V$ a subscheme of pure dimension $j$ in $\mathbb{P}^n$ we will write
\[
\deg([V]) = \int c_1(O_{\mathbb{P}^n}(1))^{j}[V] = \int h^{j}[V],
\]
that is $\deg([V])$ is the coefficient $\zeta \in \mathbb{Z}$ of $h^{n-j}$ in $[V]$, note that the term $\zeta h^{n-j}$ is

\[
A \ version \ of \ this \ chapter \ has \ been \ published \ in \ The \ Journal \ of \ Symbolic \ Computation, \ see \ Helmer [16].
\]
in the dimension $j$ Chow group $A_j(\mathbb{P}^n)$. Also note that $\deg([V]) = \deg(V)$ where $\deg(V)$ is the usual geometric notion of degree, i.e. $\deg(V)$ denotes the number of points in the intersection of $V$ with $j$ general hyperplanes.

Finally we note that in practice we will always use the presentation $A^*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1})$ and hence the Segre class and $c_{S,M}$ class will be represented as a polynomial with integer coefficients in $\mathbb{Z}[h]/(h^{n+1})$.

### 2.1.2 Chern and Chern-Schwartz-MacPherson Classes

The total Chern class of a $j$-dimensional nonsingular variety $V$ is defined as the Chern class of the tangent bundle $T_V$, we write this as $c(V) = c(T_V) \cdot [V]$ in the Chow ring of $V$, $A_*(V)$. See Fulton [10, §3.2] for a definition of the Chern class of a vector bundle. In this chapter and in Chapter 3 we will abuse notation and write $c(V)$ for the pushforward to $\mathbb{P}^n$ of the total Chern class of $V$ (as we also do with $c_{S,M}$ and Segre classes). As a consequence of the Gauss-Bonnet-Chern theorem (or the Grothendieck-Riemann-Roch theorem, see for example Schürmann and Yokura [22]), we have that the degree of the zero dimensional component of the total Chern class of a projective variety is equal to the Euler characteristic, that is

$$
\int c(T_V) \cdot [V] = \chi(V).
$$

There are several known generalizations of the total Chern class to singular varieties. All of these notions agree with $c(T_V) \cdot [V]$ for nonsingular $V$, however the Chern-Schwartz-Macpherson class is the only one of these that satisfies a property analogous to (2.6) for any $V$, i.e.

$$
\int c_{S,M}(V) = \chi(V).
$$

We review here the construction of the $c_{S,M}$ classes, given in the manner considered by MacPherson [21]. For a scheme $V$, let $C(V)$ denote the abelian group of finite
linear combinations $\sum W m_W 1_W$, where $W$ are (closed) subvarieties of $V$, $m_W \in \mathbb{Z}$, and $1_W$ denotes the function that is 1 in $W$, and 0 outside of $W$. Elements $f \in C(V)$ are known as constructible functions and the group $C(V)$ is referred to as the group of constructible functions on $V$. To make $C$ into a functor we let $C$ map a scheme $V$ to the group of constructible functions on $V$ and a proper morphism $f : V_1 \to V_2$ is mapped by $C$ to

$$C(f)(1_W)(p) = \chi(f^{-1}(p) \cap W), \quad W \subset V_1, \ p \in V_2 \text{ a closed point.}$$

Another functor from algebraic varieties to abelian groups is the Chow group functor $\mathcal{A}_*$. The $c_{SM}$ class may be realized as a natural transformation between these two functors.

**Definition 2.1.2.** The Chern-Schwartz-MacPherson class is the unique natural transformation between the constructible function functor and the Chow group functor, that is $c_{SM} : C \to \mathcal{A}_*$, is the unique natural transformation satisfying:

- (Normalization) $c_{SM}(1_V) = c(T_V) \cdot [V]$ for $V$ non-singular and complete.
- (Naturality) $f_*(c_{SM}(\phi)) = c_{SM}(C(f)(\phi))$, for $f : X \to Y$ a proper transformation of projective varieties, $\phi$ a constructible function on $X$.

For a scheme $V$ let $V_{\text{red}}$ denote the support of $V$, the notation $c_{SM}(V)$ is taken to mean $c_{SM}(1_V)$ and hence, since $1_V = 1_{V_{\text{red}}}$, we denote $c_{SM}(V) = c_{SM}(V_{\text{red}})$.

To see how the $c_{SM}$ class satisfies the relation (2.7) consider the morphism $\mathfrak{f} : V \to \text{point}$, applying the naturality property of the $c_{SM}$ class we have

$$\mathfrak{f}_*(c_{SM}(V)) = c_{SM}(C(\mathfrak{f})(1_V)) = c_{SM}(\chi(V)1_{\text{point}}) = \chi(V)c_{SM}(\text{point}) = \chi(V)[\text{point}].$$

This gives us (2.7). Note that the $c_{SM}$ classes (and constructible functions) also satisfy the same inclusion/exclusion relation as the Euler characteristic, i.e. for the Euler characteristic we have

$$\chi(V_1 \cup V_2) = \chi(V_1)\chi(V_2) - \chi(V_1 \cap V_2).$$

*A version of this chapter has been published in The Journal of Symbolic Computation, see Helmer [16]*
Constructible functions inherit this property from the Euler characteristic via the definition of the constructible function functor, specifically we have \( \mathbf{1}_{V_1 \cup V_2} = \mathbf{1}_{V_1} + \mathbf{1}_{V_2} - \mathbf{1}_{V_1 \cap V_2} \). From this we see that the \( c_{SM} \) classes will also possess an inclusion/exclusion property, giving us the relation

\[
\operatorname{Recall} \text{that from the construction of the } c_{SM} \text{ class we see that } c_{SM} \text{ classes will also possess an inclusion/exclusion property similar to that of the Euler characteristic, in particular for } V_1, V_2 \text{ subschemes of projective space } \mathbb{P}^n \text{ we have that }
\]

\[
c_{SM}(V_1 \cap V_2) = c_{SM}(V_1) + c_{SM}(V_2) - c_{SM}(V_1 \cup V_2). \tag{2.8}
\]

Note that this relation for \( c_{SM} \) classes will allow us to reduce all computation of \( c_{SM} \) classes to the case of hypersurfaces. From this property we obtain the following proposition, discussed informally by Aluffi [2]; Proposition 2.1.3 follows directly from (2.8).

**Proposition 2.1.3.** Let \( V = X_1 \cap \cdots \cap X_r = V(f_1) \cap \cdots \cap V(f_r) \) be a subscheme of \( \mathbb{P}^n = \text{Proj}(k[x_0, \ldots, x_n]) \). Write the polynomials defining \( V \) as \( F = (f_1, \ldots, f_r) \) and let \( F_{[S]} = \prod_{i \in S} f_i \) for \( S \subset \{1, \ldots, r\} \). Then,

\[
c_{SM}(V) = \sum_{S \subset \{1, \ldots, r\}} (-1)^{|S|+1} c_{SM}(V(F_{[S]}))
\]

where \(|S|\) denotes the cardinality of the integer set \( S \).

**Remark 2.1.4.** The following special case is from Suwa [25]. Let \( X \) be a smooth subvariety of \( \mathbb{P}^n \) which is a global complete intersection, further suppose that \( X = V(f_0, \ldots, f_r) \) with \( d_i = \deg f_i \), then we have

\[
c_{SM}(X) = c(X) = (1 + h)^{n+1} \cdot \prod_{i=0}^{\text{codim} X} \frac{d_i h}{1 + d_i h} \quad \text{in } A_*(\mathbb{P}^n), \tag{2.9}
\]

recall that \( c(X) = c(T_X) \cdot [X] \) is the total Chern class of the smooth variety \( X \).

We note that using Remark 2.1.4 the computation of \( c_{SM} \) classes could be made much more efficient in the particular case where the input scheme is a complete intersection which is known to be smooth.
As noted in Example 1.2.1 above, when working in $\mathbb{P}^n$, there is a very concrete relationship between the $c_{SM}$ class and the Euler characteristic of general linear sections, in particular it was shown by Aluffi [4] that there is an involution between these two objects, we state this result below.

**Theorem 2.1.5** (Theorem 1.1 Aluffi [4]). *Let $V$ be any locally closed set in $\mathbb{P}^n$. Let $V_r = V \cap L_1 \cap \cdots \cap L_r$ be the intersection of $V$ with $r$ general hyperplanes. Define the polynomial having degree at most $n$ specified by*

$$
\chi_V(t) := \sum_{r \geq 0} \chi(V_r) \cdot (-t)^r.
$$

*Define another polynomial of degree at most $n$ given by*

$$
\gamma_V(t) := \sum_{r \geq 0} \gamma_r \cdot (-t)^r
$$

*here $\gamma_r = c_{SM}(V)_r$ is the coefficient of the dimension $r$ component of $c_{SM}(V)$, that is; the polynomial $\gamma_V(t)$ is obtained by replacing $[\mathbb{P}^r] \cong h^{n-r}$ with $t^r$ in $c_{SM}(V)$. Also define the map $I$ specified by*

$$
p(t) \mapsto I(p) := \frac{t \cdot p(-t-1) + p(0)}{t+1}.
$$

*Then $I$ is an involution and we have:*

$$
\chi_V(t) = I(\gamma_V(t)), \quad \gamma_V(t) = I(\chi_V(t)). \quad (2.10)
$$

### 2.2 Review

As in the previous section we consider possibly singular closed subschemes, $V$, of the projective space $\mathbb{P}^n$ over $k$, an algebraically closed field of characteristic zero.
We review the definition of the projective degrees of a rational map in §2.2.1. In §2.2.2 we review a result of Aluffi [2] which gives an explicit expression for the Segre class \( s(V, \mathbb{P}^m) \) in terms of the projective degrees in Proposition 2.2.1. We then discuss previous algorithms to compute the Segre class \( s(V, \mathbb{P}^n) \).

In §2.2.3 we review a result of Aluffi [1] which allows one to compute the Chern-Schwartz-MacPherson class of a hypersurface by computing certain Segre classes, stated in Proposition 2.2.3. We also discuss previous algorithms which use the result stated in Proposition 2.2.3 to calculate \( c_{SM} \) classes. Finally we give a result of Aluffi [2] which gives an expression for the \( c_{SM} \) class of a hypersurface in terms of the projective degrees of a certain rational map in Theorem 2.2.4.

### 2.2.1 Projective Degrees

Here we recall the definition of the projective degrees of a rational map as in Harris [14]; the computation of these projective degrees will allow for the calculation of Segre and \( c_{SM} \) classes using Algorithms 2.3.1, 2.3.2 and 2.3.3.

Consider a rational map \( \phi : \mathbb{P}^n \rightarrow \mathbb{P}^m \). In the manner of Harris (Example 19.4 of [14]) we may define the projective degrees of the map \( \phi \) as a list of integers \( (g_0, \ldots, g_n) \) where

\[
g_i = \text{card} \left( \phi^{-1} \left( \mathbb{P}^{m-i} \cap \mathbb{P}^i \right) \right).
\]  

(2.11)

where \( \mathbb{P}^{m-i} \subset \mathbb{P}^m \) and \( \mathbb{P}^i \subset \mathbb{P}^n \) are general hyperplanes of dimension \( m - i \) and \( i \) respectively and card is the cardinality of a zero dimensional set. Note that points in \( \phi^{-1} \left( \mathbb{P}^{m-i} \cap \mathbb{P}^i \right) \) will have multiplicity one (this follows from the Bertini theorem of Sommese and Wampler [23, §A.8.7]). Let \( \Gamma_\phi \subset \mathbb{P}^n \times \mathbb{P}^m \) be the graph of \( \phi \). The numbers \( g_i \) are also used by Aluffi [1, 2, 4], where the class \([\Gamma_\phi]\) is pushed forward to a class \([G] \in A_*(\mathbb{P}^n)\) by the projection map onto the first factor of \( \mathbb{P}^n \times \mathbb{P}^m \). Aluffi refers to the class \([G]\) as the class of the shadow of the graph of the map \( \phi \). Specifically, take \( t \) to be the pull-back of the hyperplane class from the \( \mathbb{P}^m \) factor of \( \mathbb{P}^n \times \mathbb{P}^m \) and let \( \pi : \Gamma_\phi \rightarrow \mathbb{P}^n \) be the projection. In the notation of [2], the shadow of
\[ \Gamma_{\phi} \text{ is the class} \]
\[ [G] = g_0 + g_1 h + \cdots + g_n h^n \in A_*(\mathbb{P}^n), \quad (2.12) \]

where \( g_i = \deg(\pi_*(\mathcal{I} \cdot [\Gamma_{\phi}])) \), these \((g_0, \ldots, g_n)\) are also the projective degrees of the map \( \phi \).

We give a method to compute the projective degrees \( g_i \) in Theorem 2.3.1 below.

### 2.2.2 Segre classes

In this subsection we state a result of Aluffi [1] (Proposition 2.2.1) which can be used to calculate Segre classes of projective varieties. When combined with result of Theorem 2.3.1 this yields our algorithm to compute Segre classes of projective varieties described in Algorithm 2.3.2. We also review several previous results on the computation of Segre classes, the first due to Aluffi [1] and the second due to Eklund, Jost and Peterson [8].

In (2.16) we make explicit the relationship between the projective degrees of a rational map and the degrees of the residual set considered in [8].

Aluffi [2] gives the following result which allows for the computation of the Segre class of \( V \) in \( \mathbb{P}^n \) for \( V \) a subscheme of \( \mathbb{P}^n \).

**Proposition 2.2.1** (Proposition 3.1 [2]). Let \( I = (f_0, \ldots, f_m) \subset k[x_0, \ldots, x_n] \) be a homogeneous ideal defining a scheme \( V \subset \mathbb{P}^n \) and let \( h = c_1(\mathcal{O}_{\mathbb{P}^n}(1)) \) be the class of a hyperplane in \( A_*(\mathbb{P}^n) \). Since \( I \) is homogeneous we may assume, without loss of generality, that the \( \deg(f_i) = d \) for all \( i \). Let \( \phi : \mathbb{P}^n \to \mathbb{P}^m \) be the rational map specified by
\[ p \mapsto (f_0(p) : \cdots : f_m(p)), \]

let \((g_0, \ldots, g_n)\) be the projective degrees of \( \phi \) and let \( \Gamma_{\phi} \subset \mathbb{P}^n \times \mathbb{P}^m \) be the graph of \( \phi \). Write \([G]\) for the class of the shadow of the graph of the map \( \phi \) (see (2.12)), i.e.
\[ [G] = g_0 + g_1 h + \cdots + g_{n-1} h^{n-1} + g_n h^n \]
in $A_*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1})$. Then we have:

$$s(V, \mathbb{P}^n) = 1 - c(\mathcal{O}(dh))^{-1} \cdot \left( \sum_{i=0}^{n} \frac{g_i h^i}{c(\mathcal{O}(dh))^i} \right)$$

$$= 1 - \sum_{i=0}^{n} \frac{g_i h^i}{(1 + dh)^{i+1}} \in A^*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1}).$$

To use Proposition 2.2.1, Aluffi [2] notes that $\Gamma_\phi$ can be obtained explicitly as $\Gamma_\phi$ is isomorphic to the blow-up of $\mathbb{P}^n$ along $V$, and once $\Gamma_\phi$ is known the class $[G]$ can be computed directly. Specifically the algorithm of Aluffi is as follows,

- obtain $\Gamma_\phi$ explicitly (by computing $Bl_V \mathbb{P}^n \cong \Gamma_\phi$, that is the blow-up of $\mathbb{P}^n$ along $V$)
- intersect $\Gamma_\phi$ with general hyperplanes
- project the intersections down to $\mathbb{P}^n$, and compute the degree of the projections to obtain the class of the shadow of the graph, $[G]$.

Hence the main computational cost for finding Segre classes using the method of [2] is that of finding the blow-up of $\mathbb{P}^n$ along $V$.

Another method for computing Segre classes was given by Eklund, Jost and Peterson [8]. This method does not use the relation between the class of the shadow of the graph $[G]$ (see (2.12)) and the Segre class $s(V, \mathbb{P}^n)$; we summarize the result in Proposition 2.2.2 below.

**Proposition 2.2.2** (Theorem 3.2 [8]). Let $V \subset \mathbb{P}^n$ be a subscheme of dimension $\varrho$ defined by a non-zero homogeneous ideal $I = (f_0, \ldots, f_m) \subset k[x_0, \ldots, x_n]$ with the generators $f_i$ having degree $d$. Let

$$s(V, \mathbb{P}^n) = s_n + \cdots + s_0 h^\varrho \in A_*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1})$$

be the Segre class of $V$ in $\mathbb{P}^n$. For $n - \varrho \leq j \leq n$ and general elements $\gamma_1, \ldots, \gamma_j$ let

---

_A version of this chapter has been published in The Journal of Symbolic Computation, see Helmer [16]_
Let $J = (\gamma_1, \ldots, \gamma_j)$ and let $R_j \subset \mathbb{P}^n$ be the subscheme defined by $J : I^\infty$. Then we have

$$d^j = \deg(R_j) + \sum_{i=0}^{j-(n-\rho)} \binom{j}{i} d^{j-(n-\rho)-i} \rho_i.$$ 

To apply Proposition 2.2.2 to compute $s(V, \mathbb{P}^n)$, Eklund, Jost and Peterson [8] use the following method.

- $V = V(I)$, say $d$ is the degree of the homogeneous generators of $I$.
- Pick general degree $d$ polynomials $\omega_1, \ldots, \omega_j$ in $I$.
- For $j = n - \dim V = \text{codim}(V)$ to $j = n$ do:
  - Set $J = (\omega_1, \ldots, \omega_j)$ and let $R_j$ be the scheme defined by $J : I^\infty$.
  - Compute $\deg(R_j)$.
  - Set $p = j - \text{codim}(V)$,

$$s_p = d^j - \deg(R_j) - \sum_{i=1}^{p-1} \binom{j}{p-i} d^{p-i} \rho_i.$$ (2.13)

Hence the main computational cost in the algorithm of Eklund, Jost and Peterson [8] is the computation of $\deg(R_j)$. When done symbolically, this means the main cost arises from the computation of the saturation $J : I^\infty$ for each $j$. Eklund, Jost and Peterson [8] also explain that $\deg(R_j)$ can be computed numerically using homotopy continuation in Bertini [5].

There is, in fact, an explicit relationship between the projective degrees $(g_0, \ldots, g_n)$ of a rational map $\phi$ defined by an ideal $I$ (or equivalently the class $[G]$ of the shadow of the graph $\Gamma_{\phi}$ (2.12)) and the degrees of the residual sets $R_j$ in Proposition 2.2.2. Specifically let $V = V(I)$ be a subscheme of $\mathbb{P}^n$ where $I = (f_0, \ldots, f_m)$ is a homogeneous ideal in $k[x_0, \ldots, x_n]$ and let $[G] = g_0 + g_1 h + \cdots + g_{n-1} h^{n-1} + g_n h^n \in A_*(\mathbb{P}^n)$ be

A version of this chapter has been published in The Journal of Symbolic Computation, see Helmer [16].
the class of the shadow of the graph of \( \phi \) (as in Proposition 2.2.1). Since \( I \) is homogeneous we may assume that \( \deg(f_i) = d \) for all \( i = 1, \ldots, m \). Take \( \nu = \text{codim}(Y) \). Let

\[
s(V, \mathbb{P}^n) = s_n + \cdots + s_0 h^n \in A_*(\mathbb{P}^n)
\]

be the Segre class of \( V \) in \( \mathbb{P}^n \) and let \( \bar{s}_0 = 1, \bar{s}_1 = \cdots = \bar{s}_{\nu-1} = 0 \) and \( \bar{s}_i = -s_{i-\nu} \) for \( i \geq \nu \). Note that \( s_n = \cdots = s_{\nu+1} = 0 \), i.e. \( s_\nu \) is the first nonzero coefficient. In [17] Jost gives the following expression relating the \( g_j \) in the class of the graph \( \Gamma_I \) to the Segre class,

\[
g_j = \sum_{i=0}^{j} \binom{j}{i} d^{j-i} \bar{s}_i,
\]

which is obtained by rearranging and simplifying the expression of Proposition 2.2.1. The result of Proposition 2.2.2 gives the following expression for \( \deg(R_j) \) when \( j = \nu, \ldots, n \),

\[
\deg(R_j) = d^j - \sum_{i=0}^{j-(n-\nu)} \binom{j}{j-(n-\nu)-i} d^{j-(n-\nu)-i} \bar{s}_i.
\]

Reindexing the summation in (2.15) we have

\[
\deg(R_j) = d^j - \sum_{i=\nu}^{j} \binom{j}{i} d^{j-i} \bar{s}_{j-\nu}, \quad \text{for } j = \nu, \ldots, n.
\]

Since \( \bar{s}_0 = 1 \) and \( \bar{s}_1 = \cdots = \bar{s}_{\nu-1} = 0 \) we may rewrite the expression (2.14) for \( g_j \) as

\[
g_j = d^j - \sum_{i=\nu}^{j} \binom{j}{i} d^{j-i} \bar{s}_{j-\nu}, \quad \text{for } j = \nu, \ldots, n,
\]

and \( g_j = d^j \) for \( j = 0, \ldots, \nu-1 \). Hence we have that

\[
\deg(R_j) = g_j \text{ for } j = \nu, \ldots, n.
\]

In light of (2.16) we observe that the method for computing Segre classes of Eklund, Jost and Peterson [8] stated in Proposition 2.2.2 computes the same values as the
result of Theorem 2.3.1, and in fact, the method of Theorem 2.3.1 can be seen as a refinement of the method of [8]. In both cases similar systems of equations are considered, however we will see below that the method of Algorithm 2.3.1 tends to perform better.

2.2.3 Chern-Schwartz-MacPherson Classes

In this subsection we review previous results on the calculation of the $c_{SM}$ class of a projective variety due to Aluﬁ [1, 2] and Jost [17]. We then state Theorem 2.2.4, a result of Aluﬁ [2], which when combined with Corollary 2.3.3 below allows for the computation of the Chern-Schwartz-MacPherson class of a projective variety in the manner described in Algorithm 2.3.3.

A tangible realization of the $c_{SM}$ classes, in the case of hypersurfaces, was given by Aluﬁ in Theorem I.4 of [1]. We state the result in the following proposition.

Proposition 2.2.3 (Theorem I.4 [1]). Let $V = V(f)$ be a hypersurface of $\mathbb{P}^n$, for some $f \in k[x_0, \ldots, x_n]$, and assume without loss of generality that $f$ is squarefree (since $c_{SM}(V) = c_{SM}(V_{red})$) then

$$c_{SM}(V) = c(T_{\mathbb{P}^n}) \cdot \left( s(V, \mathbb{P}^n) + \sum_{m=0}^{n} \sum_{j=0}^{n-m} \binom{n-m}{j} (-V)^j \cdot (-1)^{n-m-j}s_{m+j}(Y, \mathbb{P}^n) \right) \quad (2.17)$$

where $s(V, \mathbb{P}^n)$ is the Segre class of $V$ in $\mathbb{P}^n$, and $Y$ is the singularity subscheme of $V$. That is, $Y$ is the scheme defined by the vanishing of the partial derivatives of $f$.

In [2], Aluﬁ uses Proposition 2.2.1 and Proposition 2.2.3 to give an algorithm to compute the $c_{SM}$ class of hypersurface in $\mathbb{P}^n$ (this algorithm can be extended to higher codimension using Proposition 2.1.3). That is for a hypersurface $V = V(f)$ in $\mathbb{P}^n$ and $Y$ the singularity scheme of $V$ (that is the scheme defined by the zeros of the partial derivatives of $f$) the algorithm of Aluﬁ [2] computes $s(Y, \mathbb{P}^n)$ by finding the blow up, as described above (immediately following Proposition 2.2.1), and then applying Proposition 2.2.3. Thus the main computational step of the algorithm is to compute the blow-up of $\mathbb{P}^n$ along $Y$ for each hypersurface. This can be implemented.
using any algorithm which computes the Rees algebra of \( Y \).

An alternative method for computing \( c_{SM} \) classes was given by Jost in [17]. This method also uses (2.17) to give an expression for the \( c_{SM} \) class of a hypersurface, however Jost computes the class \( s(Y, \mathbb{P}^n) \) by applying the method of [8] stated in Proposition 2.2.2 to compute Segre classes by calculating the degrees of residual sets.

Let \( V \) be a hypersurface of \( \mathbb{P}^n \) defined by the homogeneous polynomial ideal \((f)\) in \( k[x_0, \ldots, x_n] \), and since \( c_{SM}(V) = c_{SM}(V_{red}) \) we assume that \( f \in k[x_0, \ldots, x_n] \) is squarefree. Using the partial derivatives of \( f \) we define a rational map \( \varphi : \mathbb{P}^n \rightarrow \mathbb{P}^n \),

\[
\varphi : p \mapsto \left( \frac{\partial f}{\partial x_0}(p) : \cdots : \frac{\partial f}{\partial x_n}(p) \right). \tag{2.18}
\]

This map is referred to as the polar map or gradient map [6].

**Theorem 2.2.4** (Aluffi [2] Theorem 2.1). Assume, without loss of generality, that \( f \in k[x_0, \ldots, x_n] \) is squarefree. Let \( V = V(f) \) and let \((g_0, \ldots, g_n)\) be the projective degrees of the polar map \( \varphi \) (2.18), we have the following equality in \( A_*(\mathbb{P}^n) = \mathbb{Z}[h]/h^{n+1} \)

\[
c_{SM}(V) = (1 + h)^{n+1} - \sum_{j=0}^{n} g_j (-h)^j (1 + h)^{n-j}. \tag{2.19}
\]

Note that Theorem 2.2.4 follows from substituting the result of Proposition 2.2.1 (as stated in (2.24)) into the result of Proposition 2.2.3, (2.17).

## 2.3 Main Results and Algorithms

In this section we state and prove the main result of this chapter, Theorem 2.3.1. This theorem gives a method to compute the projective degrees of a rational map defined by a homogeneous ideal.

The result of Theorem 2.3.1 is then used to construct an algorithm to compute the
projective degrees using a computer algebra system, presented in Algorithm 2.3.1. Algorithm 2.3.1 is in turn used to construct Algorithm 2.3.2 which computes the Segre class \( s(V, \mathbb{P}^n) \) of a subscheme \( V \) of \( \mathbb{P}^n \) and Algorithm 2.3.3 which computes \( c_{SM}(V) \) and/or \( \chi(V) \).

### 2.3.1 Results

We now present the main result of this chapter, Theorem 2.3.1, which gives a method to compute the projective degrees of a rational map defined by a homogeneous ideal using a computer algebra system.

**Theorem 2.3.1.** Let \( I = (f_0, \ldots, f_m) \) be a homogeneous ideal in \( k[x_0, \ldots, x_n] \) defining a \( q \)-dimensional scheme \( V = V(I) \). The projective degrees \( (g_0, \ldots, g_n) \) of \( \phi : \mathbb{P}^n \rightarrow \mathbb{P}^n \),

\[
\phi : p \mapsto (f_0(p) : \cdots : f_m(p)),
\]

are given by

\[
g_i = \dim_k (k[x_0, \ldots, x_n, T]/(P_1 + \cdots + P_i + L_1 + \cdots + L_{n-i} + L_A + S)). \tag{2.20}
\]

Here \( P_\ell, L_\ell, L_A \) and \( S \) are ideals in \( k[x_0, \ldots, x_n, T] \) with

\[
P_\ell = \left( \sum_{j=0}^m \lambda_{\ell,j} f_j \right), \quad \lambda_{\ell,j} \text{ a general scalar in } k, \quad \ell = 1, \ldots, n,
\]

\[
S = \left( 1 - T \cdot \sum_{j=0}^m \theta_j f_j \right), \quad \theta_j \text{ a general scalar in } k,
\]

\[
L_\ell = \left( \sum_{j=0}^n \mu_{\ell,j} x_j \right), \quad \mu_{\ell,j} \text{ a general scalar in } k, \quad \ell = 1, \ldots, n,
\]

\[
L_A = \left( 1 - \sum_{j=0}^n \nu_j x_j \right), \quad \nu_j \text{ a general scalar in } k.
\]
Additionally $g_0 = 1$.

**Proof.** First we observe that by (2.14) we have that $g_0 = 1$. Fix some $i = 1, \ldots, n$. For the rational map $\phi$ the projective degrees (see (2.11)) are given by

$$g_i = \text{card} \left( \phi^{-1}(\mathbb{P}^m) \cap \mathbb{P}^i \right).$$

The inverse image under $\phi$ of a general hyperplane $\mathbb{P}^{m-1}$ in $\mathbb{P}^m$ is

$$\phi^{-1}(\mathbb{P}^{m-1}) = V \left( \sum_{j=0}^{m} \lambda_j f_j \right) - V(f_0, \ldots, f_m) \subset \mathbb{P}^n,$$

for $\lambda$ a general scalar in $k$.

and letting

$$L_\ell = \left( \sum_{j=0}^{n} \mu_{\ell,j} x_j \right), \quad \mu_{\ell,j} \text{ a general scalar in } k$$

for each $\ell$, this gives

$$g_i = \text{card} \left( \bigcap_{\ell=1}^{i} V \left( \sum_{j=0}^{m} \lambda_{\ell,j} f_j \right) \cap \bigcap_{\ell=1}^{n-i} V(L_\ell) - V(f_0, \ldots, f_m) \right).$$

Now let

$$W = \bigcap_{\ell=1}^{i} V \left( \sum_{j=0}^{m} \lambda_{\ell,j} f_j \right) \cap \bigcap_{\ell=1}^{n-i} V(L_\ell),$$

so $g_i = \text{card}(W - V(f_0, \ldots, f_m))$. Let $\tilde{W} = W - V(f_0, \ldots, f_m)$. By the Bertini theorem of Sommese and Wampler [23, §A.8.7] there exists open dense subsets $U_1 \subset k^{i \times m}$ and $U_2 \subset k^{n-i \times n}$ such that for $\lambda \in U_1$ and $\mu \in U_2$, $\tilde{W}$ has dimension 0 and $O_{\tilde{W},p}$ is a regular local ring (equivalently the Jacobian matrix of the generators of $W$ evaluated at points in $W - V(f_0, \ldots, f_m)$ has rank $n$). In what follows we take $\lambda \in U_1$ and $\mu \in U_2$. Let us write $W - V(f_0, \ldots, f_m) = \{p_0, \ldots, p_s\}$. Then

$$U_3 = \mathbb{P}^m - \bigcup_{i=0}^{s} V(f_0(p_i)x_0 + \cdots + f_m(p_i)x_m)$$

is open and dense in $\mathbb{P}^m$, because $(f_0(p_i), \ldots, f_m(p_i)) \neq (0, \ldots, 0)$ for all $i$. Take

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]
\[ \vartheta = (\vartheta_0, \ldots, \vartheta_m) \in U_3; \] then
\[ W \cap V \left( \sum_{j=0}^{m} \vartheta_j f_j \right) = V(f_0, \ldots, f_m) \]
is empty. Now consider the ideals \( L_\ell \) and \( \left( \sum_{j=0}^{m} \lambda_\ell_j f_j \right) \) as ideals in the ring \( k[x_0, \ldots, x_n, T] \), and define \( V_S = V(S) \) where
\[ S = \left( 1 - T \cdot \sum_{j=0}^{m} \vartheta_j f_j \right) \]
is an ideal in \( k[x_0, \ldots, x_n, T] \). For a point \( p \in V(f_0, \ldots, f_m) \) we have that
\[ f_j(p) = 0, \quad j = 0, 1, \ldots, m \]
which implies that \( p \) is not in \( V_S \) since \( p \) cannot be a solution to the equation \( 1 - T \cdot \sum_{j=0}^{m} \vartheta_j f_j = 0 \). Now take \( p \in W - V(f_0, \ldots, f_m) \) then
\[ T_p = \frac{1}{\sum_{j=0}^{m} \vartheta_j f_j(p)} \]
is well defined since for \( \vartheta \in U_3 \) we have that \( W \cap V \left( \sum_{j=0}^{m} \vartheta_j f_j \right) = V(f_0, \ldots, f_m) \) is empty, so \((p, T_p) \in V_S\). Now let \( \hat{W} \subset \mathbb{P}^n \times \mathbb{A}^1 \) be the variety given by a linear embedding of \( W \) in \( \mathbb{P}^n \times \mathbb{A}^1 \), where \( \mathbb{A}^1 = \text{Spec}(k[T]) \). We have
\[ \pi(\hat{W} \cap V_S) = W - V(f_0, \ldots, f_m), \quad (2.21) \]
where \( \pi \) is the projection \( \pi : \mathbb{P}^n \times \mathbb{A}^1 \mapsto \mathbb{P}^n \), and in particular
\[ \text{card}(\hat{W} \cap V_S) = \text{card}(W - V(f_0, \ldots, f_m)). \]
Rather than considering the intersection \( \hat{W} \cap V_S \) in \( \mathbb{P}^n \times \mathbb{A}^1 \) we take \( W \subset \mathbb{A}^n \) i.e. we
dehomogenize by taking

\[
W = \bigcap_{\ell=0}^{i} V \left( \sum_{j=0}^{m} \lambda_{\ell,j} f_{j} \right) \cap \bigcap_{\ell=1}^{n-i} V(L_{\ell}) \cap V(L_{A}) \subset \mathbb{A}^{n}
\]

and consider the intersection \( \widehat{W} \cap V_{S} \) in \( \mathbb{A}^{n+1} \). As the points in \( \phi^{-1} \left( \mathbb{P}^{m-i} \right) \cap \mathbb{P}^{i} \) have multiplicity one (by the Bertini theorem of Sommese and Wampler \[23, \S A.8.7\]) the cardinality of the zero dimensional set

\[
\bigcap_{\ell=0}^{i} V \left( \sum_{j=0}^{m} \lambda_{\ell,j} f_{j} \right) \cap \bigcap_{\ell=1}^{n-i} V(L_{\ell}) \cap V(L_{A}) \cap V_{S} \subset \mathbb{A}^{n+1}
\]

is given by the vector space dimension of

\[
k[x_{0}, \ldots, x_{n}, T]/(P_{1} + \cdots + P_{i} + L_{1} + \cdots + L_{n-i} + L_{A} + S).
\]

The computation of the projective degrees can be made slightly more efficient by employing the following lemma.

**Lemma 2.3.2.** Let \( I = (f_{0}, \ldots, f_{m}) \) be a homogeneous ideal in \( k[x_{0}, \ldots, x_{n}] \) defining a \( q \)-dimensional scheme \( V = V(I) \) in \( \mathbb{P}^{n} \) and assume, without loss of generality, that \( \deg(f_{i}) = d \) for all \( i = 0, \ldots, m \). Also let \( (g_{0}, \ldots, g_{n}) \) denote the projective degrees of \( \phi : \mathbb{P}^{n} \to \mathbb{P}^{m} \),

\[
\phi : p \mapsto (f_{0}(p) : \cdots : f_{m}(p)),
\]

we have that

\[
g_{k} = d^{k} \quad \text{for } k = 0, \ldots, \dim(V) - 1. \quad (2.22)
\]

**Proof.** Let \( s(V, \mathbb{P}^{n}) = s_{n} + \cdots + s_{0} h^{n} \in A_{*}(\mathbb{P}^{n}) \) be the Segre class of \( V \) in \( \mathbb{P}^{n} \) and let \( \delta_{0} = 1, \delta_{1} = \cdots = \delta_{\dim(V)-1} = 0 \) and \( \delta_{i} = -\delta_{i-\dim(V)} \) for \( i \geq \nu \). From (2.14) we have

\[
A \text{ version of this chapter has been published in The Journal of Symbolic Computation, see Helmer [16].}
\]
that

\[ g_j = \sum_{i=0}^{j} \binom{j}{i} d^{j-i} \tilde{s}_i. \tag{2.23} \]

Since \( \tilde{s}_0 = 1 \) and \( \tilde{s}_i = 0 \) for \( i = 1, \ldots, \dim(V) - 1 \) then we have that \( g_j = d^j \) for \( j = 0, \ldots, \dim(V) - 1 \).

\[ \square \]

Applying Theorem 2.3.1 and Lemma 2.3.2, we obtain Algorithm 2.3.1, which allows us to compute the projective degrees of a map \( \phi \) defined by a homogeneous ideal \( I \) in \( k[x_0, \ldots, x_n] \).

Using Theorem 2.3.1 and Lemma 2.3.2, in the form of Algorithm 2.3.1, and Proposition 2.2.1 we may compute the Segre class of a scheme \( Y \) in \( \mathbb{P}^n \) defined by an ideal \( I = (f_0, \ldots, f_m) \) in \( k[x_0, \ldots, x_n] \) as follows. Assume, without loss of generality, that all generators of \( I \) have degree \( d \). Applying Proposition 2.2.1, the projective degrees of the map

\[ \phi : \mathbb{P}^n \to \mathbb{P}^m \]

\[ p \mapsto (f_0(p) : \cdots : f_m(p)) \]

can be used to compute the Segre class of the scheme defined by the ideal \( I \) in \( \mathbb{P}^n \). Written explicitly in this case, the result of Proposition 2.2.1 becomes

\[ s(Y, \mathbb{P}^n) = 1 - \sum_{i=0}^{n} \frac{g_i h^i}{(1 + dh)^{i+1}} \in A_*(\mathbb{P}^n), \tag{2.24} \]

where \( Y = V(I) \), \( d = \deg(f_i) \) and \( (g_0, \ldots, g_n) \) are the projective degrees of the map \( \phi \).

We summarize this method for computing the Segre class in Algorithm 2.3.2.

If we take \( \phi \) in Theorem 2.3.1 above to be the polar map \( \varphi \) (see (2.18)) we have the following corollary, which will allow us to compute the Chern-Schwartz-MacPherson class and Euler characteristic of projective varieties.

**Corollary 2.3.3.** Let \( V \) be a hypersurface of \( \mathbb{P}^n \) defined by the homogeneous polynomial ideal \( (f) \) in \( k[x_0, \ldots, x_n] \). Since we take the \( c_{SM} \) class of \( V \) to be the \( c_{SM} \) class of its support, i.e. \( c_{SM}(V) = c_{SM}(V_{\text{red}}) \), we assume without loss of generality...
that $f$ is square-free. The projective degrees $(g_0, \ldots, g_n)$ of $\varphi : \mathbb{P}^n \to \mathbb{P}^n$,

$$
\varphi : p \mapsto \left( \frac{\partial f}{\partial x_0}(p) : \cdots : \frac{\partial f}{\partial x_n}(p) \right),
$$

are given by

$$
g_i = \dim_k \left( k[x_0, \ldots, x_n, T]/(P_1 + \cdots + P_i + L_1 + \cdots + L_{n-i} + L_A + S) \right). \quad (2.25)
$$

Here $P_\ell, L_\ell, L_A$ and $S$ are ideals in $R[T] = k[x_0, \ldots, x_n, T]$ with $P_\ell = \left( \sum_{j=0}^m \lambda_{\ell,j} f_j \right)$ for $\lambda_{\ell,j}$ a general scalar in $k$, $S = \left( 1 - T \cdot \sum_{j=0}^m \vartheta_j f_j \right)$ for $\vartheta_j$ a general scalar in $k$, $L_\ell$ a general homogeneous linear form for $\ell = 1, \ldots, n$ and $L_A$ a general affine linear form. Additionally $g_0 = 1$.

Corollary 2.3.3 combined with Theorem 2.2.4 can be used to compute the Chern-Schwartz-Macpherson Class and Euler characteristic of a projective hypersurface. This formula can be extended to higher codimension using the inclusion/exclusion relation for $c_{SM}$ classes, see Proposition 2.1.3. This is described explicitly in Algorithm 2.3.3.

### 2.3.2 Algorithms

In this subsection we use the results above to develop algorithms to compute the Segre class, $c_{SM}$ class and Euler characteristic of subschemes of $\mathbb{P}^n$.

Below we give Algorithm 2.3.1, an algorithm using the result of Theorem 2.3.1 to compute the projective degrees of a map $\phi : \mathbb{P}^n \to \mathbb{P}^m$, $\phi : p \mapsto (f_0(p) : \cdots : f_m(p))$ corresponding to an ideal $I = (f_0, \ldots, f_m)$ of $k[x_0, \ldots, x_n]$. $\text{R.ideal}(f_0, \ldots, f_i)$ is a function which creates the ideal $(f_0, \ldots, f_i)$ in the ring $R$ and $k.\text{random}()$ is the function which generates a general element of $k$.

**Algorithm 2.3.1. def projective_deg_map:**

- **Input:** $I = (f_0, \ldots, f_m)$ a homogeneous ideal in $k[x_0, \ldots, x_n]$, such that $\deg(f_i) = d$ for all $f_i \neq 0$. 

A version of this chapter has been published in The Journal of Symbolic Computation, see Helmer [16]
Output: The projective degrees \((g_0, \ldots, g_n)\) of a map \(\phi : \mathbb{P}^n \to \mathbb{P}^m\), \(\phi : p \mapsto (f_0(p) : \cdots : f_m(p))\).

- Set \(R = k[x_0, \ldots, x_n, T]\).
- Let \(\nu = \dim(V) in \mathbb{P}^n\)
- For \(i = 0\) to \(\nu - 1\):
  \(\triangleright g_i = d^i\)
- For \(i = \nu\) to \(n\):
  \(\triangleright P = \sum_{j=1}^{i} R.\text{ideal} \left( \sum_{j=0}^{m} k.\text{random()} \cdot f_j \right).\)
  \(\triangleright L = \sum_{j=1}^{n-i} R.\text{ideal} \left( \sum_{j=0}^{n} k.\text{random()} \cdot x_j \right).\)
  \(\triangleright L_A = R.\text{ideal} \left( 1 + \sum_{j=0}^{n} k.\text{random()} \cdot x_j \right).\)
  \(\triangleright V_S = R.\text{ideal} \left( 1 - T \sum_{j=0}^{m} k.\text{random()} \cdot f_j \right).\)
  \(\triangleright \text{zero}\_\text{dim}\_\text{ideal} = P + L + L_A + V_S \subset R.\)
  \(\triangleright g_i = \dim_k(k[x_0, \ldots, x_n, T]/\text{zero}\_\text{dim}\_\text{ideal}).\)

Return \((g_0, \ldots, g_n)\).

We now give Algorithm 2.3.2, an algorithm to compute the Segre class \(s(Y, \mathbb{P}^n)\) in \(A_*(\mathbb{P}^n)\) for \(Y\) a subscheme of \(\mathbb{P}^n\) defined by a homogeneous ideal \(J\).

Algorithm 2.3.2. def segre_proj_deg:

- Input: A homogeneous ideal \(J = (w_0, \ldots, w_m) in k[x_0, \ldots, x_n]\) defining a scheme \(Y = V(J) in \mathbb{P}^n\).
- Output: The Segre class \(s(Y, \mathbb{P}^n) in A_*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1})\).

A version of this chapter has been published in The Journal of Symbolic Computation, see Helmer [16]
Compute \((g_0, \ldots, g_n) = \text{projective\_deg\_map}(J)\) (i.e. calculate \((g_0, \ldots, g_n)\) using Algorithm 2.3.1 above).

Compute \(s(Y, P^n) = 1 - \sum_{i=0}^n g_i h^i (1 + dh)^{n-i}\), see (2.24).

return \(s(Y, P^n)\).

Below we present Algorithm 2.3.3, an algorithm to compute the Chern-Schwartz-Macpherson class \(c_{SM}(V)\) in \(A_\ast(P^n)\) and/or the Euler characteristic \(\chi(V)\) for \(V\) a subscheme of \(P^n\) defined by a homogeneous ideal \(I\). The function \(L_I,\text{parity}(f)\) above is a function such that \(L_I,\text{parity}(f) = 1\) if \(f\) is a product of an odd number of generators of \(I\) and \(L_I,\text{parity}(f) = -1\) if \(f\) is a product of an even number of generators of \(I\).

**Algorithm 2.3.3.** def csm_polar:

- **Input:** A homogeneous ideal \(I = (f_0, \ldots, f_r)\) in \(k[x_0, \ldots, x_n]\) defining a scheme \(V = V(I)\) in \(P^n\).

- **Output:** \(c_{SM}(V)\) in \(A_\ast(P^n) \cong \mathbb{Z}[h]/(h^{n+1})\) and/or the integer \(\chi(V)\).

- Make a list \(L_I\) of all generators and all products of generators of the ideal \(I\).

- For \(f\) in \(L_I\):

  - Set \(J = \left(\frac{\partial f}{\partial x_0}, \ldots, \frac{\partial f}{\partial x_n}\right)\).

  - Compute the projective degrees

  \[(g_0, \ldots, g_n) = \text{projective\_deg\_map}(J)\] [See Algorithm 2.3.1].

  - Compute \(c_{SM}(V(f)) = (1 + h)^{n+1} - \sum_{j=0}^n g_j (-h)^j (1 + h)^{n-j}\), see Theorem 2.2.4.

  - Store \(c_{SM}(V(f))\).
Apply the inclusion/exclusion property of $c_{SM}$ classes (Proposition 2.1.3) to obtain
\[ c_{SM}(V) = \sum_{f \in \mathcal{L}_I} \mathcal{L}_I \cdot \text{parity}(f) \cdot c_{SM}(V(F_{I_S})) \]

Return $c_{SM}(V)$ and/or $\chi(V) = \int c_{SM}(V)$.

### 2.4 Performance

In this section we compare the performance of our algorithms to compute Segre classes, $c_{SM}$ classes and Euler to other existing algorithms. All algorithms are implemented in Macaulay2 [13] to offer a fair comparison for testing purposes. The Macaulay2 [13] implementations use Bertini [5] for numerical computations when a numeric option is provided. The methods segre proj deg (Algorithm 2.3.2) and csm_polar (Algorithm 2.3.3) are also implemented in Sage [24] and timings for the Sage implementation of csm_polar (Algorithm 2.3.3) are included in Table 2.2. The Sage implementation of our algorithm uses PHCpack [26] for the numerical computation option.

A list of all examples used for testing benchmarks in Table 2.1 and Table 2.2 can be found below in Appendix A. The examples are given in the form of Macaulay2 [13] input.


A version of this chapter has been published in The Journal of Symbolic Computation, see Helmer [16]
Segre (Aluffi) and CSM (Aluffi) refer to the algorithms of Aluffi [2], as implemented by Aluffi in the Macaulay2 program available from Aluffi’s webpage, http://www.math.fsu.edu/~aluffi/CSM/CSM.html. The main computational step in the both algorithms of Aluffi is the computation of the Rees algebra. Specifically to calculate $s(V, \mathbb{P}^n)$ Aluffi computes $Bl_Y \mathbb{P}^n$ and to calculate $c_{SM}(V)$ Aluffi computes $Bl_Y \mathbb{P}^n$ for $Y$ the singularity subscheme of each hypersurface appearing in Proposition 2.1.3.

The algorithm segreClass (E.J.P.) is the algorithm based on Proposition 2.2.2 given by Eklund, Jost and Peterson in [8]. CSM (Jost) is the algorithm described in [17]. For testing of both segreClass (E.J.P.) and CSM (Jost) we used the implementation of Jost available in the “CharacteristicClasses” Macaulay2 package on the webpage http://www.math.illinois.edu/Macaulay2/doc/Macaulay2-1.6/share/doc/Macaulay2/CharacteristicClasses/html/. In Macaulay2 version 1.7 and above Jost’s implementations are accessed using the option “Algorithm=>ResidualSymbolic”. The main computational step for the algorithms of both [8] and [17] is the computation of the saturations $J : I^n$ to compute the residuals as in (2.13). Specifically to calculate $s(V, \mathbb{P}^n)$ Jost’s implementation computes the residuals via saturations as described in Proposition 2.2.2 and to calculate $c_{SM}(V)$ the implementation computes $s(Y, \mathbb{P}^n)$ in the same way for $Y$ the singularity subscheme of each hypersurface appearing in Proposition 2.1.3.

The method segre_proj_deg uses Algorithm 2.3.2. Algorithm 2.3.3 is referred to as csm_polar in Table 2.2; the Macaulay2 implementation is referred to as csm_polar (M2) and the Sage implementation is csm_polar (Sage). The primary computational cost of Algorithm 2.3.2 and Algorithm 2.3.3 is the computation of the projective degrees $(g_0, \ldots, g_n)$ which is done by computing the vector space dimension of a ring modulo a zero dimension ideal. This computation can be done symbolically using Gröbner bases or numerically using Bertini [5] or some other package for homotopy continuation.

All symbolic computations are performed over the finite field with 32749 elements, the numeric computations are done over $\mathbb{Q}$. Note that the $c_{SM}$ class is, technically,
only defined when working over fields of characteristic zero (see, for example, [3]
for further discussion), however since the result of the computation is the same
when working over \( \mathbb{Q} \) and over a finite field for a large prime on all examples con-
sidered we give the run times over the finite field with 32749 elements for symbolic
computations. This approach is also used for example computations of characteris-
tic classes by Aluffi [2] and Jost [17], as well as by Eklund, Jost and Peterson [8].
We also note that even when the symbolic methods are run over \( \mathbb{Q} \) they still per-
form better than the numeric versions for each algorithm. All computations were
performed on a computer with a 2.40GHz Intel Core i5-450M CPU and 4 GB of
RAM.

We would also like to remark that in the process of developing Algorithm 2.3.1
we considered other methods to remove the points in \( V(f_1, \ldots, f_n) \) (see Theorem
2.3.1) which involved performing primary decompositions and evaluating at points
in \( V(f_1, \ldots, f_n) \). However, the main speed up over the algorithm of [8] and over the
direct numeric calculations was achieved by structuring the equations as they are
given in Theorem 2.3.1, i.e. by adding the ideal

\[
S = \left( 1 - T \cdot \sum_{j=0}^{m} \vartheta_j f_j \right), \quad \vartheta_j \text{ a general scalar in } k,
\]

and working in \( k[x_0, \ldots, x_n, T] \).

The algorithms of Eklund, Jost and Peterson [8] and Jost [17] consider similar al-
gebraic objects (namely the degrees of the residual sets, see Proposition 2.2.2) to
those used in the calculation of the projective degrees in Algorithm 2.3.1. As such
it is likely that the performance of the algorithms of [8] and [17] could also be im-
proved by structuring the equations of the residuals considered in [8] in the same
way as we do here to compute the projective degrees using Theorem 2.3.1.
Table 2.1: Run time comparison of different algorithms for computing the Segre class of a projective variety. Timings for a numerical implementation of the algorithms using Bertini [5] are included in brackets where available. We use - to denote computations that were stopped after ten minutes (600 s), for the numeric computations that do not finish in less than ten minutes we simply omit the result.

2.4.1 Timings for the Computation of Segre Classes

In Table 2.1 we compare the running times of the Segre class computation method using Algorithm 2.3.2 with the running times of two other algorithms to compute Segre classes.

The method of Algorithm 2.3.2 and that of Eklund, Jost and Peterson [8] also have numeric implementations, which use the program Bertini [5] for homotopy continuation. However, the numeric implementations of both algorithms are significantly slower than the corresponding symbolic implementations. Only one example in Table 2.1 finished running in the allotted time (this is the rational normal curve in \(\mathbb{P}^7\)); the numeric timings are listed in brackets for this case.

We note that for all examples except the degree 21 variety in \(\mathbb{P}^9\) and the Segre embedding of \(\mathbb{P}^2 \times \mathbb{P}^3\) in \(\mathbb{P}^{11}\) our algorithm performs favourably in comparison to the other algorithms. For these two examples it seems that the particular structure of the ideals being considered happens to favour the computation of the Rees algebra. These examples were included to show that even though Algorithm 2.3.2 tends to be faster in general there are still some cases where the special structure of the ideal being considered makes another technique, such as computing the Rees algebra, more advantageous. Such outliers are less likely to turn up in the \(c_{SM}\) class computations since for any codimension greater than one we must compute many

\[
\begin{array}{|l|c|c|c|}
\hline
\text{Input} & \text{Segre (Aluffi [2])} & \text{segreClass(E.J.P. [8])} & \text{segre_proj_deg (Alg. 2.3.2)} \\
\hline
\text{Rational normal curve in } \mathbb{P}^7 & - & 7s (9s) & 0.5s (15s) \\
\text{Segre embedding of } \mathbb{P}^2 \times \mathbb{P}^3 \text{ in } \mathbb{P}^{11} & 2s & - & 3.0s \\
\text{Smooth deg. 81 variety in } \mathbb{P}^7 & - & 36.4s & 8.2s \\
\text{Degree 10 variety in } \mathbb{P}^8 & - & 59s & 0.9s \\
\text{Degree 21 variety in } \mathbb{P}^9 & 0.5s & 33s & 0.9s \\
\text{Degree 48 variety in } \mathbb{P}^6 & - & 173s & 2.9s \\
\hline
\end{array}
\]

A version of this chapter has been published in The Journal of Symbolic Computation, see Helmer [16].
classes of different ideals arising from the inclusion/exclusion, and hence the special structure of any one particular ideal plays less of a role.

2.4.2 Timings for the Computation of $c_{SM}$ Classes and Euler Characteristics

In Table 2.2 we compare the running times of our algorithm to compute the $c_{SM}$ class and Euler characteristic (Algorithm 2.3.3) with the running times of several other algorithms to compute the $c_{SM}$ class and Euler characteristic.

The function euler in Table 2.2 is the built in Macaulay2 function which calculates Hodge numbers to compute the Euler characteristic, and does not compute the $c_{SM}$ class. The method euler only works for smooth projective varieties. Note that the Hodge numbers are found by computing the ranks of appropriate cohomology rings and this process is computationally expensive in general; this is likely the reason that the euler function does not perform well for examples in larger ambient dimension and with larger degree.

We observe that the symbolic implementation of the algorithm described in Algorithm 2.3.3 performs better than the other existing algorithms in all cases shown in Table 2.2. It is perhaps not surprising that the algorithm of Aluffi [2] takes longer than the others in many cases as it computes the Rees algebra for each hypersurface, which is in general rather difficult. The algorithm of Jost [17] computes the Segre class explicitly, using saturations to find the residuals, before computing the $c_{SM}$ class. This also seems to be slower in general than the projective degree calculations of Algorithm 2.3.3.

We observe that the numeric implementations of the algorithm of Jost [17] and csm_polar are slower than their symbolic counterparts in all tested cases, with the majority not finishing in the allotted time of ten minutes. As was the case with the Segre class computations the symbolic implementation of each algorithm tends to be much faster regardless of which algorithm or which numerical package is used.

A version of this chapter has been published in The Journal of Symbolic Computation, see Helmer [16]
The reason for the consistently superior performance of the symbolic methods for the types of equations considered in these characteristics class computations is not clear to us. We do, however, believe that the numeric implementations could still be useful for computation both now and in the future, as they are easily parallelizable and their effectiveness on these types of systems could improve over time.

To give the reader a more clear picture of where computation time is spent we consider in more detail the smooth degree 6 variety in $\mathbb{P}^7$ from Table 2.2. Call this example $V = V(f_0, f_1)$. When we compute $c_{SM}(V)$ using inclusion/exclusion the majority of the computation time (approximately 60%) is spent computing $c_{SM}(V(f_0 \cdot f_1))$. The main cost of this computation is the calculation of the projective degrees of the rational map associated to the ideal of the singularity subscheme of $V(f_0 \cdot f_1)$. To compute the required projective degrees we must, essentially, solve 6 different zero dimensional systems in a 9 dimensional affine space with the polynomials having degree at most 7. In this case the projective degrees of the rational map associated to the ideal of the singularity subscheme of $V = V(f_0, f_1)$ were $(g_0, \ldots, g_7) = (1, 4, 10, 22, 46, 94, 190, 254)$ with the last six projective degrees being the number of solutions to the 6 zero dimensional systems we have to solve. That is the 6 zero dimensional systems considered for this example have 10, 22, 46, 94, 190 and 254 solutions, respectively.

Also note that, as with all the smooth examples, we could compute this $c_{SM}$ class directly from the Segre class of $V$ using the relation

$$c_{SM}(V) = c_F(V) = (1 + h)^{n+1} \cdot s(V, \mathbb{P}^n) \in A^*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1})$$

for $V$ smooth, see (1.4) and Remark 3.1.2. Computing the $c_{SM}$ class (or Chern class since we are in the smooth case) in this manner would, of course, be much faster since we would only need to compute one Segre class $s(V, \mathbb{P}^n)$ using Algorithm 2.3.2. The running time of Algorithm 2.3.2, our algorithm to compute the Segre class $s(V, \mathbb{P}^n)$, is approximatively 0.1s for the smooth degree 6 variety in $\mathbb{P}^7$ from Table 2.2.

We believe that given the favourable performance of Algorithm 2.3.2 and Algorithm A version of this chapter has been published in The Journal of Symbolic Computation, see Helmer [16].
Table 2.2: Comparison of Algorithm 2.3.3 (csm_polar) with different known algorithms to compute the $c_{SM}$ class and Euler characteristic of a projective variety. The - denotes a computation that did not finish after running for ten minutes (600s), n/a indicates the variety is singular and hence the algorithm euler is not applicable. Numeric timings are given in brackets (-) where available, numeric computations taking longer than 600s are omitted.

2.3.3 on a wide variety of examples we can conclude that these methods provide a useful complement to the existing methods which compute Segre and $c_{SM}$ classes and the Euler characteristic for subschemes of projective space.

2.4.3 Running Time Bounds

We now give running time bounds for Algorithms 2.3.1, 2.3.2 and 2.3.3. Suppose we are considering a homogeneous ideal $I = (f_0, \ldots, f_m)$ in $k[x_0, \ldots, x_n]$ defining a $g$-dimensional scheme $V = V(I)$ in $\mathbb{P}^d$ and assume, without loss of generality, that $\deg(f_i) = d$ for all $i = 0, \ldots, m$. Throughout this subsection let $\delta(D, N)$ be the total number of arithmetic operations required to find the number of points in a zero dimensional affine variety $W$ defined by a polynomial system containing $N$ degree $D$ polynomials in $N$ variables. It will be convenient to write the complexity bounds given in this subsection in terms of $\delta$ as one could use any known algorithm which solves zero dimensional systems to compute the projective degrees required to perform Algorithms 2.3.1, 2.3.2 and 2.3.3.

In particular, using the algorithm of Lecerf [20] (given as Theorem 6.3.2 in Chapter 6) or the algorithm of Giusti, Lecerf and Salvy [12] we have that the number of
arithmetic operations to solve such a system is polynomial in $O(N^5 D^3 N)$. Using one of the algorithms of [20] or [12] and the bounds for our algorithms (Algorithms 2.3.1 and 2.3.2) in terms of $\delta$ given in Proposition 2.4.1 and Corollary 2.4.2 we have that the computation of either the projective degrees or the Segre class will require approximatively $O(\text{dim}(V)(n + 1)^5 d^3(n+1))$ arithmetic operations. Using one of the algorithms of [20] or [12] and the bounds for Algorithm 2.3.3 (our $c_{SM}$ algorithm) in terms of $\delta$ given in Corollary 2.4.3 we have that our $c_{SM}$ algorithm requires approximately $O(2^{m+1}(n - 1) \cdot (n + 1)^5(d - 1)^{3(n+1)})$ arithmetic operations.

Note that the bound $O(N^5 D^3 N)$ on the algorithms of Lecerf [20] and of Giusti, Lecerf and Salvy [12] is exponential in $N$, the ambient dimension. As such using one of these algorithms to find the number of points in the zero dimensional sets considered in Algorithms 2.3.1, 2.3.2 would result in algorithms exponential in the ambient dimension. Hence these algorithms would still become impractical when the degree and ambient dimension are too large. Using the algorithm of [20] or [12] the method to compute the $c_{SM}$ class in Algorithm 2.3.3 is exponential in both the number of generators and the ambient dimension.

There also exist known bounds on some Gröbner basis algorithms for zero dimensional systems. For example, in [15] Hashemi and Lazard show that several known Gröbner basis algorithms for zero dimensional systems (such as Lakshman [18], Lakschman and Lazard [19], and others) have running time complexities which are polynomial in an expression of order approximately $O(c \cdot N \cdot (3\bar{D})^{3N})$. Here $c$ is the maximum size of the coefficients of input polynomials, $N$ is the number of variables and $\bar{D}$ is the arithmetic mean value of the degrees of input polynomials defining the zero dimensional system. Run time bounds of similar order for other Gröbner basis algorithms applied to zero dimensional systems are also given by several authors see, for example, Faugere, Gianni, Lazard, and Mora [9].

Further we note that while all of the running time bounds for solving zero dimensional systems discussed above are essentially polynomial in the Bézout bound $D^N$ (for $N$ equations of degree $D$ in $N$ variables with $S$ solutions), which is the upper bound on our actual number of solutions $S$, the complexity is still exponential.
relative to the number of digits, \( \log(S) \), in a computer representation of the number \( S \). That is, for such an algorithm to be polynomial with respect to the number of solutions of our system, which is the number we wish to compute, we would need a bound polynomial in \( \log(S) \) rather than polynomial in \( S \) or \( D^N \) as we have here. Hence, because \( S \) is exponential in \( \log(S) \), these algorithms have complexity which is exponential relative to the number of digits in the value we wish to obtain from them (which is the number of solutions to our given zero dimensional polynomial system). In the context of the calculation of projective degrees \((g_0, \ldots, g_n)\) this means we might expect that the time to compute a given projective degree \( g_j \) (which requires we find the number of solutions to one zero dimensional system) would be roughly exponential in the number of digits in the integer \( g_j \).

In practice the current implementations of Algorithms 2.3.1, 2.3.2 and 2.3.3 use the Gröbner basis algorithms built into Sage [24] and Macaulay2 [13], and hence the running time bounds expected for the implementations depend on the appropriate choice of \( \delta \) for these Gröbner basis methods.

**Proposition 2.4.1.** Let \( I = (f_0, \ldots, f_m) \) be a homogeneous ideal in \( k[x_0, \ldots, x_n] \) defining a \( \varrho \)-dimensional scheme \( V = V(I) \) in \( \mathbb{P}^n \) and assume, without loss of generality, that \( \deg(f_i) = d \) for all \( i = 0, \ldots, m \). Also let \((g_0, \ldots, g_n)\) denote the projective degrees of \( \phi : \mathbb{P}^n \to \mathbb{P}^m \),

\[
\phi : p \mapsto (f_0(p) : \cdots : f_m(p)).
\]

We have that the number of arithmetic operations required to compute the projective degrees \((g_0, \ldots, g_n)\) using Algorithm 2.3.1 has order

\[
O(\dim(V) \cdot \delta(d + 1, n + 2)).
\]

**Proof.** From Lemma 2.3.2 we must compute the expression

\[
g_i = \dim_k (k[x_0, \ldots, x_n, T]/(P_1 + \cdots + P_i + L_1 + \cdots + L_{n-i} + L_A + S)). \quad (2.26)
\]
appearing in Theorem 2.3.1 \( \dim(V) \) times, the equation defining \( S \) will have the largest degree which will be \( d + 1 \). Note that since we work in \( k[x_0, \ldots, x_n, T] \) as an affine space we have \( n + 2 \) variables.

Examining Algorithm 2.3.2 we note that only one set of projective degrees needs to be calculated to compute the Segre class \( s(V, \mathbb{P}^n) \) hence we have the following corollary to Proposition 2.4.1.

**Corollary 2.4.2.** Let \( I = (f_0, \ldots, f_m) \) be a homogeneous ideal in \( k[x_0, \ldots, x_n] \) defining a \( \varrho \)-dimensional scheme \( V = V(I) \) in \( \mathbb{P}^n \) and assume, without loss of generality, that \( \deg(f_i) = d \) for all \( i = 0, \ldots, m \). The number of arithmetic operations required to compute the Segre class \( s(V, \mathbb{P}^n) \) using Algorithm 2.3.2 has order

\[
O(\dim(V) \cdot \delta((d + 1), n + 2)),
\]

where \( \delta \) is as in Proposition 2.4.1.

Now consider Algorithm 2.3.3 to compute the \( c_{S,M} \) class, we have the following.

**Corollary 2.4.3.** Let \( I = (f_0, \ldots, f_m) \) be a homogeneous ideal in \( k[x_0, \ldots, x_n] \) defining a \( \varrho \)-dimensional scheme \( V = V(I) \) in \( \mathbb{P}^n \) and assume, without loss of generality, that \( \deg(f_i) = d \) for all \( i = 0, \ldots, m \). Also let \( D = \deg(f_0 \cdot f_1 \cdots f_m) \). The number of arithmetic operations required to compute the \( c_{S,M}(V) \) using Algorithm 2.3.3 has order

\[
O(2^{m+1}(n - 1) \cdot \delta(D + 1, n + 2)),
\]

where \( \delta \) is as in Proposition 2.4.1.

**Proof.** There are \( 2^{m+1} \) subsets of \( \{f_0, \ldots, f_m\} \). The largest degree of a hypersurface considered in Algorithm 2.3.3 will be that of the hypersurface \( V(f_0 \cdots f_m) \). The conclusion follows from Proposition 2.4.1. \( \square \)
Bibliography


*A version of this chapter has been published in The Journal of Symbolic Computation, see Helmer [16]*


Chapter 3

An Improved Algorithm for Complete Intersections

As in Chapter 2 let \( V \) be a possibly singular subscheme of the projective space \( \mathbb{P}^n \) over \( k \) an algebraically closed field of characteristic zero. All previous methods to compute \( c_{SM}(V) \), including all those considered in Chapter 2, require the use of the inclusion/exclusion property of \( c_{SM} \) classes given in Proposition 2.1.3 when \( V \) has codimension higher than one. For \( V_1, V_2 \) subschemes of \( \mathbb{P}^n \) the inclusion/exclusion property for \( c_{SM} \) classes states

\[
c_{SM}(V_1 \cap V_2) = c_{SM}(V_1) + c_{SM}(V_2) - c_{SM}(V_1 \cup V_2). \tag{3.1}
\]

While the use of this property allows for the computation of \( c_{SM}(V) \) for \( V \) of any codimension, it requires exponentially many \( c_{SM} \) computations relative to the number of generators of \( I \). Additionally some of the schemes considered while performing inclusion/exclusion may have significantly higher degree than the original scheme \( V \). For a review of these algorithms using inclusion/exclusion see Chapter 2.

Below we discuss an algorithm that will allow for the direct computation of the \( c_{SM} \) classes of arbitrary, possibly singular, globally complete intersection subschemes of \( \mathbb{P}^n \) defined by a homogeneous polynomial ideal \( I = (f_0, \ldots, f_m) \) where the scheme
defined by \((f_0, \ldots, f_{m-1})\) is smooth (allowing for a possible rearrangement of the generators of \(I\)). We also give an extension of this method to all globally complete intersection subschemes of \(\mathbb{P}^n\) via a form of the inclusion/exclusion property of \(c_{SM}\) classes which considered only the generators of \(I\) which define a singular subscheme of \(\mathbb{P}^n\). This new method can be implemented symbolically using Gröbner bases methods or numerically using polynomial homotopy continuation via a package such as Bertini [4]. We see that this new method complements existing methods for computing \(c_{SM}\) classes by providing performance improvements, particularly when the input ideal has relatively few generators which define singular schemes (i.e. when we have to do relatively few steps in the partial inclusion/exclusion).

In Section 3.1 we review several important definitions and results which will be used in the following sections.

In Section 3.2 we give a new expression for the \(c_{SM}\) class of a complete intersection subscheme \(V(f_0, \ldots, f_m)\) of \(\mathbb{P}^n\) such that \(V(f_0, \ldots, f_{m-1})\) is smooth in Theorem 3.2.1. This result is based on an expression for the Milnor class of a scheme of this type due to Fullwood [5]. This expression allows us to state an algorithm to compute the \(c_{SM}(V)\) for a complete intersection \(V\) in \(\mathbb{P}^n\). This new algorithm offers performance improvements over the standard inclusion/exclusion method when only a few of the generators of the ideal defining the scheme \(V\) are singular. We give some running time results for this method in Table 3.1 and Table 3.2.

The Macaulay2 [7] implementation of the algorithms for computing \(c_{SM}\) classes and Euler characteristics of projective varieties presented in this chapter can be found at https://github.com/Martin-Helmer/char-class-calc. These implementations are accessed via the “CharClassCalc” package using the CSM and Euler methods and the option Alg=Composite. See Appendix A.1 for a further description of the package and its syntax.

A version of this chapter has been submitted for publication [8]
3.1 Background

The algorithm given in Section 3.2 will rely on Algorithm 2.3.2 which finds the Segre class by calculating the projective degrees of a certain rational map, and on Algorithm 2.3.1 which uses Theorem 2.3.1 to find the projective degrees using a computer algebra system.

All characteristics classes considered here will be understood to be elements of some Chow ring. Recall that we express the Chow ring of a n-dimensional non-singular variety \( M \) as \( A^\ast(M) = \oplus_{i=0}^n A^i(M) \), where \( A^i(M) \) is the Chow group of \( M \) having codimension \( i \) in \( M \), that is \( A^i(M) \) is the group of codimension \( i \)-cycles modulo rational equivalence. Where convenient we will also write \( A_j(M) \) for the Chow group of dimension \( j \), that is the group of dimension \( j \)-cycles modulo rational equivalence.

All computations of characteristic classes will take place in the Chow ring of \( \mathbb{P}^n \), \( A^\ast(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1}) \) (recall that \( h = c_1(O_{\mathbb{P}^n}(1)) \) is the rational equivalence class of a hyperplane in \( \mathbb{P}^n \) and recall that \( c_1(O_{\mathbb{P}^n}(1)) \) is the first Chern class of the line bundle \( O_{\mathbb{P}^n}(1) \), see Fulton [6, §2.5] for details).

For a smooth scheme \( X \) let \( T_X \) denote the tangent bundle to \( X \). For a vector bundle \( E \) on \( X \) let \( c(E) \) denote the total Chern class of \( E \), see Fulton [6, §3.2]. We will write \( c(X) = c(T_X) \cdot [X] \) for the total Chern class of \( X \) in the Chow ring of \( X, A^\ast(X) \).

As in Chapter 2 we will frequently abuse notation and, given a scheme \( V \) in \( \mathbb{P}^n \) we will write \( c(V) \), \( s(V, \mathbb{P}^n) \) and \( c_{SM}(V) \) for the pushforwards to \( \mathbb{P}^n \) of each characteristic class, i.e. we will consider the various characteristic classes as their pushforwards in \( A^\ast(\mathbb{P}^n) \) rather than in \( A^\ast(V) \).

There exist several different generalizations of the total Chern class to singular schemes besides the Chern-Schwartz-Macpherson class (the \( c_{SM} \) class is discussed extensively in Chapter 2 see §2.1.2 for a definition). All of these notions agree with \( c(T_V) \cdot [V] \) for nonsingular \( V \), however recall that the Chern-Schwartz-Macpherson class is unique in the sense that it is the only generalization which satisfies a prop-
erty analogous to (2.6) for any $V$, i.e.

$$
\int c_{SM}(V) = \chi(V).
$$

(3.2)

In this chapter we will also make use of another generalization of the total Chern class to singular schemes called the Chern-Fulton-Johnson class and denoted $c_{FJ}$. For simplicity we will give the definition of $c_{FJ}$ only for the case where $X$ is a closed locally complete intersection subscheme of a smooth ambient variety $M$, since this will be sufficient for our purposes here. For a complete definition and an excellent discussion of the Chern-Fulton-Johnson classes and other related notions see Aluffi [3]. Let $X$ be a closed locally complete intersection subscheme of a smooth ambient variety $M$ and let $T_M$ denote the tangent bundle of $M$, define

$$
c_{FJ}(X) = c(T_M) \cdot s(X, M).
$$

(3.3)

Also note that since we assume that $X$ is a locally complete intersection (meaning there exists a regular embedding $i : X \to M$) then by Proposition 4.1 of Fulton [6] we have

$$
c_{FJ}(X) = c(T_M) \cdot s(X, M) = c(T_M) \cdot \left( c(N_X M)^{-1} \cdot [X] \right).
$$

Here $N_X M$ is the normal bundle to $X$ in $M$ (that is the vector bundle with sheaf of sections $(I/I^2)$ where $I$ is the ideal sheaf of $X$). Finally, let $V$ be a subscheme of $M$; we define the Milnor class of $V$ as

$$
\mathcal{M}(V) = (-1)^{\text{codim}(V)} (c_{FJ}(V) - c_{SM}(V)).
$$

(3.4)

Note that other sign conventions may be used in definition of the Milnor class, we use the sign convention used by [5], see Fullwood [5] or Aluffi [3] for more details.

Let $M$ be a smooth algebraic variety and let $V$ be a subscheme of $M$. From the definition of the Milnor class in (3.4) we have the following formula for the class

A version of this chapter has been submitted for publication [8]
\(c_{SM}(V)\) in \(A^*(M)\):
\[
c_{SM}(V) = c_{FJ}(V) - (-1)^\text{codim}(V) \mathcal{M}(V). \tag{3.5}
\]

We now define several notations of Aluffi [1, §1.4] for operations in the Chow ring. Let \(\alpha = \sum_{i \geq 0} \alpha^{(i)}\) be a cycle class in \(A^*(M)\) with \(\alpha^{(i)}\) denoting the piece of \(\alpha\) of codimension \(i\) in \(A^*(M)\), that is \(\alpha^{(i)} \in A^i(M)\). Also let \(L\) be some line bundle on \(M\). Define the following notations,
\[
\alpha^\vee = \sum_{i \geq 0} (-1)^i \alpha^{(i)}, \quad \text{and} \quad \alpha \otimes_M L = \sum_{i \geq 0} \frac{\alpha^{(i)}}{c(L)^i}. \tag{3.6}
\]

In [5, §1.1], Fullwood gives a new formula for the Milnor class of a subscheme \(V \subset M\) which is a global complete intersection of any codimension with an additional assumption on the structure of \(V\).

**Theorem 3.1.1** (Theorem 1.1 of Fullwood [5]). Let \(M\) be a smooth algebraic variety over an algebraically closed field of characteristic zero. Let \(V\) be a possibly singular global complete intersection corresponding to the zero scheme of a vector bundle \(E \to M\). Let \(j = \text{rk}(E)\). Additionally assume that \(V = M_1 \cap \cdots \cap M_j\) for some hypersurfaces \(M_1, \ldots, M_j\) and assume that, for some ordering of the hypersurfaces, \(M_1 \cap \cdots \cap M_{j-1}\) is smooth. Let \(L \to M\) denote the line bundle associated to the divisor \(M_j\) and let \(Y\) denote the singularity subscheme of \(V\). Then we have
\[
\mathcal{M}(V) = \frac{c(T_M)}{c(E)} \cdot (c(E^\vee \otimes L) \cdot (s(Y, M)^\vee \otimes_M L)). \tag{3.7}
\]

Note that if \(V\) is non-singular we will have that \(\mathcal{M}(V) = 0\).

**Remark 3.1.2.** We also note that if \(V = V(I)\) is a non-singular subscheme of \(\mathbb{P}^n\) (even if it is not a complete intersection) we may simply write the following in \(A^*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1})\):
\[
c_{SM}(V) = c_F(V) = c(T_{\mathbb{P}^n}) \cdot s(V, \mathbb{P}^n) = (1 + h)^{n+1} s(V, \mathbb{P}^n). \tag{3.8}
\]

Hence we need compute only the Segre class \(s(V, \mathbb{P}^n)\); this can be done directly using
Algorithm 2.3.2 above. Thus, in particular, inclusion/exclusion is not required in the smooth case. See Fulton [6, §4.2.6] or Aluffi [3] for more details. Recall that $c_F(V)$ is the Chern-Fulton class defined in (1.4) above.

All algorithms considered in this chapter will make use of the so-called projective degrees of a rational map to compute characteristics classes. To compute the projective degrees $g_i$ we may apply Theorem 2.3.1 in the form presented in Algorithm 2.3.1. This computation is probabilistic and yields the correct result for a choice of objects lying in an open dense Zariski set of the corresponding parameter space, see Chapter 2 for details.

### 3.2 Main Results and Algorithms

In this section we describe our new algorithm to compute the $c_{SM}$ class (and hence the Euler characteristic) of a complete intersection subscheme of $\mathbb{P}^n$ over a algebraically closed field of characteristic zero.

Let $V = V(f_0, \ldots, f_m)$ be a complete intersection subscheme of $\mathbb{P}^n$ such that the scheme $V(f_0, \ldots, f_{m-1})$ is non-singular (allowing for a possible reordering of the generators) and let $J$ be the ideal generated by the $(m + 1) \times (m + 1)$ minors of the Jacobian matrix of partial derivatives of $f_0, \ldots, f_m$. The primary result needed for the algorithms described below is given in Theorem 3.2.1 which gives a formula for $c_{SM}(V)$ in terms of the Segre class of $s(Y, \mathbb{P}^n)$ where $Y = V(J) \cap V$ is the singularity subscheme of $V$. This Segre class can then be computed using (2.24) and a method to compute the projective degrees such as Theorem 2.3.1. Theorem 3.2.1 follows from Fullwood [5, Theorem 1.1]. We summarize this method in Algorithm 3.2.1.

In Proposition 3.2.2 and Corollary 3.2.3 we extend the result of Theorem 3.2.1 to any (global) complete intersection subscheme of $\mathbb{P}^n$ with a type of inclusion/exclusion which considers only the singular generators of the ideal. Hence the number of required Segre class computations is exponential in the number of singular
generators. At worst, if all generators define singular schemes, this reduces to inclusion/exclusion as in Proposition 2.1.3. We present this generalized version of Algorithm 3.2.1 in Algorithm 3.2.2 below.

In Section 3.3 we compare the running time of Algorithm 3.2.2 described below to other algorithms to compute \( c_{SM} \) classes for complete intersection varieties in \( \mathbb{P}^n \). We see that for nearly all the cases considered the new algorithm does indeed provide a performance improvement. While the new method to compute \( c_{SM} \) classes is not applicable in all cases it does seem to complement existing methods by providing an efficient approach for a certain subset of problems, particularly those where the ideal defining a complete intersection \( V \) has only a few generators which define a singular scheme.

### 3.2.1 The Main Result

Combining the relation (3.5), the result of Fullwood [5] given in (3.7), and the expression for the \( c_{FJ} \) class of a locally complete intersection of Suwa [10] we obtain Theorem 3.2.1. This result combined with Proposition 3.2.2 will allow us to devise a more efficient algorithm to compute \( c_{SM} \) classes of possibly singular complete intersection varieties.

**Theorem 3.2.1.** Let \( k \) be an algebraically closed field of characteristic zero and let \( I = (f_0, \ldots, f_m) \) be a homogeneous ideal in \( k[x_0, \ldots, x_n] \). Assume that \( V = V(I) \) is a complete intersection subscheme of \( \mathbb{P}^n \) and let \( Y \) be the singularity subscheme of \( V \). Let \( \deg(f_i) = d_i \), and further assume that \( V(f_0, \ldots, f_{m-1}) \) is smooth scheme theoretically. Let

\[
A^\ast(\mathbb{P}^n) \cong \mathbb{Z}[[h]]/(h^{n+1})
\]

denote the Chow ring of \( \mathbb{P}^n \) where \( h = c_1(\mathcal{O}_{\mathbb{P}^n}(1)) \) is the hyperplane class in \( \mathbb{P}^n \). Then
we have the following relation in $A^*(\mathbb{P}^n)$:

$$c_{SM}(V) = (1 + h)^{n+1} \cdot \prod_{i=0}^{m} \frac{d_i h}{1 + d_i h}$$

$$\frac{(-1)^n(1 + h)^{n+1}}{\prod_{i=0}^{m}(1 + d_i h)} \left( \sum_{p=0}^{m} \sum_{i=0}^{p} \binom{m - i}{p - i} (-1)^i d_m^{p-i} \cdot \tilde{c}_i \right) \cdot \left( \sum_{i=0}^{n} \frac{(-1)^i s_i h^i}{(1 + d_m h)^i} \right),$$

where we write

$$\prod_{i=0}^{m}(1 + d_i h) = \sum_{i=0}^{m} \tilde{c}_i h^i, \quad \text{and} \quad s(Y, \mathbb{P}^n) = \sum_{i=0}^{n} s_i h^i.$$

**Proof.** First consider the result of (3.7), taking $M = \mathbb{P}^n$. Since $V$ is a complete intersection it may be defined as the zero scheme of a rank $m + 1$ vector bundle $\mathcal{E}$. Let $\mathcal{L} \rightarrow \mathbb{P}^n$ be the line bundle associated to $V(f_m)$. Then we have that $\mathcal{L} = O(d_m h)$, $c(\mathcal{E}) = \prod_{i=0}^{m}(1 + d_i h)$ and $c(T_{\mathbb{P}^n}) = (1 + h)^{n+1}$. Combining this with (3.7) we have

$$\mathcal{M}(V) = \frac{c(T_{\mathbb{P}^n})}{c(\mathcal{E})} \cdot (c(\mathcal{E}^\vee \otimes \mathcal{L}) \cdot (s(Y, \mathbb{P}^n)^\vee \otimes_{\mathbb{P}^n} \mathcal{L}))$$

$$= \frac{(1 + h)^{n+1}}{\prod_{i=0}^{m}(1 + d_i h)} \sum_{p=0}^{m} \sum_{i=0}^{p} \binom{m - i}{p - i} c_1(\mathcal{E}^\vee) c_1(\mathcal{L})^{p-i} \cdot (s(Y, \mathbb{P}^n)^\vee \otimes_{\mathbb{P}^n} O(d_m h))$$

Let

$$c(\mathcal{E}) = \prod_{i=0}^{m}(1 + d_i h) = \sum_{i=0}^{m} \tilde{c}_i h^i, \quad \text{and} \quad s(Y, \mathbb{P}^n) = \sum_{i=0}^{n} s_i h^i.$$
using (3.6) we may expand the expression \((s(Y, \mathbb{P}^n))^\vee \otimes_{\mathbb{P}^n} O(d_mh))\) as,

\[
\left( \sum_{i=0}^{n} s_i h_i \right)^{\vee} \otimes_{\mathbb{P}^n} O(d_mh) = \left( \sum_{i=0}^{n} (-1)^i s_i h_i \right) \otimes_{\mathbb{P}^n} O(d_mh)
\]

\[
= \sum_{i=0}^{n} (-1)^i s_i h_i \\
= \sum_{i=0}^{n} (-1)^i s_i h_i \\
= \sum_{i=0}^{n} (-1)^i s_i h_i.
\]

We may now write,

\[
\mathcal{M}(V) = \frac{(1 + h)^{n+1}}{\prod_{i=0}^{m}(1 + d_i h)} \left( \sum_{p=0}^{m+1} h_i \sum_{p=0}^{m} \binom{m+1}{p} \binom{m+1}{p} \sum_{i=0}^{m+1} (-1)^i d_m^{p-i} \cdot \tilde{c}_i \right) \left( \sum_{i=0}^{n} (-1)^i s_i h_i \right) \\
(1 + d_m h)^i.
\]

Since \(V\) is a complete intersection in \(\mathbb{P}^n\) from Suwa [10] we have

\[
c_{FJ}(V) = (1 + h)^{n+1} \cdot \prod_{i=0}^{m} \frac{d_i h}{1 + d_i h},
\]

and applying the relation \(c_{SM}(V) = c_{FJ}(V) - (-1)^m \mathcal{M}(V)\) gives the desired result. \(\square\)

Hence we may conclude that the computation of \(c_{SM}\) classes in the case of the theorem above requires only the computation of \(s(Y, \mathbb{P}^n)\) (where \(Y\) is the singularity subscheme of \(V\)), which can be accomplished by means of the projective degree calculation of Theorem 2.3.1 for the rational map specified by the ideal corresponding to \(Y\) and an application of the formula (2.24).

The singularity subscheme \(Y\) of \(V\) as given above will be \(Y = V(J) \cap V\) where \(J\) is the ideal in \(k[x_0, \ldots, x_n]\) generated by the \((m + 1) \times (m + 1)\) minors of the \((m + 1) \times (n + 1)\) Jacobian matrix of partial derivatives, i.e. the matrix \(a_{i,j} = \left( \frac{df_i}{dx_j} \right)\) for \(i = 0, \ldots, m, j = 0, \ldots, n\) (here we index the first row and column of the Jacobian matrix by 0). In practice we will use the ideal \((I + J) : (x_0, \ldots, x_n)^{\infty}\) as the ideal of

---

A version of this chapter has been submitted for publication [8]
the singularity subscheme \( Y \).

Since the only unknown in the expression of Theorem 3.2.1 is the Segre class \( s(Y,\mathbb{P}^n) \) we may obtain an Algorithm to compute \( c_{SM} \) classes (in the setting of the theorem) by combining Theorem 3.2.1 with the method to compute Segre classes using the projective degree of a rational map given in Algorithm 2.3.2 above.

To extend the result of Theorem 3.2.1 to any complete intersection subscheme of \( \mathbb{P}^n \) we will use Proposition 3.2.2 below. For a scheme \( V = V(I) \subset \mathbb{P}^n \) this proposition describes a type of inclusion/exclusion for \( c_{SM} \) class which considers only the generators of \( I \) which define singular subschemes. If the majority of generators of \( I \) define a non-singular subscheme of \( \mathbb{P}^n \) this result combined with Theorem 3.2.1 can offer a speed advantage in comparison to methods which use only inclusion/exclusion.

**Proposition 3.2.2.** Let \( Z \subset \mathbb{P}^n \) be smooth (scheme-theoretically) and let \( X_1 = V(f_1), X_2 = V(f_2) \) be singular hypersurfaces in \( \mathbb{P}^n \). If \( V = Z \cap X_1 \cap X_2 \), then we have

\[
c_{SM}(V) = c_{SM}(Z \cap X_1) + c_{SM}(Z \cap X_2) - c_{SM}(Z \cap (X_1 \cup X_2)),
\]

(3.9)

here \( X_1 \cup X_2 \) is the scheme generated by \( f_1 \cdot f_2 \). Additionally, when \( V \) is a complete intersection each of the terms in (3.9) can be computed using Theorem 3.2.1.

**Proof.** This result follows directly from the inclusion/exclusion property of the \( c_{SM} \) class, see (2.8). \( \Box \)

**Corollary 3.2.3.** Let \( V = Z \cap V(f_1) \cdots \cap V(f_r) \) be a subscheme of \( \mathbb{P}^n \), with the subscheme \( Z \) being non-singular. Write the polynomials defining \( W = V(f_1) \cap \cdots \cap V(f_r) \) as \( F = (f_1, \ldots, f_r) \) and let \( F_{\{S\}} = \prod_{i \in S} f_i \) for \( S \subset \{1, \ldots, r\} \). Then,

\[
c_{SM}(Z \cap W) = \sum_{S \subset \{1, \ldots, r\}} (-1)^{|S|+1} c_{SM}(Z \cap V(F_{\{S\}}))
\]

where \(|S|\) denotes the cardinality of the integer set \( S \). The expressions \( c_{SM}(W \cap V(F_{\{S\}})) \) can be computed using Theorem 3.2.1 when \( V \) is a complete
intersection.

This result allows us to extend the application of Theorem 3.2.1 to complete intersections \( V = V(I) \subseteq \mathbb{P}^n \) where several of the generators of the ideal \( I \) define a singular scheme. At worst, when all of the generators are singular, this will reduce to inclusion/exclusion. However if only a few of the generators are singular this could offer improved performance over the standard inclusion/exclusion procedure.

### 3.2.2 Algorithms

In Algorithm 3.2.1 we summarize the algorithm to compute \( c_{SM} \) classes for projective varieties \( V \) satisfying the assumptions of Theorem 3.2.1. In Algorithm 3.2.2 we give an algorithm which is applicable for any subscheme \( V \) of \( \mathbb{P}^n \) defined by a homogeneous ideal. This algorithm takes advantage of the result of Corollary 3.2.3 combined with Theorem 3.2.1 when \( V \) is a complete intersection. If \( V \) is smooth the result of Remark 3.1.2 is used. If \( V \) is neither smooth nor a complete intersection then inclusion/exclusion is used.

Below we present Algorithm 3.2.1, an algorithm to compute \( c_{SM}(V) \) for \( V = V(f_0, \ldots, f_m) \) where \( V(f_0, \ldots, f_{m-1}) \) is smooth (scheme theoretically).

**Algorithm 3.2.1. Input:** A homogeneous ideal \( I = (f_0, \ldots, f_m) \) in \( k[x_0, \ldots, x_n] \) defining a complete intersection scheme \( V = V(I) \subseteq \mathbb{P}^n \) such that \( V(f_0, \ldots, f_{m-1}) \) is smooth (scheme theoretically).

**Output:** \( c_{SM}(V) \) in \( A^*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1}) \) and/or \( \chi(V) \).

- **Find the singularity subscheme** \( Y = V(J) \), of \( X \)
  - Set \( K \) equal to the \((m + 1) \times (m + 1)\) minors of the Jacobian matrix of \( I \), that is the matrix with entries \( a_{i,j} = \left( \frac{d f_i}{d x_j} \right) \) for \( i = 0, \ldots, m, \ j = 0, \ldots, n. \)
  - \( J = (K + I) : (x_0, \ldots, x_n)^\infty. \)
  - \( Y = V(J). \)
• Apply Theorem 2.3.1 with the rational map defined by the ideal \( J \) to compute the projective degrees \( g_0, \ldots, g_n \).

• Compute \( s(Y, \mathbb{P}^n) \) by using (2.24) and the projective degrees \( g_0, \ldots, g_n \) computed above.

• Apply Theorem 3.2.1 to obtain \( c_{SM}(V) \).

Below we present Algorithm 3.2.2, an algorithm to compute \( c_{SM}(V) \) for \( V = V(I) \) any subscheme of \( \mathbb{P}^n \). This algorithm takes advantage of the result of Corollary 3.2.3 combined with Theorem 3.2.1 when \( V \) is a complete intersection. If \( V \) is smooth the result of Remark 3.1.2 is used.

**Algorithm 3.2.2. Input:** a homogeneous ideal \( I = (f_0, \ldots, f_m) \) in \( k[x_0, \ldots, x_n] \) defining a scheme \( V = V(I) \subset \mathbb{P}^n \).

**Output:** \( c_{SM}(V) \) in \( A^*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1}) \) and/or \( \chi(V) \).

• if \( V \) is non-singular (i.e. if the singularity subscheme \( Y \) of \( V \) is empty):
  
  o if \( \text{codim}(V) = m + 1 \) (i.e. \( V \) is a complete intersection):
    
    ▶ \( V \) is smooth so \( s(Y, \mathbb{P}^n) = 0 \) in Theorem 3.2.1, let \( d_i = \text{deg}(f_i) \).
    
    ▶ \( c_{SM}(V) = (1 + h)^{n+1} \cdot \prod_{i=0}^{m} \frac{d_i h}{1 + d_i h} \).
    
    ▶ **Return** \( c_{SM}(V) \) and/or \( \chi(V) \).

  o Compute the projective degrees \( (g_0, \ldots, g_n) \) of the rational map defined by the ideal \( I \) using Theorem 2.3.1.

  o **Compute** \( s(V, \mathbb{P}^n) \) by using Eq. (2.24) and the projective degrees \( (g_0, \ldots, g_n) \) obtained above.

  o **Compute** \( c_{SM}(V) = (1 + h)^{n+1} s(V, \mathbb{P}^n) \).

  o **Return** \( c_{SM}(V) \) and/or return \( \chi(V) \).

• else if \( \text{codim}(V) = m + 1 \) (i.e. \( V \) is a complete intersection):

A version of this chapter has been submitted for publication [8]
for \( j = 1, \ldots, m \) and for each subset \( f_{i_0}, \ldots, f_{i_{m-j}} \) of \( f_1, \ldots, f_m \) containing \( m + 1 - j \) elements:

- if \( V(f_{i_0}, \ldots, f_{i_{m-j}}) \) is non-singular:
  - Let \( Z = V(f_{i_0}, \ldots, f_{i_{m-j}}) \).
  - Let \( F \) be the set \( f_{\ell_{m-j+1}}, \ldots, f_{\ell_m} \) and let \( F_S = \prod_{i \in S} f_i \) for \( S \subset \{ \ell_{m-j+1}, \ldots, \ell_m \} \).
  - Apply Corollary 3.2.3 to obtain
    \[
    c_{SM}(V) = \sum_{S \subset \{ \ell_{m-j+1}, \ldots, \ell_m \}} (-1)^{|S|+1} c_{SM}(Z \cap V(F_S))
    \]
    and compute each \( c_{SM} \) class in the summation using Theorem 3.2.1 as presented in Algorithm 3.2.1.
  - Return \( c_{SM}(V) \) and/or \( \chi(V) \).

- else: Compute \( c_{SM}(V) \) using Algorithm 2.3.3, that is using inclusion/exclusion.

3.3 Performance

In this section we test the performance of Algorithms 3.2.1 and 3.2.2 on a variety of examples, note that Algorithm 3.2.2 uses Algorithm 3.2.1 to perform the actual \( c_{SM} \) class computations.

For most examples the main computational cost of Algorithm 3.2.1 is the computation of the projective degrees \( g_0, \ldots, g_n \). This can be accomplished in a number of different ways. The method we will use for this computation consists of finding the degree of the zero dimensional ideal described in Theorem 2.3.1. This can be accomplished symbolically using Gröbner bases calculations, or numerically using

\[ A \text{ version of this chapter has been submitted for publication [8]} \]
<table>
<thead>
<tr>
<th>INPUT</th>
<th>CSM (Aluffi [2])</th>
<th>CSM (Jost [9])</th>
<th>csm_dir (Alg. 3.2.2)</th>
<th>csm_I_E (Alg. 2.3.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1 \subset \mathbb{P}^7$</td>
<td>-</td>
<td>-</td>
<td>0.3s (0.2s) [4.8s]</td>
<td>- (116.5s) [-]</td>
</tr>
<tr>
<td>$V_2 \subset \mathbb{P}^4$</td>
<td>-</td>
<td>1.7s [-]</td>
<td>0.3s (0.1s) [1.3s]</td>
<td>1.2s (1.2s) [44.1s]</td>
</tr>
<tr>
<td>$V_3 \subset \mathbb{P}^6$</td>
<td>-</td>
<td>27.7s [-]</td>
<td>7.2s (2.2s) [-]</td>
<td>33.2s (53.2s) [-]</td>
</tr>
<tr>
<td>$V_4 \subset \mathbb{P}^5$</td>
<td>-</td>
<td>- [-]</td>
<td>4.6s (0.7s) [5.5s]</td>
<td>- (-) [-]</td>
</tr>
<tr>
<td>$V_5 \subset \mathbb{P}^6$</td>
<td>-</td>
<td>- [-]</td>
<td>19.9s (7.9s) [24.9s]</td>
<td>- (-) [-]</td>
</tr>
</tbody>
</table>

Table 3.1: Run times (over $\mathbb{Q}$) of different algorithms for computing $c_{SM}(V)$ and $\chi(V)$ for $V$ a complete intersection subscheme of $\mathbb{P}^n$. The timings in [ ] are those of numeric implementations using Bertini [4]. The timings in () are from an implementation of the result of Proposition 2.3.1 which uses a saturation rather than computing the degree of the zero dimensional ideal to find the projective degree.

Homotopy continuation via a package such as PHCpack [11] or Bertini [4]. The symbolic methods are in general much faster. A secondary computational cost, which for some examples can become the primary computational cost, is the cost of computing the singularity subscheme. Within this computation the main cost is that of saturating out by the irrelevant ideal. Note that if one does not saturate out the irrelevant ideal when computing the singularity subscheme in Algorithms 3.2.1 and 3.2.2 one will still obtain the correct answer, however the cost of computing the projective degrees associated to the ideal of the singularity subscheme will become much greater. Hence it seems in most case the extra time to saturate by the irrelevant ideal results in a net improvement in running times for Algorithms 3.2.1 and 3.2.2.

In Table 3.1 and Table 3.2 we give the running times of the algorithm discussed here in comparison to several other algorithms which use inclusion/exclusion to compute the $c_{SM}$ class and Euler characteristic. All methods shown in the tables are implemented in Macaulay2 [7], the numeric implementations use Bertini [4]. All test computations were performed on a computer with an Intel i5-450M processor and 4 GB of RAM.

A version of this chapter has been submitted for publication [8]
In the tables in this section we take

\[ V_1 = V \left( 21x_0^2 + 5x_1^2 - 24x_2^2 + 13x_3^2 + 8x_4^2 - 106x_5^2 + 2x_6^2 + 14x_7^2, x_1^2x_5 - x_0^2x_4 \right), \]
\[ V_2 = V \left( 3x_0^2 + 19x_1^2 + 8x_2^2 + 12x_3^2 + 13x_4^2, 34x_0 + 5x_1 + 19x_2 + 127x_3 - 15x_4, 27x_0^2 - x_4^2 \right), \]
\[ V_3 = V \left( 3x_0^2 + 19x_1^2 + 8x_2^2 + 12x_3^2 + 9x_4^2 + 3x_5^2 + 25x_6^2, x_2x_3 - x_3x_4 \right), \]
\[ V_4 = V \left( 5x_0^2 + 9x_1^2 + 79x_2^2 + 2x_3^2 + 35x_4^2 + 73x_5^2, 23x_0 + 9x_1 + 7x_2 + 2x_3 + 4x_4 + 32x_5, x_2x_0x_3 - x_3x_5x_4 \right), \]
\[ V_5 = V \left( 3x_0^2 + 17x_1^2 - 47x_2^2 + 3x_3^2 + 38x_4^2 - 727x_5^2 + 12x_6^2, x_0x_6 - x_0^2, 43x_0^2 + 52x_0x_1 + 94x_1^2 + 5x_0x_2 + 13x_1x_2 + x_2^2 + x_0x_3 + 4x_1x_3 + 98x_2x_3 + x_3^2 + x_0x_4 + 74x_1x_4 + 13x_2x_4 + 71x_3x_4 + 23x_4^2 + 12x_0x_5 + 2x_1x_5 + x_2x_5 + 65x_3x_5 + 92x_4x_5 + 27x_5^2 + 5x_0x_6 + 103x_1x_6 + 38x_2x_6 + x_3x_6 + 6x_4x_6 + 2x_5x_6 + 95x_6^2 \right). \]

\( V_6 \) is a smooth variety of degree eight and codimension three in \( \mathbb{P}^{10} \) defined by three random quadratic forms. \( V_7 \) is a variety of degree eight and codimension three in \( \mathbb{P}^{10} \) defined by two random quadratic forms and one degree two polynomial which defines a singular scheme.

\[ V_8 = V \left( -2x_0^3 + 24x_1^3 + x_2^3 + x_3^3 - 7x_4^3, -9x_0^3 + 43x_1^3 + x_2^3 - 98x_3^3 - 73x_4^3, x_1x_4 - x_0x_4, x_1x_0 \right) \]
\[ V_9 = V \left( -3x_0^3 + 4x_1^3 + x_2^3 + x_3^3 - 7x_4^3 - 15x_5^3, -31x_0 + 14x_1 - 9x_2 + 17x_3 - 7x_4 - 15x_5, (x_1 - x_5)x_4, x_3x_0 \right). \]

For \( V_1 \subset \mathbb{P}^7 \) we have \( \deg(V_1) = 4 \) and \( \text{codim}(V_1) = 2 \), for \( V_2 \subset \mathbb{P}^4 \) we have \( \deg(V_2) = 4 \) and \( \text{codim}(V_2) = 3 \), for \( V_3 \subset \mathbb{P}^6 \) we have \( \deg(V_3) = 6 \) and \( \text{codim}(V_3) = 2 \), for \( V_4 \subset \mathbb{P}^5 \) we have \( \deg(V_4) = 2 \) and \( \text{codim}(V_4) = 3 \), and for \( V_5 \subset \mathbb{P}^6 \) we have \( \deg(V_5) = 8 \) and \( \text{codim}(V_5) = 3 \). The variety \( V_8 \) has dimension zero in \( \mathbb{P}^4 \) and \( \deg(V_8) = 24 \). The variety \( V_9 \) has dimension one in \( \mathbb{P}^5 \) and \( \deg(V_9) = 12 \). The equations for the examples used in Tables 3.1 and 3.2 can be also be found in

A version of this chapter has been submitted for publication [8]

The method CSM (Aluffi [2]) is the implementation of Aluffi described in [2], this implementation uses inclusion/exclusion and considers the projective degrees as the multi-degree of the blowup of $\mathbb{P}^n$ along the subscheme defined by the partial derivatives for each hypersurface considered in the inclusion/exclusion. The method CSM (Jost [9]) is the algorithm of Jost which computes the projective degrees by finding the degrees of residual sets via saturation, this method also uses inclusion/exclusion. The method csm_dir (Th. 3.2.1) is the method of Algorithm 3.2.2. The method csm_IE (Alg. 2.3.3) is the method of Algorithm 2.3.3 described in Chapter 2.

In Table 3.1 computations are performed over $\mathbb{Q}$. In Table 3.2 computations are performed over $\text{GF}(32749)$. While the $c_{SM}$ class is only defined over fields of characteristic zero doing the computations over $\text{GF}(32749)$ yields the same $c_{SM}$ classes found by working over $\mathbb{Q}$ for all examples considered here.

For the smooth variety $V_6$ the computation of $c_{SM}(V_6)$ by Algorithm 3.2.1 or Algorithm 3.2.2 calculates the singularity subscheme $Y$ of $V_6$ first, but since $V_6$ is smooth then $s(Y, \mathbb{P}^n) = 0$ is obtained immediately after $Y$ is computed without the need to calculate the projective degrees. Hence in this case very nearly all of the time is spent computing the singularity subscheme $Y$. Similarly, for the variety $V_7$ the computation of $c_{SM}(V_7)$ using Algorithm 3.2.2 spends the majority of the computation time finding the singularity subscheme of $V_7$ (approximately 90% of the 59.5s average runtime).

For the varieties $V_8$ and $V_9$ the result of Theorem 3.2.1 is not directly applicable and hence the method csm_dir (Th. 3.2.1), which is our implementation of Algorithm 3.2.2, must apply Corollary 3.2.3. We see that for the case of the variety $V_8$ Algorithm 3.2.2 still provides a marked advantage in comparison to inclusion/exclusion only. However for $V_9$ the algorithm csm_IE (Alg. 2.3.3) which does only inclusion/exclusion is faster. We note that for $V_9$ about 18s of the 19.8s computation time for csm_dir (Alg. 3.2.2) is spent on computing the singularity subschemes (i.e. about 90% of the time is spent finding the singularity subschemes).

*A version of this chapter has been submitted for publication [8]*
When computing the $c_{SM}$ classes of $V_1, V_2, V_3, V_4, V_5$ or $V_8$ the majority of the computation time of our direct algorithm (Algorithm 3.2.2) is spent calculating the projective degrees required to find the Segre class of the singularity subscheme, as one would expect.

<table>
<thead>
<tr>
<th>INPUT</th>
<th>CSM (Aluffi)</th>
<th>CSM (Jost [9])</th>
<th>$\text{csm} \text{ dir} , (\text{Alg. 3.2.2})$</th>
<th>$\text{csm} \text{ I} \text{ E} , (\text{Alg. 2.3.3})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1 \subset \mathbb{P}^7$</td>
<td>-</td>
<td>47.6s</td>
<td>0.2s</td>
<td>1.1s</td>
</tr>
<tr>
<td>$V_2 \subset \mathbb{P}^4$</td>
<td>-</td>
<td>0.3s</td>
<td>0.1s</td>
<td>0.3s</td>
</tr>
<tr>
<td>$V_3 \subset \mathbb{P}^6$</td>
<td>-</td>
<td>1.5s</td>
<td>0.2s</td>
<td>0.9s</td>
</tr>
<tr>
<td>$V_4 \subset \mathbb{P}^3$</td>
<td>-</td>
<td>-</td>
<td>0.1s</td>
<td>0.9s</td>
</tr>
<tr>
<td>$V_5 \subset \mathbb{P}^6$</td>
<td>-</td>
<td>132.6</td>
<td>0.5s</td>
<td>1.9s</td>
</tr>
<tr>
<td>$V_6 \subset \mathbb{P}^{10}$</td>
<td>-</td>
<td>-</td>
<td>21.5s</td>
<td>-</td>
</tr>
<tr>
<td>$V_7 \subset \mathbb{P}^{10}$</td>
<td>-</td>
<td>-</td>
<td>59.5s</td>
<td>-</td>
</tr>
<tr>
<td>$V_8 \subset \mathbb{P}^4$</td>
<td>-</td>
<td>67.9s</td>
<td>0.7s</td>
<td>20.9s</td>
</tr>
<tr>
<td>$V_9 \subset \mathbb{P}^3$</td>
<td>-</td>
<td>311.5s</td>
<td>19.8s</td>
<td>5.3s</td>
</tr>
</tbody>
</table>

Table 3.2: Run times of different algorithms for computing $c_{SM}(V)$ and $\chi(V)$ for $V$ a complete intersection subscheme of $\mathbb{P}^n$. We use - to denote computations that were stopped after ten minutes (600 s). All computations are performed over the finite field $\mathbb{F}(32749)$.

Overall in Tables 3.1 and 3.2 we see that, for the types of examples for which the result of Theorem 3.2.1 is applicable it offers a performance increase over the algorithms which use inclusion/exclusion. Additionally we see that the symbolic implementations tend to be faster than the numeric implementations, even when the symbolic versions run over $\mathbb{Q}$, and we also see that we can expect a further speed-up using the symbolic implementations when they are run over a finite field.

From the results in the tables we can conclude that Algorithm 3.2.1 provides a significant performance improvement for the computation of $c_{SM}(V)$ when $V = V(f_0, \ldots, f_m)$ is a complete intersection subscheme of $\mathbb{P}^n$ such that $V(f_1, \ldots, f_{m-1})$ is smooth. The performance gain offered by Algorithm 3.2.2 when one must remove several of the generators of $I = (f_0, \ldots, f_m)$ to obtain a smooth scheme is less clear, in some cases it seems to offer a performance improvement however in some cases the cost of computing several singularity subschemes and their Segre classes is too great for us to see any benefit in using Algorithm 3.2.2 over pure

A version of this chapter has been submitted for publication [8]
inclusion/exclusion.

In any case Algorithm 3.2.1 and Algorithm 3.2.2 complement other methods to compute $c_{SM}$ classes and Euler characteristics by offering an effective way to significantly improve performance for a certain class of examples. Additionally it seems likely that, with some minor heuristic adjustments to the criterion one uses to decide whether to use the specialized inclusion/exclusion of Corollary 3.2.3 or the usual inclusion/exclusion of Proposition 2.1.3, the method of Algorithm 3.2.2 would be able to offer marked improvement in many cases, and in worst cases to perform similarly to an algorithm using only inclusion/exclusion.
Bibliography


Chapter 4
Algorithms to Compute the Topological Euler Characteristic and the Chern-Schwartz-MacPherson Class of Arbitrary Subschemes of Products of Projective Space

In this chapter we present algorithms to compute the Segre and Chern-Schwartz-MacPherson classes and the Euler characteristics of arbitrary subschemes of products of projective spaces. These algorithms generalize the algorithms presented in Chapter 2 and Chapter 3 above.

We now give an example of the computation of the Segre class, the $c_{SM}$ class and the Euler characteristic for a singular variety in $\mathbb{P}^4 \times \mathbb{P}^2$. Note that since the variety $V$ considered in the example is singular the results could not be obtained with standard Chern class computations.

**Example 4.0.1.** Let $k$ be an algebraically closed field of characteristic zero and let $V = V(I)$ be the subvariety of $\mathbb{P}^4 \times \mathbb{P}^2 \cong \text{Proj}(k[x_0, \ldots, x_4]) \times \text{Proj}(k[y_0, \ldots, y_2])$ defined by the ideal

$$I = (5x_0y_0, 9x_2y_1y_2 - 4x_1y_2^2)$$
\( R = k[x_0, x_1, x_2, x_3, x_4, y_0, y_1, y_2] \). Also let \( A^*(\mathbb{P}^4 \times \mathbb{P}^2) \equiv \mathbb{Z}[h_1, h_2]/(h_1^5, h_2^3) \) be the Chow ring of \( \mathbb{P}^4 \times \mathbb{P}^2 \).

Using Algorithm 4.3.1 with input I we obtain the Segre class

\[
s(V, \mathbb{P}^4 \times \mathbb{P}^2) = 170h_1^4 h_2^2 - 30h_1^4 h_2 - 90h_1^3 h_2^2 + 3h_1^4 + 18h_1^3 h_2 + 40h_1^2 h_2^2 - 2h_1^3 + 9h_1^2 h_2 - 13h_1 h_2^2 + h_1^2 + 3h_1 h_2 + 2h_2^2 \in A^*(\mathbb{P}^4 \times \mathbb{P}^2).
\]

Using Algorithm 4.3.2 with input I we obtain the Chern-Schwartz-MacPherson class

\[
c_{SM}(V) = 13h_1^4 h_2^2 + 11h_1^4 h_2 + 23h_1^3 h_2^2 + 3h_1^4 + 16h_1^3 h_2 + 21h_1^2 h_2^2 + 3h_1^3 + 11h_1^2 h_2 + 10h_1 h_2^2 + h_1^2 + 3h_1 h_2 + 2h_2^2 \in A^*(\mathbb{P}^4 \times \mathbb{P}^2)
\]

and/or the Euler characteristic \( \chi(V) = 13 \).

In §4.1 we review previous results and relevant background that will allow us to construct the algorithms presented in the following sections.

The main results of this chapter are presented in §4.2. Let \( V \) be a subscheme of \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \). In that section we first prove Theorem 4.2.1 which gives an expression for the Segre class \( s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \) in terms of classes in the Chow ring which depend solely on the so called projective multi-degrees (see (4.9)). These projective multi-degrees generalize the projective degrees of (2.11). Theorem 4.2.2 gives a method to compute the projective multi-degrees and hence can be used to compute the Segre class \( s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \). In Theorem 4.2.3 we give an expression for the \( c_{SM} \) class of certain types of complete intersection subschemes of multi-projective space; this result extends Theorem 3.2.1 to the multi-projective setting.

In §4.3 we apply the results of §4.2 to construct algorithms to compute the Segre and Chern-Schwartz-MacPherson classes and the Euler characteristic. Our algorithm to compute Segre classes of arbitrary subschemes of products of projective space is given in Algorithm 4.3.1. This algorithm generalizes Algorithm 2.3.2 and is based directly on the results of Theorem 4.2.1 and Theorem 4.2.2. In Algorithm 4.3.2 we
present an algorithm to compute the $c_{SM}$ class in the multi-projective setting using inclusion/exclusion. In Algorithm 4.3.3 we present an algorithm to compute the $c_{SM}$ class of certain complete intersection subschemes of multi-projective space without using inclusion/exclusion. Algorithm 4.3.3 generalizes Algorithm 3.2.1.

In §4.4 we discuss the performance of these algorithms on a variety of examples. Running time bounds for Algorithm 4.3.1 and Algorithm 4.3.2 are given in §4.4.2.

The Macaulay2 [9] implementation of the algorithms for computing Segre classes, $c_{SM}$ classes and Euler characteristics of subschemes of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ presented in this chapter can be found at https://github.com/Martin-Helmer/char-class-calc. These implementations are accessed via the “MultiProjChar” package, see Appendix A.4 and the examples in Appendix A.5 for the package syntax.

4.1 Review

In this section we review some necessary background information and define some notation which will be used to prove the results and construct the algorithms presented in later sections.

4.1.1 $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ and its Chow Ring

Let $k$ be an algebraically closed field of characteristic zero. We will consider

$$\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} = \text{Proj}(k[x_0^{(1)}, \ldots, x_{n_1}^{(1)}) \times \cdots \times \text{Proj}(k[x_0^{(m)}, \ldots, x_{n_m}^{(1)})$$

so that the coordinate ring of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ will be given by

$$R = k[x_0^{(1)}, \ldots, x_{n_1}^{(1)}, x_0^{(2)}, \ldots, x_{n_2}^{(2)}, \ldots, x_0^{(m)}, \ldots, x_{n_m}^{(m)}].$$
For a multi-homogeneous polynomial \( f \) in \( R \), i.e. a polynomial homogeneous in each of block of variables \( x^{(i)} = (x^{(i)}_0, \ldots, x^{(i)}_{n_i}) \), we will let \( \deg(f) \) denote the vector in \( \mathbb{Z}^m \) given by

\[
\deg(f) = (\deg_{x^{(1)}}(f), \deg_{x^{(2)}}(f), \ldots, \deg_{x^{(m)}}(f)),
\]

and we will refer to \( \deg(f) \) as the multi-degree of \( f \). For a collection of multi-homogeneous polynomials \( f_0, \ldots, f_r \) in \( R \) we will let

\[
\max(\deg(f_0), \ldots, \deg(f_r)) = (c_1, \ldots, c_m)
\]
denote the vector in \( \mathbb{Z}^m \) with \( c_i \) being the smallest integer such that \( \deg_{x^{(i)}}(f_j) \leq c_i \) for all \( i \) and all \( j \).

For an ideal \( I \) in \( R \) and a vector \( d = (d_1, \ldots, d_m) \in \mathbb{Z}^m \) we will write \( I(d) \) for the multi-degree \( d \) part of \( I \), that is \( I(d) \) will denote the collection of polynomials \( f \in I \) such that \( \deg(f) = (d_1, \ldots, d_m) \).

Recall that the Chow ring of \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \) is given by

\[
A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \cong \mathbb{Z}[h_1, \ldots, h_m]/(h_1^{n_1+1}, \ldots, h_m^{n_m+1}),
\]

where \( h_i = c_1(\mathcal{O}_{\mathbb{P}^{n_i}}(1)) \) is the rational equivalence class of a hyperplane in \( \mathbb{P}^{n_i} \).

Recall that \( c_1(\mathcal{O}_{\mathbb{P}^{n_i}}(1)) \) denotes the first Chern class of the line bundle \( \mathcal{O}_{\mathbb{P}^{n_i}}(1) \). Hence a hypersurface \( W = V(f) \) in \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \) with \( \deg(f) = (d_1, \ldots, d_m) \) will have class \([W] = d_1h_1 + \cdots + d_mh_m \) in \( A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \).

Let \( V \) be a subscheme of \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \). As in Chapters 2 and 3 we will abuse notation and write \( s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \) and \( c_{SM}(V) \) for the pushforwards to \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \) of the Segre class and \( c_{SM} \) class respectively.
4.1.2 Previous Segre Class Algorithms in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$

In [12] Moe and Qviller give an algorithm to compute the Segre class of a subscheme of a smooth projective toric variety. This would in particular allow one to compute Segre classes of subschemes of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$.

The algorithm of Moe and Qviller [12] is based on a result which gives an expression for the Segre class of a subscheme of a smooth projective toric variety in terms of the Chow ring classes of certain residual sets which are computed via saturation. We state a simplified version of this result (for the case of subschemes of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$) below.

**Proposition 4.1.1** (Proposition 4 of [12] specialized to $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$). Let $S$ be the coordinate ring of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ and let $V = V(I)$ be a closed subscheme of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ where $I = (f_0, \ldots, f_r)$ is an ideal in $S$. Set $n = n_1 + \cdots + n_m$, let $w = \max(\deg(f_i))$ and let $\alpha = \sum_{i=1}^{m} w_i h_i$ in the Chow ring $A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) = \mathbb{Z}[h_1, \ldots, h_m]/(h_1^{n_1+1}, \ldots, h_m^{n_m+1})$.

Let $B$ be the irrelevant ideal; for generic polynomials $\tilde{f}_1, \ldots, \tilde{f}_d$ in $I(w)$, and for all $d = \text{codim}(V), \ldots, n$ let

$$ R_d = V\left((\tilde{f}_1, \ldots, \tilde{f}_d) : B^\infty \right) : I^\infty. \quad (4.2) $$

Write $s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) = s_0 + \cdots + s_{n-\text{codim}(V)}$ for the pushforward to $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ of the Segre class of $V$ in $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ with $s_i \in A^{\text{codim}(V)+i}(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})$. We have that

$$ s_0 = \alpha^{\text{codim}(V)} - [R_{\text{codim}(V)}], $$

$$ s_i = \alpha^{i+\text{codim}(V)} - [R_{i+\text{codim}(V)}] - \sum_{j=0}^{i-1} \binom{i + \text{codim}(V) - j}{j} \alpha^{-j} s_j, \quad \forall i \geq 1. $$

Note that this result of Moe and Qviller [12] generalizes the previous result of Eklund, Jost and Peterson [5] stated in Proposition 2.2.2 above, which gave an expres-
sion for the Segre class of a subscheme of $\mathbb{P}^n$ in terms of residual sets similar to the $R_d$ above. Moe and Qviller [12] describe their algorithm which uses this result to obtain the Segre classes by computing the saturations $\left( (\tilde{f}_1, \ldots, \tilde{f}_d) : B^\infty \right) : I^\infty$ in Section 5 of [12].

### 4.1.3 $c_{S,M}$ Class of a Hypersurface For a Subscheme of any Smooth Variety

We now give Theorem I.4 of Alu [2] in the setting in which it was originally stated by Alu [2]. This theorem was given for the special case of $M = \mathbb{P}^n$ above (see Proposition 2.2.3). This more general version will allow us to apply the result in the multi-projective setting.

**Proposition 4.1.2** (Theorem I.4 of Alu [2]). Let $V$ be a hypersurface in a non-singular variety $M$ and let $Y$ be the singularity subscheme of $V$. Then we have

$$c_{S,M}(V) = c(T_M) \cdot \left( s(V, M) + \sum_{m=0}^{n} \sum_{j=0}^{n-m} \binom{n-m}{j} [V]^j \cdot (-1)^{n-m}s_{m+j}(Y, M) \right) \quad (4.3)$$

where $[V]$ is the class of $V$ in $A^*(M)$. Here $s_{m+j}(Y, M)$ denotes the dimension $m + j$ component of $s(Y, M)$ and $T_M$ denotes the tangent bundle to $M$.

### 4.2 Main Results

In this section we present the main results of this chapter. We first prove Theorem 4.2.1 which extends the result of Proposition 3.1 of Alu [3] (given as Proposition 2.2.1 above) to the multi-projective setting. We then prove Theorem 4.2.2 which will extend the result of Theorem 2.3.1 to the multi-projective setting and will allow us to compute a multi-projective analogue to the projective degrees of Chapter 2 (see (2.11)). In Theorem 4.2.3 we give an expression for the $c_{S,M}$ class...
of certain types of complete intersection subschemes of multi-projective space; this result extends Theorem 3.2.1 to the multi-projective setting and is proved using an expression of Fullwood [6] for the Milnor class.

These results will allow us to extend all of the algorithms presented in Chapters 2 and 3 to the multi-projective setting.

4.2.1 The Segre Class of Subvarieties of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$

Let $\mathbb{P} = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ denote multi-projective space over an algebraically closed field of characteristic zero with coordinate ring $R$ and set $n = n_1 + \cdots + n_m$. Let $I = (f_0, \ldots, f_r)$ be a multi-homogeneous ideal in $R$ defining a subscheme $V = V(I)$ of $\mathbb{P}$. Assume, without loss of generality, that all generators of $I$ have the same multi-degree so that $\deg(f_i) = (d_1, \ldots, d_m)$ for all $i$. Define a rational map $\phi: \mathbb{P} \to \mathbb{P}^r$ given by

$$\phi: p \mapsto (f_0(p) : \cdots : f_r(p)).$$

Let $\Gamma_I \subset \mathbb{P} \times \mathbb{P}^r$ denote the closure of the graph of $\phi$. Let $\mathfrak{h}$ denote the pullback to $\mathbb{P}$ of the hyperplane class in $\mathbb{P}^r$ and let $\pi: \Gamma_I \to \mathbb{P}$ be the projection. The shadow of the graph $\Gamma_I$ is the class

$$G = \sum_{\iota=0}^n [Y_\iota] \in A^*(\mathbb{P}),$$

where $[Y_\iota] = \pi_* (\mathfrak{h}^\iota \cdot [\Gamma_I])$. Note that by definition $[Y_\iota] = \left[ \phi^{-1}(\mathbb{P}^{r-\iota}) \right]$ where $\mathbb{P}^{r-\iota}$ denotes a general hyperplane of dimension $r - \iota$ in $\mathbb{P}^r$. Put another way $[Y_\iota]$ is the class of the closure of the inverse image under $\phi$ of a general codimension $\iota$ hyperplane in $\mathbb{P}^r$. Hence we may also write

$$[Y_\iota] = \left[ V(P_1 + \cdots + P_\iota) - V(I) \right]$$

(4.7)
with the $P_i$ being general linear combinations of $(f_0, \ldots, f_r)$. Also note that $[Y_\iota] = (d_1 h_1 + \cdots + d_m h_m)^\iota$ for $\iota < \text{codim}(V)$ since $V$ has no components of codimension less than $\text{codim}(V)$, i.e. for $\iota < \text{codim}(V)$

$$[Y_\iota] = [V(P_1 + \cdots + P_r)]. \quad (4.8)$$

Observe that $[Y_\iota]$ has pure codimension $\iota$. Hence the class $[Y_\iota] \in A^*(\mathbb{P})$ will have the form

$$[Y_\iota] = \sum_{0 \leq i_1 \leq n_1, \ldots, 0 \leq i_m \leq n_m} \gamma_{(i_1, \ldots, i_m)} h_1^{i_1} \cdots h_m^{i_m}, \quad (4.9)$$

we will refer to the $\gamma_{(i_1, \ldots, i_m)}$ as the **projective multi-degrees** of the rational map $\phi$. Note that these projective multi-degrees reduce to the usual projective degree of Chapter 2 when $\mathbb{P} = \mathbb{P}^n$ is a single projective space. We will, however, often find it notationally simpler to work with the classes $[Y_\iota]$ and the class $G$ of (4.6) in the multi-projective setting.

In Theorem 4.2.1 below we use the notation of Aluffi [1, §1.4] defined previously in (3.6). Recall that if $\alpha = \sum_{i \geq 0} \alpha^{(i)} \in A^*(\mathbb{P})$ with $\alpha^{(i)}$ denoting the piece of $\alpha$ of codimension $i$ in $A^*(\mathbb{P})$ (that is $\alpha^{(i)} \in A^i(\mathbb{P})$) and if $L$ is a line bundle on $\mathbb{P}$ we will write

$$\alpha \otimes L = \sum_{i \geq 0} \frac{\alpha^{(i)}}{c(L)^i}. \quad (4.10)$$

**Theorem 4.2.1.** Let $I = (f_0, \ldots, f_r)$ be a multi-homogeneous ideal in $R$ defining a $\varrho$-dimensional scheme $V = V(I)$, and assume, without loss of generality, that all the polynomials $f_i$ generating $I$ have the same multi-degree $(d_1, \ldots, d_m)$. With $G$ as in (4.6) we have

$$s(V, \mathbb{P}) = 1 - \frac{G \otimes O_\mathbb{P}(d_1 h_1 + \cdots + d_m h_m)}{c(O_\mathbb{P}(d_1 h_1 + \cdots + d_m h_m))}. \quad (4.11)$$

*Proof.* By construction the graph $\Gamma_I$ is isomorphic to the blow-up of $\mathbb{P}$ along $V$, $Bl_V \mathbb{P}$. Note that since all generators of $I$ have the same multi-degree $(d_1, \ldots, d_m)$ then $V$ is the zero scheme of a section of $O(d_1 h_1 + \cdots + d_m h_m)^{r+1}$. Let $E = \pi^{-1}(V)$
be the exceptional divisor of the blow-up \( B\nu \mathbb{P} \). From Fulton [7, §4.4] we have that \( \sigma^*(O_{\mathbb{P}}(1)) = \pi^*(O_B(d_1h_1 + \cdots + d_nh_m)) \otimes O(-E) \) where \( \sigma : B\nu \mathbb{P} \to \mathbb{P} \) is the projection; let \([E]\) be the class of the exceptional divisor in the Chow ring of \( \mathbb{P} \times \mathbb{P}^r \). From this we have \( \eta = (d_1h_1 + \cdots + d_nh_m) - [E] \) and hence \([E] = d_1h_1 + \cdots + d_nh_m - \eta \). Applying Fulton [7, Corollary 4.2.2] (given in (1.3) above) we have

\[
\sigma(V, \mathbb{P}) = \pi_* \left( \frac{[E]}{1 + [E]} \right) = \pi_* \left( \frac{d_1h_1 + \cdots + d_nh_m - \eta}{1 + d_1h_1 + \cdots + d_nh_m - \eta} \right).
\]

We may simplify this expression as follows:

\[
\pi_* \left( \frac{d_1h_1 + \cdots + d_nh_m - \eta}{1 + d_1h_1 + \cdots + d_nh_m - \eta} \right) = \pi_* \left( \frac{[\Gamma_1]}{1 + d_1h_1 + \cdots + d_nh_m - \eta} \right)
\]

\[
= 1 - \pi_* \left( \frac{1}{1 + d_1h_1 + \cdots + d_nh_m} \cdot \frac{1 + d_1h_1 + \cdots + d_nh_m - \eta}{1 + d_1h_1 + \cdots + d_nh_m} \cdot [\Gamma_1] \right)
\]

\[
= 1 - \frac{1}{c(O(d_1h_1 + \cdots + d_nh_m))} \cdot \pi_* \left( \frac{1}{1 - \frac{\eta}{1 - d_1h_1 + \cdots + d_nh_m}} \cdot [\Gamma_1] \right)
\]

\[
= 1 - \frac{1}{c(O(d_1h_1 + \cdots + d_nh_m))} \cdot \pi_* \left( \frac{1}{1 - \frac{1}{1 - \eta}} \cdot [\Gamma_1] \otimes O(d_1h_1 + \cdots + d_nh_m) \right)
\]

\[
= 1 - \frac{G \otimes O(d_1h_1 + \cdots + d_nh_m)}{c(O(d_1h_1 + \cdots + d_nh_m))}.
\]

This concludes the proof. \(\square\)

We remark that Theorem 4.2.1 generalizes to multi-projective space the result of Aluffi [2] (given above in Proposition 2.2.1) which we used to construct Algorithm 2.3.2, our algorithm to compute the Segre class in \( \mathbb{P}^n \) given in Chapter 2.

### 4.2.2 Computing the Projective Multi-degrees

We now prove a result which will allow us to compute the classes \([Y_i]\) of (4.6), and hence to compute the class \(G\) appearing in Theorem 4.2.1, using a computer algebra system by calculating the projective multi-degrees \(\gamma_{(i_1, \ldots, i_n)}\) as in (4.9).

**Theorem 4.2.2.** Let \( R \) be the coordinate ring of \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \). For \( i = 0, \ldots, n \)
suppose that $I = (f_0, \ldots, f_r)$ is a multi-homogeneous ideal in $R$ and suppose that $Y_i = V(P_1 + \cdots + P_i) - V(I)$ with the $P_i$ being general linear combinations of $(f_0, \ldots, f_r)$, i.e. $Y_i$ is as in (4.6). $Y_i$ is a subscheme of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ having pure codimension $i$ so that

$$[Y_i] = \sum_{i_1 + \cdots + i_m = i} \gamma_{(i_1, \ldots, i_m)} h_1^{i_1} \cdots h_m^{i_m}$$

in the Chow ring

$$A^* (\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \cong \mathbb{Z}[h_1, \ldots, h_m]/(h_1^{n_1+1}, \ldots, h_m^{n_m+1}).$$

Also let $(a_1, \ldots, a_m) = (n_1, \ldots, n_m) - (i_1, \ldots, i_m)$, we have that the projective multi-degrees are given by

$$\gamma_{(i_1, \ldots, i_m)} = \dim_k (R[T]/(P_1 + \cdots + P_i + L_{(a_1, \ldots, a_m)} + L_A + S)),$$

where $P_1, \ldots, P_i$ are ideals defined by general linear combinations of the generators of $I$, i.e. for general $\lambda_{j,i}$

$$P_j = \left( \sum_{i=0}^r \lambda_{j,i} f_i \right),$$

$S$ is an ideal given by

$$S = \left( 1 - T \sum_{i=0}^r \vartheta_i f_i \right)$$

for general $\vartheta_i$, $L_{(a_1, \ldots, a_m)}$ is an ideal generated by $a_1$ general homogeneous linear forms of multi-degree $(1, 0, 0, \ldots, 0)$, $a_2$ general homogeneous linear forms of multi-degree $(0, 1, 0, \ldots, 0)$, and so on, and lastly $L_A$ is the ideal generated by the $m$ affine linear forms

$$L_A = (1 - \ell_{(1,0,0,\ldots,0)}, 1 - \ell_{(0,1,0,\ldots,0)}, \ldots, 1 - \ell_{(0,0,0,\ldots,1)}), \quad (4.11)$$

where $\ell_{(0,0,\ldots,1,\ldots,0)}$ is a homogeneous linear form having multi-degree $(0, 0, \ldots, 1, \ldots, 0)$. 


Further

\[ [Y_i] = (d_1h_1 + \cdots + d_mh_m)^\iota \in A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \quad \text{for} \ i = 0, \ldots, \text{codim}(V) - 1. \]

**Proof.** The statement for \( \iota < \text{codim}(V) \) is given in (4.8).

Now take \( \iota \) such that \( \text{codim}(V) \leq \iota \leq n \). We wish to compute the class \([Y_i]\) in the Chow ring \( A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \) where \( Y_i \) is the projective closure of the open set

\[ \tilde{Y}_i = V(P_1 + \cdots + P_i) - V(I). \]

We know that the monomial basis for \( A_0(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \) is \( h_1^{a_1} \cdots h_m^{a_m} \), further if we let \((a_1, \ldots, a_m) = (n_1, \ldots, n_m) - (i_1, \ldots, i_m)\) we see that

\[ [Y_i] \cdot h_1^{a_1} \cdots h_m^{a_m} = \gamma_{(i_1, \ldots, i_m)} h_1^{n_1} \cdots h_m^{n_m} \]

since all other terms of \([Y_i]\) must possess a higher power of some \( h^i \) and hence will vanish when multiplied by \( h_1^{a_1} \cdots h_m^{a_m} \). Now if we choose sufficiently general linear forms (so that all intersections are transverse) then the zero dimensional set associated to \( \gamma_{(i_1, \ldots, i_m)} h_1^{a_1} \cdots h_m^{a_m} \) is given by

\[ \tilde{Y}_i \cap V(L(a_1, \ldots, a_m)) = (V(P_1 + \cdots + P_i) - V(I)) \cap V(L(a_1, \ldots, a_m)) \]

and hence to find \( \gamma_{(i_1, \ldots, i_m)} \) we must find the degree of \( \tilde{Y}_i \cap V(L(a_1, \ldots, a_m)) \), i.e. the number of points in \( \tilde{Y}_i \cap V(L(a_1, \ldots, a_m)) \) since \( \tilde{Y}_i \cap V(L(a_1, \ldots, a_m)) \) has dimension zero. Hence we wish to compute

\[ \gamma_{(i_1, \ldots, i_m)} = \text{card} \left( \bigcap_{l=1}^{r} V \left( \sum_{j=0}^{r} \lambda_{l,j}f_j \right) \cap V(L(a_1, \ldots, a_m)) - V(f_0, \ldots, f_r) \right), \]
where card denotes the number of points in a zero dimensional set. Let

\[ W = \bigcap_{l=1}^{i} V \left( \sum_{j=0}^{r} \lambda_{l,j} f_j \right) \cap V(L(a_1, \ldots, a_m)). \]

By the Bertini theorem of Sommese and Wampler [13, §A.8.7] (which may be applied in this setting via the Segre embedding \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \to \mathbb{P}^N \)) we have that there exists Zariski open dense sets \( U_1, U_2 \) so that for constants \( \lambda_{l,j} \) and linear forms \( \ell_{(0, \ldots, 1, \ldots, 0)} \) chosen in \( U_1 \) and \( U_2 \) respectively we have that

\[ \tilde{W} = \bigcap_{l=1}^{i} V \left( \sum_{j=0}^{r} \lambda_{l,j} f_j \right) \cap V(L(a_1, \ldots, a_m)) - V(f_0, \ldots, f_r) \]

has dimension 0 and the Jacobian matrix of the defining equations of \( W \) evaluated at points in \( \tilde{W} = W - V(f_0, \ldots, f_r) \) has full rank. In what follows we assume that \( \lambda_{l,j} \) and \( \ell_{(0, \ldots, 1, \ldots, 0)} \) lay in the desired sets \( U_1 \) and \( U_2 \). Hence we may write the set \( \tilde{W} \) as a finite collection of points, that is we may write \( W - V(f_0, \ldots, f_r) = \{ p_0, \ldots, p_s \} \).

Then

\[ U_3 = \mathbb{P}^r - \bigcup_{i=0}^{s} V(f_0(p_i)x_0 + \cdots + f_r(p_i)x_r) \]

is open and dense in \( \mathbb{P}^r \), because \( (f_0(p_i), \ldots, f_r(p_i)) \neq (0, \ldots, 0) \) for all \( i \). Take \( \vartheta = (\vartheta_0, \ldots, \vartheta_r) \in U_3 \); then

\[ W \cap V \left( \sum_{j=0}^{r} \vartheta_j f_j \right) - V(f_0, \ldots, f_r) \]

is empty. Now consider the ideals \( L(a_0, \ldots, a_m) \) and \( \left( \sum_{j=0}^{r} \lambda_{l,j} f_j \right) \) as ideals in the ring \( R[T] \), and define \( V_S = V(S) \) where

\[ S = \left( 1 - T \cdot \sum_{j=0}^{r} \vartheta_j f_j \right) \]

is an ideal in \( R[T] \).
For a point \( p \in V(f_0, \ldots, f_r) \) we have that
\[
f_j(p) = 0, \quad j = 0, 1, \ldots, r
\]
which implies that \( p \) is not in \( V_S \) since \( p \) cannot be a solution to the equation \( 1 - T \cdot \sum_{j=0}^r \theta_j f_j = 0 \). Now take \( p \in W - V(f_0, \ldots, f_r) \) then
\[
T_p = \frac{1}{\sum_{j=0}^r \theta_j f_j(p)}
\]
is well defined since for \( \theta \in U_3 \) we have that \( W \cap V \left( \sum_{j=0}^r \theta_j f_j \right) - V(f_0, \ldots, f_r) \) is empty, so \( (p, T_p) \in V_S \). Now let \( \hat{W} \subset \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \times \mathbb{A}^1 \) be the variety given by a linear embedding of \( W \) in \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \times \mathbb{A}^1 \), where \( \mathbb{A}^1 = \text{Spec}(k[T]) \). We have
\[
\pi(\hat{W} \cap V_S) = W - V(f_0, \ldots, f_r),
\]
where \( \pi \) is the projection \( \pi : \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \times \mathbb{A}^1 \rightarrow \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \), and in particular
\[
\text{card}(\hat{W} \cap V_S) = \text{card}(W - V(f_0, \ldots, f_r)).
\]
Rather than considering the intersection \( \hat{W} \cap V_S \) in \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \times \mathbb{A}^1 \) we take \( W \subset \mathbb{A}^n \) i.e. we dehomogenize with respect to each projective space \( \mathbb{P}^{n_j} \) by taking
\[
W = \bigcap_{\ell=0}^i V \left( \sum_{j=0}^r \lambda_{\ell,j} f_j \right) \cap V(L_{(a_1, \ldots, a_m)}) \cap V(L_A) \subset \mathbb{A}^{n_1 + \cdots + n_m}
\]
and we then consider the intersection \( \hat{W} \cap V_S \) in \( \mathbb{A}^{n_1 + \cdots + n_m + 1} \). Here \( L_A \) is the collection of affine linear forms (with one affine linear form for each projective space in the product \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \)) given above in (4.11). As the points in \( \hat{W} \) have multiplicity one by the Bertini theorem of Sommese and Wampler [13, §A.8.7] (again via the Segre embedding) the cardinality of the zero dimensional set
\[
\bigcap_{\ell=0}^i V \left( \sum_{j=0}^r \lambda_{\ell,j} f_j \right) \cap V(L_{(a_1, \ldots, a_m)}) \cap V(L_A) \cap V_S \subset \mathbb{A}^{n+1}
\]
is given by the vector space dimension of

\[ R[T]/(P_1 + \cdots + P_t + L_{(a_1, \ldots, a_m)} + L_A + S). \]

\[ \square \]

### 4.2.3 The $c_{SM}$ Class of Complete Intersections

In this subsection we prove Theorem 4.2.3 which extends the result of Theorem 3.2.1 to the case where $V$ is a subscheme of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$. We will use this result to construct Algorithm 4.3.3. As with Theorem 3.2.1, starting from Theorem 1.1 of Fullwood [6] given in (3.7) and using basic relations between the $c_{SM}$ and Chern-Fulton-Johnson class, see (3.5), we obtain the following theorem.

**Theorem 4.2.3.** Let $V = V(f_0, \ldots, f_r)$ be a possibly singular global complete intersection subscheme of $\mathbb{P} = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ over an algebraically closed field of characteristic zero. Additionally assume that for some ordering of the hypersurfaces $V_0 = V(f_0), \ldots, V_r = V(f_r)$ we have that $V_0 \cap \cdots \cap V_{r-1}$ is smooth. Also let $n = n_1 + \cdots + n_m$ and let

\[ A^* (\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \cong \mathbb{Z}[h_1, \ldots, h_m]/(h_1^{n_1+1}, \ldots, h_m^{n_m+1}) \]

be the Chow ring of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ so that we have $[V_i] = d_1^{(i)} h_1 + \cdots + d_m^{(i)} h_m$ where $(d_1^{(i)}, \ldots, d_m^{(i)})$ is the multi-degree of $f_i$. Then we have

\[
\begin{align*}
c_{SM}(V) &= \frac{(1 + h_1)^{r+1} \cdots (1 + h_m)^{r+1}}{(1 + [V_0]) \cdots (1 + [V_r])} \left( [V_0] \cdots [V_r] + \left( -1 \right)^{r} \sum_{j=0}^{r} \sum_{i=0}^{j} (r+1-j) \cdot [V_j] \cdot \left( \sum_{i=0}^{n} \left( -1 \right)^{i} s^{(i)}(Y, \mathbb{P}) \right) \right),
\end{align*}
\]

(4.13)

where $c_i$ is the dimension $i$ component of $(1 + [V_0]) \cdots (1 + [V_r])$ and $s^{(i)}(Y, \mathbb{P})$ is the codimension $i$ component of the Segre class of $Y$ in $\mathbb{P}$ where $Y$ denotes the singularity subscheme of $V$.

**Proof.** Recall from (3.5) in §3.2 that $c_{SM}(V) = c_{FJ}(V) + (-1)^{r} M(V)$. Note that we have $c(T_\mathbb{P}) = (1 + h_1)^{r+1} \cdots (1 + h_m)^{r+1}$. Since $V$ is a complete intersection we have
that
\[ s(V, \mathbb{P}) = \frac{[V_0] \cdots [V_r]}{(1 + [V_0]) \cdots (1 + [V_r])}, \]

hence the Chern-Fulton-Johnson class is
\[ c_{FJ}(V) = c(T_P) \cdot s(V, \mathbb{P}) = \frac{(1 + h_1)^{n_1+1} \cdots (1 + h_{m})^{n_m+1}[V_0] \cdots [V_r]}{(1 + [V_0]) \cdots (1 + [V_r])}. \]

In the notation of Theorem 1.1 of Fullwood (see (3.7)) we have
\[ M(V) = \frac{c(T_P)}{c(E)} \cdot (c(E^\vee \otimes L) \cdot (s(Y, \mathbb{P}^\vee \otimes \mathbb{P} L)), \tag{4.14} \]

where \( E \) is the line bundle associated to \( V_0 \cap \cdots \cap V_r \) and \( L \) is the line bundle associated to \( V_r \). Hence \( c(E) = (1 + [V_0]) \cdots (1 + [V_r]) \) and \( c(L) = 1 + [V_r] \), using Remark 3.2.3. of Fulton [7] and (3.6) to expand \( c(E^\vee \otimes L) \) and \( s(Y, \mathbb{P}^\vee \otimes \mathbb{P} L \) respectively we have
\[ M(V) = \frac{(1 + h_1)^{n_1+1} \cdots (1 + h_{m})^{n_m+1}}{(1 + [V_0]) \cdots (1 + [V_r])} \left( \sum_{j=0}^r \sum_{i=0}^j (r-i)^j [V_r]^{j-i} c_i \right) \cdot \left( \sum_{i=0}^n \frac{(-1)^i s^{(i)}(Y, \mathbb{P})}{(1 + [V_r])^i} \right). \]

Putting this together gives the expression in (4.13). □

### 4.3 Algorithms for Subschemes of a Product of Projective Spaces

In this section we extend all algorithms given in Chapters 2 and 3 to the setting where \( V \) is a subscheme of some product of projective spaces.

We first use the results of Theorem 4.2.1 and of Theorem 4.2.2 to construct Algorithm 4.3.1 which computes the Segre class of a subscheme of \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \). This algorithm generalizes Algorithm 2.3.2, our algorithm to compute the Segre class in \( \mathbb{P}^n \).

In Algorithm 4.3.2 we give an algorithm which uses Proposition 4.1.2 combined
with the inclusion/exclusion property of \(c_{SM}\) classes and Algorithm 4.3.1 to construct an algorithm to compute \(c_{SM}(V)\) for \(V\) a subscheme of \(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}\). As with Segre classes, Algorithm 4.3.2 reduces to the Algorithm 2.3.3 in the case where \(V \subset \mathbb{P}^n\).

We also give Algorithm 4.3.3 which computes the \(c_{SM}\) class of a complete intersection subscheme of \(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}\) with a certain structure, this algorithm generalizes Algorithm 3.2.1. We use the result of Theorem 4.2.3 to construct Algorithm 4.3.3.

### 4.3.1 Segre Class

As above we consider a subscheme \(V\) of a product of projective spaces \(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}\) and we let \(R\) denote the coordinate ring of \(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}\). In Algorithm 4.3.1 we give an algorithm to compute the Segre class \(s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})\). This algorithm is based on the results of Theorem 4.2.1 and Theorem 4.2.2. More specifically we use Theorem 4.2.1 to give an expression for the Segre class \(s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})\) in terms of classes \([Y_i] \in A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})\) which we compute by calculating the projective multi-degrees using Theorem 4.2.1. Note this algorithm generalizes Algorithm 2.3.2 to multi-projective space and is constructed in a similar manner.

Let \(R\).random(multiDeg =\((d_1, \ldots, d_m))\) be a function which creates a general polynomial \(f\) in \(R\) having multi-degree \(\text{deg}(f) = (d_1, \ldots, d_m)\). Let UnitVector\((m, i)\) be a function which makes a vector of length \(m\) with 1 as the \(i^{th}\) entry and zero for all other entries, hence \(R\).random(multiDeg =UnitVector\((m, i)\)) makes a linear form in \(R\) having multi-degree \((0, \ldots, 1, \ldots, 0)\) of length \(m\) with 1 as the \(i^{th}\) entry. We will also use the tensor notation of (4.10) in the description of our algorithm; this notation is easily implemented in a computer algebra system.

**Algorithm 4.3.1.** Input: A multi-homogeneous ideal \(I = (f_0, \ldots, f_r)\) in the coordinate ring \(R\) of \(\mathbb{P} = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}\) defining a scheme \(V = V(I)\).

Output: \(s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})\) in \(A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})\).
• Compute the Chow ring $A = A^\ast(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \cong \mathbb{Z}[h_1, \ldots, h_m]/(h_1^{n_1+1}, \ldots, h_m^{n_m+1})$ from the structure of the graded coordinate ring $R$.

• Let $n = n_1 + \cdots + n_m$ be the dimension of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$.

• Let $d = (d_1, \ldots, d_m) = \max(\deg(f_0), \ldots, \deg(f_r))$.

• Let $P_j = \sum_{l=0}^{r} R \cdot \text{random(multiDeg}=d - \deg(f_j)) \cdot \lambda_{j,l} f_l$ for $j = 1, \ldots, n$ and for general $\lambda_{j,l}$.

• Let $c = \text{codim}(V)$.

• For $\iota = c$ to $n$:
  
  • $J_\iota = R[T].\text{ideal}(P_1, \ldots, P_\iota)$.
  
  • $K_\iota = J_\iota + R[T].\text{ideal}(1 - T \cdot \sum_{j=0}^{r} \vartheta_j f_j)$; $\vartheta_j$ a general scalar in $k$.

  • Let $\Omega^{(\iota)} = \{\omega_1^{(\iota)}, \ldots, \omega_\nu^{(\iota)}\}$ denote the generators of $A^\ast(\mathbb{P})$.

  • For $\omega$ in $\Omega^{(\iota)}$:
    
    • Let $(i_1, \ldots, i_m) = \text{multdeg}(\omega)$ and set $(a_1, \ldots, a_m) = (n_1, \ldots, n_m) - (i_1, \ldots, i_m)$.

    • Let $L = 0$.

    • For $i = 1$ to $m$:
      
      • $L = L + \sum_{j=1}^{a_i} R[T].\text{ideal}(R \cdot \text{random(multiDeg}=\text{UnitVector}(m, i)))$.

      • $L_A = \sum_{j=1}^{m} R[T].\text{ideal}(1 - R \cdot \text{random(multiDeg}=\text{UnitVector}(m, j)))$.

      • Set $\gamma_\omega = \dim_k (R[T]/(K_\iota + L + L_A))$.

      • Set $[Y_\iota] = \sum_{\omega \in \Omega^{(\iota)}} \gamma_\omega \cdot \omega$ in $A^\ast(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})$.

• Set $G = \left(1 + \sum_{i=1}^{c-1} (d_1 h_1 + \cdots + d_m h_m)^i + \sum_{i=\infty}^{n} [Y_\iota]\right) \in A$. 
\[ s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) = 1 - \frac{G \otimes O_{\mathbb{P}}(d_1 h_1 + \cdots + d_m h_m)}{1 + d_1 h_1 + \cdots + d_m h_m} \in A. \]

\[ \text{Return } s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}). \]

### 4.3.2 The $c_{SM}$ Class Via Inclusion/Exclusion

In Algorithm 4.3.2 we give an algorithm to compute the $c_{SM}$ class of an arbitrary subscheme of a product of projective spaces $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$. This algorithm will make use of Algorithm 4.3.1 to compute Segre classes. It will also employ Proposition 4.1.2 (which is Theorem I.4 of Aluffi [2]) giving a relation between the $c_{SM}$ class of a hypersurface and the Segre class. To work in higher codimension we will employ the inclusion/exclusion property of the $c_{SM}$ class.

**Algorithm 4.3.2.** Input: A multi-homogeneous ideal $I = (f_0, \ldots, f_r)$ in the coordinate ring $R$ of $\mathbb{P} = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ defining a scheme $V = V(I)$. Output: $c_{SM}(V)$ in $A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})$ and/or $\chi(V)$.

- **Compute** the Chow ring $A = A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \cong \mathbb{Z}[h_1, \ldots, h_m]/(h_1^{n_1+1}, \ldots, h_m^{n_m+1})$ using the degree structure of the generators of $R$.
- $n = n_1 + \cdots + n_m$.
- Let $csm = 0 \in A$.
- Let $S$ be the set of all distinct non-empty subsets of $\{f_0, \ldots, f_r\}$.
- For $\{f_{i_1}, \ldots, f_{i_s}\} \in S$
  - Let $g = f_{i_1} \cdots f_{i_s}$ in $R$.
  - Let $J$ be the Jacobian ideal of $g$, that is the ideal defining the singularity subscheme $Y = V(J)$ of $W = V(g)$. $J$ is generated by $g$ and by the partial derivatives of $g$ with respect to the generators of $R$.
  - Let $(d_1, \ldots, d_m) = \deg(g)$.
Let \([W] = [V(g)] = d_1h_1 + \cdots + d_mh_m\).

Calculate \(s(W, \mathcal{P}) = s(V(g), \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) = \frac{[W]}{1+[W]} \in A\).

Compute \(s(Y, \mathcal{P}) = s(V(J), \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})\) using Algorithm 4.3.1.

c\((\mathcal{T} \mathcal{P}) = (1 + h_1)^{n_1+1} \cdots (1 + h_m)^{n_m+1}.

csm = csm + (-1)^{i+1}c(T_{\mathcal{P}}) \cdot \left( s(W, \mathcal{P}) + \sum_{j=0}^{n} \sum_{l=0}^{n-j} \binom{n-j}{l} [W]^l \cdot (-1)^{n-j} s_{j+1}(Y, \mathcal{P}) \right).

- \(c_{SM}(V) = \text{csm} \).
- Set \(\chi(V)\) equal to the coefficient of \(h_1^{n_1} \cdots h_m^{n_m}\) in \(c_{SM}(V)\).

Return \(c_{SM}(V)\) and/or \(\chi(V)\).

4.3.3 \(c_{SM}:\) The Complete Intersection Case

We now give an algorithm to compute \(c_{SM}(V)\) for a complete intersection subscheme \(V\) of \(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}\) which satisfies the assumptions of Theorem 4.2.3.

Algorithm 4.3.3. Input: A multi-homogeneous ideal \(I = (f_0, \ldots, f_r)\) in \(R\), the coordinate ring of \(\mathbb{P} = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}\) defining \(V = V(I)\) a complete intersection subscheme of \(\mathbb{P}\) such that \(V(f_0) \cap \cdots \cap V(f_{r-1})\) is smooth.

Output: \(c_{SM}(V)\) in \(A^*(\mathbb{P})\) and/or \(\chi(V)\).

- Compute the Chow ring \(A = A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}) \cong \mathbb{Z}[h_1, \ldots, h_m]/(h_1^{n_1+1}, \ldots, h_m^{n_m+1})\) using the degree structure of the generators of \(R\).
- \(n = n_1 + \cdots + n_m\).
- Let \(B\) be the irrelevant ideal of \(R\).
- Let \(K\) be the ideal defined by the \((r+1)\times(r+1)\) minors of the Jacobian matrix of \(I\).
- Let \(J = (K + I) : B^\infty\) so that \(Y = V(J)\) is the singularity subscheme of \(V\).
• Compute \( s(Y, P) \) using Algorithm 4.3.1.

• For \( i = 0 \) to \( n \):
  
  ◦ Set \( c_i \) equal to the dimension \( i \) component of \((1 + [V(f_0)]) \cdots (1 + [V(f_r)])\).
  
  \[
  c_{SM}(V) = \frac{(1 + h_1 p_{i+1}) \cdots (1 + h_m p_{m+1})}{(1 + [V(f_0)]) \cdots (1 + [V(f_r)])} \left\{ (1 + \sum_{j=0}^{r} \sum_{i=0}^{i} (r-j)(-1)^i V(f_i))^{i-1} c_i \right\}
  \]

  ◦ Set \( \chi(V) \) equal to the coefficient of \( h_1^{p_1} \cdots h_m^{p_m} \) in \( c_{SM}(V) \).

• Return \( c_{SM}(V) \) and/or \( \chi(V) \)

We note that Algorithm 4.3.3 could be extended to work for any complete intersection by performing inclusion/exclusion only on the singular generators of the ideal similar to Proposition 3.2.2 and Algorithm 3.2.2.

4.4 Performance

In this section we discuss the real life performance of our algorithms to compute Segre classes, \( c_{SM} \) classes and the Euler characteristic of subschemes of a product of projective spaces. We also give running time bounds for Algorithm 4.3.1 and Algorithm 4.3.3 in §4.4.2.

4.4.1 Run Time Tests

In Table 4.1 we compare the run times of our algorithm to compute the Segre class in multi-projective space (Algorithm 4.3.1) to the run times of the algorithm of Moe and Qviller [12]. For this comparison we use the Macaulay2 implementation of Moe and Qviller linked to in [12], this implementation was obtained from http://sourceforge.net/projects/toricsegregclass/. Also note that the run times we give for the algorithm of Moe and Qviller [12] in Table 4.1 are likely less than the total run time of their algorithm since their implementation is broken in to two
parts; one part runs in Macaulay2 [9] and the second part runs in Sage [14]. We only give the running time for the Macaulay2 [9] component of their algorithm and do not add in the extra time to run the second part in Sage which would be needed to actually obtain the desired result using Moe and Qviller’s implementation; this is described in [12].

Note that the equations defining the test cases used in all tables appearing in this chapter can be found in Appendix A.5.

As was noted previously when doing performance testing in Chapters 2 and 3, the $c_{SM}$ class is, technically, only defined when working over fields of characteristic zero (see, for example, Aluffi [4] for further discussion). However, as was the case in previous chapters, since the result of the computation is the same when working over $\mathbb{Q}$ and over a finite field of large prime characteristic on all examples considered we give the run times over the finite field with 32749 elements for symbolic computations. This approach is also used for example computations of characteristic classes by Aluffi [3] and Jost [10], as well as by Eklund, Jost and Peterson [5].

As can be seen in Table 4.1 Algorithm 4.3.1 is consistently and often quite considerably faster than the algorithm of Moe and Qviller [12]. We note that the method of Moe and Qviller is valid in more generality, i.e. it is applicable for any subscheme of a smooth projective toric variety, and does not make any attempt to pre-process the Chow ring structure. Because of this the algorithm of [12] does many more degree computations than necessary, particularly in the case where one is considering subschemes of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ as we do here. This may at least partially explain the slow running times of the algorithm of [12] for the examples in Table 4.1.

In Table 4.2 we give the running times to compute the $c_{SM}$ class and/or Euler characteristic using Algorithm 4.3.2, our algorithm to compute the $c_{SM}$ class and/or Euler characteristic of a subscheme of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ using inclusion/exclusion. Because there are no other known algorithms to compute the $c_{SM}$ class and Euler characteristic in this setting there are no other methods to compare to.
Table 4.1: Run time comparison of different algorithms for computing the Segre class of a projective variety. We use - to denote computations that were stopped after ten minutes (600 s). Computations were performed over $\mathbb{F}(32749)$ on a computer with a 2.40GHz Intel Core i5-450M CPU and 4 GB of RAM.

Let $V = V(f_0, f_1, f_2) \subset \mathbb{P}^2 \times \mathbb{P}^2 \times \mathbb{P}^3$ be the example from Table 4.2 which has codimension 3 in $\mathbb{P}^2 \times \mathbb{P}^2 \times \mathbb{P}^3$ with degree $(2, 1, 0), (0, 1, 2), (1, 2, 0)$ equations. For this example (and for the other examples as well) the majority of the running time is spent computing the $c_{SM}$ class of the hypersurface of largest degree appearing in the inclusion/exclusion procedure. In this case that is the class $c_{SM}(V(f_0 \cdot f_1 \cdot f_2))$ and around 85% of the total computation time is spent computing this class. To compute this class in practice using Algorithm 4.3.2 we must find the projective multi-degrees associated to the ideal defining the singularity subscheme of $V(f_0 \cdot f_1 \cdot f_2)$. To find all these projective multi-degrees we must, essentially, solve 35 different zero dimensional polynomial systems in 11 dimensional affine space each containing equations which have degrees of up to 10. The 35 zero dimensional systems we consider in this example have 2, 3, 3, 6, 6, 6, 9, 6, 4, 9, 9, 12, 18, 12, 18, 12, 12, 18, 27, 18, 36, 36, 36, 24, 36, 24, 54, 54, 72, 72, 72, 108, and 144 solutions, respectively.

In Table 4.3 we compare the running times of Algorithm 4.3.3, our direct algorithm to compute the $c_{SM}$ class and Euler characteristic using Theorem 4.2.3, to the running time of Algorithm 4.3.2, our algorithm using inclusion/exclusion in the multi-projective setting. Note that Algorithm 4.3.3 is only valid when the input ideal $f_0, \ldots, f_r$ contains some $f_0, \ldots, f_{r-1}$ such that $V(f_0, \ldots, f_{r-1})$ is smooth. For
Table 4.2: Running times for Algorithm 4.3.2 to compute the $c_{SM}$ class and Euler characteristic of a subscheme of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$. Computations were performed over $\mathbb{GF}(32749)$ on a computer with a 2.9GHz Intel Core i7-3520M CPU and 8 GB of RAM.

<table>
<thead>
<tr>
<th>Input</th>
<th>Algorithm 4.3.2</th>
<th>Algorithm 4.3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Codimension 2 in $\mathbb{P}^2 \times \mathbb{P}^2$</td>
<td>0.3s</td>
<td></td>
</tr>
<tr>
<td>Codimension 2 in $\mathbb{P}^6 \times \mathbb{P}^2$ with degree (3, 0), (0, 2) eqs.</td>
<td>3.9s</td>
<td></td>
</tr>
<tr>
<td>Codimension 2 in $\mathbb{P}^5 \times \mathbb{P}^3$ with degree (2, 1) and (1, 1) eqs.</td>
<td>12.4s</td>
<td></td>
</tr>
<tr>
<td>Codimension 2 in $\mathbb{P}^2 \times \mathbb{P}^2 \times \mathbb{P}^3$ with degree (2, 1, 0) and (0, 1, 2) eqs.</td>
<td>4.8s</td>
<td></td>
</tr>
<tr>
<td>Codimension 3 in $\mathbb{P}^2 \times \mathbb{P}^2 \times \mathbb{P}^3$ with degree (2, 1, 0), (0, 1, 2), (1, 2, 0) eqs.</td>
<td>52.4s</td>
<td></td>
</tr>
</tbody>
</table>

development of the examples in the table Algorithm 4.3.3 does indeed provide a performance improvement. Note that the running time of Algorithm 4.3.3 includes the time required to compute the singularity subscheme, which is often a considerable percentage of the overall run time of the algorithm particularly in larger dimension. For the singularity subscheme computation we saturate by the irrelevant ideal, which can be a difficult computation for products of many projective spaces as the structure of the irrelevant ideal gets increasingly complicated. As such a more efficient way to compute the singularity subscheme than that presented in Algorithm 4.3.3 could result in a more marked performance gain versus the inclusion/exclusion method.

Table 4.3: Running times for Algorithm 4.3.3, our direct algorithm to compute the $c_{SM}$ class and Euler characteristic of a subscheme of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ which satisfies the assumptions of Theorem 4.2.3. These running times are compared to the running times of Algorithm 4.3.2. Computations performed over $\mathbb{GF}(32749)$ on a computer with a 2.9GHz Intel Core i7-3520M CPU and 8 GB of RAM.

<table>
<thead>
<tr>
<th>Input</th>
<th>Algorithm 4.3.2</th>
<th>Algorithm 4.3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Codimension 3 in $\mathbb{P}^2 \times \mathbb{P}^2$</td>
<td>1.6s</td>
<td>0.3s</td>
</tr>
<tr>
<td>Codimension 2 in $\mathbb{P}^2 \times \mathbb{P}^3$</td>
<td>1.9s</td>
<td>1.0s</td>
</tr>
<tr>
<td>Codimension 3 in $\mathbb{P}^2 \times \mathbb{P}^2 \times \mathbb{P}^2$</td>
<td>5.7s</td>
<td>0.2s</td>
</tr>
<tr>
<td>Codimension 2 in $\mathbb{P}^3 \times \mathbb{P}^2 \times \mathbb{P}^2$</td>
<td>3.1s</td>
<td>0.9s</td>
</tr>
</tbody>
</table>
4.4.2 Running Time Bounds

We now consider running time bounds for Algorithms 4.3.1 and 4.3.2. Suppose we are working with a multi-homogeneous ideal $I = (f_0, \ldots, f_r)$ in the coordinate ring $R$ of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ defining a $g$-dimensional scheme $V = V(I)$. Further assume, without loss of generality, that $\text{deg}(f_i) = (d_1, \ldots, d_m)$ for all $i = 0, \ldots, r$.

Throughout this subsection let $\delta(D, N)$ be the number of arithmetic operations required to find the number of points in a zero dimensional affine variety $W$ defined by a polynomial system containing $N$ degree $D$ polynomials in $N$ variables. Using the algorithm of Lecerf [11] (given as Theorem 6.3.2 in Chapter 5) or the algorithm of Giusti, Lecerf and Salvy [8] we have that the number of arithmetic operations to solve such a system is polynomial in $O(N^5 D^{3N})$.

Note that the $D^N$ term in the $O(N^5 D^{3N})$ complexity bound on the algorithms of Lecerf [11] and of Giusti, Lecerf and Salvy [8] is obtained by using a standard Bézout bound in a projective space. As such the estimate $O(N^5 D^{3N})$ could be sharpened in some cases, in the context of our algorithms in this chapter, by using the results of Chapter 6 which would take into account the multi-projective structure of the input.

In practice the current implementations of Algorithms 4.3.1 and 4.3.2 use the Gröbner basis algorithms built into Macualay2 [9], so the running time bounds will be different. For this reason we present the complexity results in terms of the complexity of solving zero dimensional polynomial systems.

**Proposition 4.4.1.** Let $I = (f_0, \ldots, f_r)$ be a multi-homogeneous ideal in the coordinate ring $R$ of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ defining a $g$-dimensional scheme $V = V(I)$ and let $n = n_1 + \cdots + n_m$. Further assume, without loss of generality, that $\text{deg}(f_i) = (d_1, \ldots, d_m)$ for all $i = 0, \ldots, r$. We have that the number of arithmetic operations required to compute the Segre class $s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})$ using Algorithm 4.3.1 is of order

$$O\left(\delta(d_1 + \cdots + d_m + 1, n + 2) \cdot \sum_{\ell = \text{codim}(V)}^{n} \binom{m + \ell - 1}{\ell}\right).$$
Proof. Consider the Chow group $A^\iota(\mathbb{P}^n_1 \times \cdots \times \mathbb{P}^n_m)$, ignoring the cases where $\iota > n_i$ for some $i$ there will be
\[\binom{m + \iota - 1}{\iota}\]
monomials in the basis of $A^\iota(\mathbb{P}^n_1 \times \cdots \times \mathbb{P}^n_m)$ since this is the number of $m$-tuples of non-negative integers with sum $\iota$. When $\iota > n_i$ then $\iota - n_i$ of these terms will be zero for each such $i$, hence we solve a zero dimensional system at most
\[\sum_{\iota = \text{codim}(V)}^n \binom{m + \iota - 1}{\iota}\]
times to compute the Segre class. Note that the largest possible total degree of equations considered in Algorithm 4.3.1 is $d_1 + \cdots + d_m + 1$. □

Examining Algorithm 4.3.2 we note that one Segre class, namely that of the appropriate singularity subscheme, must be calculated for each subset of the generators of $I$ when finding $c_{SM}(V(I))$.

Corollary 4.4.2. Let $I = (f_0, \ldots, f_r)$ be a multi-homogeneous ideal in the coordinate ring $R$ of $\mathbb{P}^n_1 \times \cdots \times \mathbb{P}^n_m$ defining a $\rho$-dimensional scheme $V = V(I)$ and let $n = n_1 + \cdots + n_m$. Further assume, without loss of generality, that $\deg(f_i) = (d_1, \ldots, d_m)$ for all $i = 0, \ldots, r$. Further let $\kappa$ be the minimum codimension of the singularity subscheme of all hypersurfaces of all products of the generators of $I$. The number of arithmetic operations required to compute $c_{SM}(V)$ using Algorithm 4.3.2 has order
\[O\left(2^{r+1} \cdot \delta((r + 1)(d_1 + \cdots + d_m + 1), n + 2) \cdot \sum_{\iota = \kappa}^n \binom{m + \iota - 1}{\iota}\right),\]
where $\delta$ is as in Proposition 4.4.1.

Proof. There are $2^{r+1}$ subsets of $\{f_0, \ldots, f_r\}$. The maximum total degree of elements in the Jacobian ideal of $f_0 \cdots f_r$ will be $(r + 1)(d_1 + \cdots + d_m)$. □
Bibliography


Chapter 5

$c_{SM}$ Classes of Complete Simplicial Toric Varieties

In this chapter we present Algorithm 5.3.1 which computes the Chern-Schwartz-MacPherson class and Euler characteristic and Algorithm 5.3.2 which computes only the Euler characteristic of a complete simplicial toric variety $X_\Sigma$ defined by a fan $\Sigma$. These algorithms are based on a result of Barthel, Brasselet and Fieseler [1] which gives an expression for the $c_{SM}$ class of a toric variety in terms of torus orbit closures. Note that we will only consider toric varieties $X_\Sigma$ over $\mathbb{C}$.

We begin by reviewing necessary background on the construction of toric varieties from a fan and their properties in Section 5.1. In Section 5.2 we review properties of the Chow rings of toric varieties which will be used in Algorithm 5.3.1. In Section 5.3 we present Algorithm 5.3.1 which computes the $c_{SM}$ class and Euler characteristic and Algorithm 5.3.2 which computes only the Euler characteristic of a complete simplicial toric variety $X_\Sigma$ defined by a fan $\Sigma$ and give several examples of its use. Algorithm 5.3.2 will offer improved performance in comparison to Algorithm 5.3.1 when one wishes to compute only the Euler characteristic.

In §5.4 we test the run times of Algorithm 5.3.1 and Algorithm 5.3.2 on some examples. Note that both algorithms are strictly combinatorial and hence the runtimes do not depend on the degree of the equations defining the toric variety in any way.
Indeed the computations use only the combinatorial data coming from the fan defining the toric variety to find the $c_{S, M}$ class. The runtime results of Algorithm 5.3.1 and Algorithm 5.3.2 applied to a selection of examples are presented in Table 5.1 below; from this testing it seems that we can expect even reasonably large examples to finish in an acceptable amount of time.

We also note that the restriction to complete simplicial toric varieties is not required in the statement of the result of Barthel, Brasselet and Fieseler [1] on which our algorithm is based, indeed these restrictions are present on the algorithm only for the purpose of simplifying the construction of the Chow ring of the toric variety. If one was able to construct the Chow ring in a simple manner with the restrictions removed the algorithm could be applied unchanged in this more general setting.

The Macaulay2 [4] implementation of the algorithms for computing the $c_{S, M}$ class and Euler characteristic of a complete simplicial toric variety presented in this chapter can be found at https://github.com/Martin-Helmer/char-class-calc. This implementation is accessed via the “CharToric” package, see Example 5.3.4 and Appendix A.6 for the package syntax.

5.1 Review

In this section we review some definitions and results regarding toric varieties which will be needed in later sections. For our purposes we will call a variety $X$ a toric variety if $X$ is a normal variety which contains a torus $T = (\mathbb{C}^*)^n$ as an open dense set (in the Zariski topology) together with an action $T \times X \rightarrow X$ of $T$ on $X$ which extends the usual multiplication in $T$.

5.1.1 Constructing the Toric Variety of a Fan

Often we will wish to define the toric variety using a lattice $N$ (which is isomorphic to $\mathbb{Z}^n$ for some $n$) and a fan $\Sigma$ in $N$ which is a collection of strongly convex rational
polyhedral cones in the real vector space \( N_\mathbb{R} = N \otimes \mathbb{Z} \mathbb{R} \). To this end we recall some terminology, a more complete overview can be found in Cox, Little and Schenck [2] or Fulton [3].

The rational polyhedral cone \( \sigma \) associated to \( \{v_1, \ldots, v_l\} \subset N \) is

\[
\sigma = \langle v_1, \ldots, v_l \rangle = \left\{ \sum_{i=1}^l \lambda_i v_i \bigg| \lambda_i \geq 0 \right\} \subseteq N_\mathbb{R}.
\] (5.1)

Such a cone will be termed strongly convex if it contains no line through the origin, i.e. if \( v \in \sigma \) then \( -v \not\in \sigma \).

For a rational polyhedral cone \( \sigma \) we let \( \text{Span}(\sigma) \) denote the smallest subspace of the vector space \( N_\mathbb{R} \) which contains \( \sigma \). The dimension of \( \sigma \) is denoted \( \dim(\sigma) \) and is defined to be the dimension of \( \text{Span}(\sigma) \). The one dimensional cone \( \rho \) generated by one element \( v \in N \), i.e. \( \rho = \langle v \rangle \), will be referred to as a ray.

Let \( M = \text{Hom}(N, \mathbb{Z}) \) denote the dual lattice of \( N \), also let \( M_\mathbb{R} = M \otimes \mathbb{Z} \mathbb{R} \). For a cone \( \sigma \subset N_\mathbb{R} \) define the dual cone \( \sigma^\vee \subset M_\mathbb{R} \) as the set of vectors nonnegative on \( \sigma \), that is

\[
\sigma^\vee = \left\{ u \in M_\mathbb{R} \mid u(v) \geq 0, \ \forall v \in \sigma \right\}.
\] (5.2)

For any dual vector \( u \in M_\mathbb{R} \) we may also define

\[
u^\perp = \left\{ v \in N_\mathbb{R} \mid u(v) = 0 \right\},
\] (5.3)

which will allow us to define a face \( \tau \) of a cone \( \sigma \) by choosing \( u \in \sigma^\vee \) and setting

\[
\tau = \sigma \cap u^\perp = \left\{ v \in \sigma \mid u(v) = 0 \right\}.
\]

We may now define a fan more explicitly. A fan \( \Sigma \) is a collection of strongly convex rational polyhedral cones such that:

- if \( \tau \) is a face of a cone \( \sigma \in \Sigma \) then \( \tau \) is a cone in \( \Sigma \)
- if \( \sigma_1, \sigma_2 \in \Sigma \) are cones then \( \sigma_1 \cap \sigma_2 \) is a face of both \( \sigma_1 \) and \( \sigma_2 \).
This second condition gives us a way to glue cones, specifically a face where two cones meet can be thought of as gluing the two cones together. For the remainder of this chapter by cone we shall mean a strongly convex rational polyhedral cone.

Given a cone \( \sigma \) we may define a semigroup \( S \) by

\[
S_\sigma = \sigma^\vee \cap M = \{ u \in M \mid u(v) \geq 0 \ \forall v \in \sigma \}
\]

(5.4)

with the group operation being vector addition. This group is finitely generated (Gordon’s Lemma, see Proposition 1.2.17 of [2] for example).

To construct an affine variety from a cone \( \sigma \) we first construct the affine coordinate ring \( \mathbb{C}[S_\sigma] \) associated to \( \sigma \). We may write \( \chi^u \) for the element of the \( \mathbb{C} \)-algebra \( \mathbb{C}[S_\sigma] \) corresponding to the semigroup element \( u \in S_\sigma \), for \( u_1 + u_2 \in S_\sigma \) we require that \( \chi^{u_1} \chi^{u_2} = \chi^{u_1+u_2} \). Each element of \( \mathbb{C}[S_\sigma] \) is expressed as a finite sum \( \sum c_i \chi^{u_i} \) for \( c_i \in \mathbb{C} \) and \( u_i \in S_\sigma \). We can now define an affine variety \( U_\sigma \) associated to \( \sigma \) as

\[
U_\sigma = \text{Spec}(\mathbb{C}[S_\sigma])
\]

with coordinate ring \( \mathbb{C}[S_\sigma] \). Further we can think of the maximal ideals of \( \mathbb{C}[S_\sigma] \) as points in \( \mathbb{C}^n \) when our lattice is \( \mathbb{N} \approx \mathbb{Z}^n \).

Stated more formally we have the following:

**Theorem 5.1.1** (Theorem 1.2.18 of [2]). Let \( \sigma \subset \mathbb{N}_\mathbb{R} \approx \mathbb{R}^n \) be a strongly convex rational polyhedral cone with semigroup \( S_\sigma = \sigma^\vee \cap M \). Then

\[
U_\sigma = \text{Spec}(\mathbb{C}[S_\sigma])
\]

(5.5)

is an affine variety with \( \dim(U_\sigma) = n \), further \( U_\sigma \) contains the torus \( T_N = \mathbb{N} \otimes \mathbb{Z} \mathbb{C}^* \approx (\mathbb{C}^*)^n \) as an open dense set (in the Zariski topology) together with an action of the torus \( T_N \) on \( U_\sigma \).

To construct a variety \( X_\Sigma \) from a fan \( \Sigma \) we will need a way to patch together these affine varieties associated to cones, this is given in the following proposition. In this
proposition we shall use the terminology of Fulton [3]; in particular by a principal open set we shall mean the complement of a Zariski closed set.

**Proposition 5.1.2** (Lemma in §1.3 of [3]). If \( \tau \) is a face of a cone \( \sigma \) then the map \( U_{\tau} \to U_{\sigma} \) embeds \( U_{\tau} \) as a principal open set in \( U_{\sigma} \). In particular if two cones \( \sigma_1, \sigma_2 \) intersect in a common face \( \tau = \sigma_1 \cap \sigma_2 \) then \( U_{\tau} \) embeds as a principal open set in \( U_{\sigma_1} \) and in \( U_{\sigma_2} \).

For a fan \( \Sigma \) this proposition gives us a way of patching together the affine varieties \( U_{\sigma} \) for \( \sigma \in \Sigma \) to form an algebraic variety \( X_{\Sigma} \). Specifically \( X_{\Sigma} \) is the disjoint union of the \( U_{\sigma} \), that is

\[
X_{\Sigma} = \bigsqcup_{\sigma \in \Sigma} U_{\sigma} \tag{5.6}
\]

with \( U_{\sigma_1} \) and \( U_{\sigma_2} \) glued together by identifying \( U_{\sigma_1 \cap \sigma_2} \) as a principal open subset of \( U_{\sigma_1} \) and of \( U_{\sigma_2} \). In more detail, since any two cones \( \sigma_1, \sigma_2 \) in a fan \( \Sigma \) share a face \( \sigma_1 \cap \sigma_2 \) then from Proposition 5.1.2 we have an injection \( \phi : U_{\sigma_1 \cap \sigma_2} \to U_{\sigma_1} \) and an injection \( \theta : U_{\sigma_1 \cap \sigma_2} \to U_{\sigma_2} \) so that we have a map

\[
f : \phi(U_{\sigma_1 \cap \sigma_2}) \to \theta(U_{\sigma_1 \cap \sigma_2})
\]

specified by

\[
f : x \mapsto \theta(\phi^{-1}(x))
\]

for \( x \in U_{\sigma_1} = \phi(U_{\sigma_1 \cap \sigma_2}) \) with the inverse specified by \( f^{-1} : y \mapsto \phi(\theta^{-1}(y)) \). The map \( f \) patches together \( U_{\sigma_1} \) and \( U_{\sigma_2} \) for any two cones in \( \sigma_1, \sigma_2 \in \Sigma \).

For a fan \( \Sigma \) the set of rays (that is one dimensional cones \( \rho = \langle v \rangle, v \in N \)) will be denoted \( \Sigma(1) \). Note that any cone \( \sigma \in \Sigma \) can be constructed from a set of rays in \( \Sigma(1) \). Further if a cone \( \sigma = \langle v_1, \ldots, v_d \rangle \) is constructed of the rays \( \rho_1 = \langle v_1 \rangle, \ldots, \rho_d = \langle v_d \rangle \) we will write \( \sigma = \rho_1 + \cdots + \rho_d \).

Examples of toric varieties include projective spaces and products of projective spaces. In the example below we detail how \( \mathbb{P}^2 \) can be defined as the toric variety of a fan.

**Example 5.1.3** (Constructing \( \mathbb{P}^2 \) as the toric variety of a fan). Let \( \Sigma \) be the fan
defined by the three cones

\[ \sigma_0 = \langle e_1, e_2 \rangle \]
\[ \sigma_1 = \langle e_2, -e_1 - e_2 \rangle \]
\[ \sigma_2 = \langle e_1, -e_1 - e_2 \rangle \]

and their faces where \( e_1 = (1, 0), e_2 = (0, 1) \) are the standard basis vectors in \( \mathbb{Z}^2 \).

Note that \( \Sigma(1) = \{ \langle e_1 \rangle, \langle e_2 \rangle, \langle -e_1 - e_2 \rangle \} \) with \( \sigma_0 = \langle e_1 \rangle + \langle e_2 \rangle \) and so on. Note also that there are seven cones total in \( \Sigma \), i.e. \( \sigma_0, \sigma_1, \sigma_2 \) and their four faces, three of which are the rays in \( \Sigma(1) \) (given by the faces \( \sigma_i \cap \sigma_j \) for \( i \neq j \)) and the fourth face being \( \sigma_0 \cap \sigma_1 \cap \sigma_2 = \{ 0 \} \).

First compute the \( S_{\sigma_i} = \sigma_i^\vee \cap M \). Some \( u \in M_{\mathbb{Z}} \) will have the form \( u = ae_1^\vee + be_2^\vee \) and \( v \in \sigma_0 \) will have the form \( ce_1 + de_2 \) for \( a, b, c, d \in \mathbb{R} \). Consider \( S_{\sigma_0} = \sigma_0^\vee \cap M \), we know that for \( u \in S_{\sigma_0} \) we have \( u(v) \geq 0 \) for all \( v \in \sigma_0 \), expanding \( u(v) \) we obtain:

\[
u(v) = (ae_1^\vee + be_2^\vee)(ce_1 + de_2) = ae_1^\vee (ce_1 + de_2) + be_2^\vee (ce_1 + de_2) = ac(e_1e_1^\vee) + bd(e_2e_2^\vee)\]

Since \( \sigma_0 \) is a strongly convex rational polyhedral cone we know that \( c \geq 0 \) and \( d \geq 0 \), so to have \( u(v) = ac(e_1)(e_1^\vee) + bd(e_2)(e_2^\vee) \geq 0 \) we must have \( a \geq 0 \) and \( b \geq 0 \) for \( u \in \sigma_0^\vee \cap M \), from this we can conclude that \( \sigma_0^\vee \) is generated by \( e_1^\vee \) and \( e_2^\vee \). Hence we may represent elements of \( \mathbb{C}[S_{\sigma_0}] \) as \( \chi^{ae_1^\vee + be_2^\vee} \) \((a, b \in \mathbb{Z})\) with multiplication in \( \mathbb{C}[S_{\sigma_0}] \) adding exponents in \( S_{\sigma} \). If we set \( x = \chi^{e_1^\vee} \) and \( y = \chi^{e_2^\vee} \) we may then write \( \mathbb{C}[S_{\sigma_0}] = \mathbb{C}[x, y] \). With this identification we see that \( U_{\sigma_0} \) is the affine variety with coordinate ring \( \mathbb{C}[x, y] \), i.e. \( U_{\sigma_0} = \text{Spec}(\mathbb{C}[x, y]) \cong \mathbb{C}^2 \).

Similarly we find that \( \mathbb{C}[S_{\sigma_1}] = \mathbb{C}[x^{-1}, yx^{-1}] \) and \( \mathbb{C}[S_{\sigma_2}] = \mathbb{C}[xy^{-1}, y^{-1}] \), so \( \mathbb{C}[S_{\sigma_0}] \cong \mathbb{C}[S_{\sigma_1}] \cong \mathbb{C}[S_{\sigma_2}] \cong \mathbb{C}[x, y] \) and hence \( U_i \cong \mathbb{C}^2 \) for all \( i \). Taking \( \mathbb{P}^2 = \text{Proj}(\mathbb{C}[z_0, z_1, z_2]) \) we may identify \( x = \frac{z_1}{z_0} \) and \( y = \frac{z_2}{z_0} \) giving us that \( U_{\sigma_i} = \text{Spec}(\mathbb{C}[S_{\sigma_i}]) \) is the principal open set \( V_i = \mathbb{P}^2 \setminus V(z_i) \), we may then glue the affine varieties in the usual manner.

More specifically for the face \( \sigma_0 \cap \sigma_1 = \langle e_2 \rangle \) the gluing of \( U_{\sigma_0} \) and \( U_{\sigma_1} \) on \( U_{(e_1)} \)
is realized by sending \((x, y) \mapsto (x^{-1}, yx^{-1})\) or equivalently, if we let \(a_0, a_1, a_2\) be homogeneous coordinates in \(\mathbb{P}^2\), by setting
\[
\left( \frac{1}{a_1} : \frac{a_2}{a_0} \right) = \left( \frac{a_0}{a_1} : 1 : \frac{a_2}{a_1} \right),
\]
on \(V_0 \cap V_1 = U_{(e_2)}\). The glueing proceeds in a similar fashion for the remaining faces, thus we have \(\mathbb{P}^2 = X_\Sigma\).

Since the implementations of the algorithms given in this chapter are done using the “NormalToricVarieties” package in the Macaulay2 [4] computer algebra system we note that we can define \(\mathbb{P}^2\) as a toric variety of a fan in Macaulay2 as follows:

\[
\begin{align*}
\text{Rho} &= \{\{1,0\}, \{0,1\}, \{-1,-1\}\} \\
\text{Sig} &= \{\{0,1\}, \{0,2\}, \{1,2\}\} \\
\text{PP2} &= \text{normalToricVariety}(\text{Rho}, \text{Sig})
\end{align*}
\]

Here \(\text{Rho}\) is the list of rays \(\Sigma(1)\) and \(\text{Sig}\) is the list of listes of indexes of rays in \(\text{Rho}\) which define the cones \(\sigma_0, \sigma_1, \sigma_2\), i.e. the first list \(\{0,1\}\) in \(\text{Sig}\) indicates we choose the cone defined by the position zero and position one rays of \(\text{Rho}\) which is the cone \(\sigma_0\) defined by \(e_1, e_2\) and so on.

### 5.1.2 Orbits and Orbit Closures

In this subsection we briefly review the definition of the orbit closure of a torus orbit corresponding to a cone. We will make reference to these objects when constructing Algorithm 5.3.1, for a more complete review we recommend §3.2 of Cox, Little and Schenck [2].

Let \(X_\Sigma\) be the toric variety of the fan \(\Sigma\) in \(N_\mathbb{R}\). Since the torus \(T_N = N \otimes_{\mathbb{Z}} \mathbb{C}^*\) acts on \(X_\Sigma\) one may define the orbits \(O(\sigma)\) for each cone \(\sigma \in \Sigma\), specifically we have the following theorem, often called the Orbit-Cone Correspondence.

**Theorem 5.1.4** (Theorem 3.2.6 of Cox, Little and Schenck [2]). *Let \(X_\Sigma\) be the toric variety of the fan \(\Sigma\) in \(N_\mathbb{R}\). We have the following:*
• There is a bijective correspondence

\[ \{ \text{cones } \sigma \in \Sigma \} \leftrightarrow \{ T_N - \text{orbits in } X_\Sigma \}, \]
\[ \sigma \leftrightarrow O(\sigma) \cong \text{Hom}_\mathbb{Z}(\sigma^\perp \cap M, \mathbb{C}^*). \]

• Let \( n = \dim(N_\mathbb{R}). \) For each cone \( \sigma \in \Sigma, \) \( \dim(O(\sigma)) = n - \dim(\sigma). \)

For simplicity we will take the following proposition as the definition of the orbit closure \( V(\sigma) = \overline{O(\sigma)} \subset X_\Sigma \) corresponding to a cone \( \sigma \in \Sigma. \)

**Proposition 5.1.5** (Proposition 3.2.7. of Cox, Little and Schenck [2]). Let \( \tau \) be a cone in \( \Sigma \) and let \( N_\tau \) be the sublattice of \( N \) spanned by the points in \( \tau \cap N. \) Let \( N(\tau) = N/N_\tau, \) additionally for each cone \( \sigma \in \Sigma \) containing \( \tau \) let \( \overline{\sigma} \) be the image cone in \( N(\tau)_\mathbb{R} \) under the homomorphism \( N_\mathbb{R} \rightarrow N(\tau)_\mathbb{R}. \) Then

\[ \text{Star}(\tau) = \{ \overline{\sigma} \subset N(\tau)_\mathbb{R} \mid \text{for all } \sigma \in \Sigma \text{ such that } \tau \text{ is a face of } \sigma \} \]

is a fan in \( N(\tau)_\mathbb{R} \) and for any \( \tau \in \Sigma \) we have that the orbit closure \( V(\tau) = \overline{O(\tau)} \) of \( \tau \) in \( X_\Sigma \) is a subvariety of \( X_\Sigma \) which is isomorphic to the toric variety \( X_{\text{Star}(\tau)}. \)

### 5.1.3 Complete Simplicial Toric Varieties

Recall that a variety \( X \) is *complete* if for every variety \( Z \) the projection map \( \pi_Z : X \times Z \rightarrow Z \) is a closed mapping in the Zariski topology. By Theorem 3.4.6 of [2] a toric variety \( X_\Sigma \) of a fan \( \Sigma \) is complete if the fan \( \Sigma \) is complete, i.e. if

\[ \bigcup_{\sigma \in \Sigma} \sigma = N_\mathbb{R}. \]

We now briefly discuss singularities of toric varieties following §11.4 of Cox, Little and Schenck [2]. Consider a variety \( X \) with structure sheaf \( O_X, \) the structure sheaf \( O_X \) is a sheaf in the Zariski topology. If one switches to the classical topology the sheaf corresponding to \( O_X \) is the sheaf of analytic functions on \( X, \) denoted \( O_X^{\text{an}}. \)
With this notation any open set $U \subset X$ in the classical topology will give an analytic variety $(U, O^a_{U}) = (U, O^a_{X|U})$. Let $X_1, X_2$ be varieties, points $p_1 \in X_1$ and $p_2 \in X_2$ are termed *locally analytically equivalent* if there are neighbourhoods $p_1 \in U_1 \subset X_1$ and $p_2 \in U_2 \subset X_2$ in the classical topology such that $U_1$ is isomorphic to $U_2$ as an analytic variety and this isomorphism takes $p_1$ to $p_2$.

Let $X$ be an irreducible variety of dimension $n$, a point $p \in X$ is a *finite quotient singularity* if there is a finite subgroup $G \subset \text{GL}(n, \mathbb{C})$ such that $p \in X$ is locally analytically equivalent to $0 \in \mathbb{C}^n \setminus G$. We call a variety $X$ *orbifold* or *quasismooth* or say $X$ has only finite quotient singularities if every point $p \in X$ is a finite quotient singularity. Note that in particular smooth toric varieties are orbifold.

A cone $\sigma$ is *simplicial* if its minimal generators are linearly independent over $\mathbb{R}$. A fan $\Sigma$ is simplicial if every cone $\sigma \in \Sigma$ contained in the fan is simplicial. The toric variety $X_\Sigma$ of the fan $\Sigma$ is called simplicial if the fan $\Sigma$ is simplicial.

**Proposition 5.1.6** (Theorem 3.1.19 of [2]). A toric variety $X_\Sigma$ is orbifold, that is has only finite quotient singularities, if and only if $\Sigma$ is simplicial.

For our purposes in this chapter we are interested in toric varieties which are orbifold primarily because these varieties are also what is called rationally smooth (see §12.4 of [2]) and hence, roughly speaking, their cohomology rings behave like the cohomology rings of smooth varieties, except we work over $\mathbb{Q}$ rather than $\mathbb{Z}$.

## 5.2 The Chow Ring of a Complete Simplicial Toric Variety

Let $X_\Sigma$ be an $n$-dimensional complete and simplicial toric variety defined by a fan $\Sigma$. Similar to the construction of the Chow ring in the smooth case given in §2.1.1 above we may construct the Chow ring of $X_\Sigma$ from the Chow groups, that is the groups $A^j(X_\Sigma)$ of codimension $j$-cycles on $X_\Sigma$ modulo rational equivalence. The only difference in this case will be that we work over the rational number field $\mathbb{Q}$.
rather than the integers. Since $X_\Sigma$ is a $n$-dimensional complete and simplicial toric variety the intersection product can be defined on rational cycles (see §12.5 of [2]) so that we have the rational Chow ring of $X_\Sigma$ given by the graded ring

$$A^*(X_\Sigma) \otimes \mathbb{Q} = \bigoplus_{j=0}^{n} A^j(X_\Sigma) \otimes \mathbb{Q}. \quad (5.7)$$

For each cone $\sigma$ in the fan $\Sigma$ the orbit closure $V(\sigma)$ is a subvariety of codimension $\dim(\sigma)$. We will write $[V(\sigma)]$ for the rational equivalence class of $V(\sigma)$ in $A^\dim(\sigma)(X_\Sigma)$.

**Proposition 5.2.1** (Lemma 12.5.1 of [2]). The collections $[V(\sigma)] \in A_j(X_\Sigma)$ for $\sigma \in \Sigma$ having dimension $n - j$ generate $A_j(X_\Sigma)$, the Chow group of dimension $j$. Further the collection $[V(\sigma)]$ for all $\sigma \in \Sigma$ generates $A^*(X_\Sigma)$ as an abelian group.

The following proposition gives us a simple method to compute the rational Chow ring of a complete, simplicial toric variety $X_\Sigma$. We will use this result to compute the rational Chow ring $A^*(X_\Sigma) \otimes \mathbb{Q}$ in Algorithms 5.3.1 and 5.3.2, our algorithms to compute the $c_{SM}$ class and Euler characteristic of a complete, simplicial toric variety.

**Proposition 5.2.2** (Theorem 12.5.3 of Cox, Little, Schenck [2]). Let $X_\Sigma$ be a complete and simplicial toric variety with generating rays $\Sigma(1) = \rho_1, \ldots, \rho_r$ where $\rho_j = \langle v_j \rangle$ for $v_j \in \mathbb{N}$. Then we have that

$$\mathbb{Q}[x_1, \ldots, x_r]/(I + J) \cong A^*(X_\Sigma) \otimes \mathbb{Q}, \quad (5.8)$$

with the isomorphism map specified by $[x_i] \mapsto [V(\rho_i)]$. Here $I$ denotes the Stanley-Reisner ideal of the fan $\Sigma$, that is the ideal in $\mathbb{Q}[x_1, \ldots, x_r]$ specified by

$$I = (x_{i_1} \cdots x_{i_s} \mid i_j \text{ distinct and } \rho_{i_1} + \cdots + \rho_{i_s} \text{ is not a cone of } \Sigma) \quad (5.9)$$

and $J$ denotes the ideal of $\mathbb{Q}[x_1, \ldots, x_r]$ generated by linear relations of the rays,
that is $\mathcal{J}$ is generated by linear forms

$$\sum_{j=1}^{r} m(v_j)x_j$$  \hspace{1cm} (5.10)

for $m$ ranging over some basis of $M$.

### 5.3 Algorithms to Compute the Chern-Schwartz-MacPherson Class and Euler Characteristic of a Complete Simplicial Toric Variety

In this section we present Algorithm 5.3.1 which computes the $c_{SM}$ class and Euler characteristic and Algorithm 5.3.2 which computes only the Euler characteristic of a complete simplicial toric variety defined by a fan $\Sigma$. Algorithm 5.3.2 will offer improved performance when only the Euler characteristic is desired. The main ingredient in these algorithms is the following result of Barthel, Brasselet and Fieseler [1].

**Proposition 5.3.1** (Main Theorem of Barthel, Brasselet and Fieseler [1]). Let $X_\Sigma$ be an $n$-dimensional complex toric variety specified by a fan $\Sigma$. We have that the Chern-Schwartz-MacPherson class of $X_\Sigma$ can be written in terms of orbit closures as

$$c_{SM}(X_\Sigma) = \sum_{\sigma \in \Sigma} [V(\sigma)] \in A^*(X_\Sigma)_\mathbb{Q}$$  \hspace{1cm} (5.11)

where $V(\sigma)$ is the closure of the torus orbit corresponding to $\sigma$.

We now recall the definition of the multiplicity of a simplicial cone, for more details see §6.4 of [2]. Let $\sigma = \langle v_1, \ldots, v_d \rangle$ be a simplicial cone and let

$$N_\sigma = \text{Span}(\sigma) \cap N,$$  \hspace{1cm} (5.12)

recall that $\text{Span}(\sigma) \subset N_\mathbb{R}$ is the smallest subspace of the vector space $N_\mathbb{R}$ which
contains $\sigma$. We note that the index of the subgroup $\mathbb{Z}v_1 + \cdots + \mathbb{Z}v_d \subset N_\sigma$ in $N_\sigma$ is finite. We define the multiplicity of $\sigma$ as

$$\text{mult}(\sigma) = [N_\sigma : \mathbb{Z}v_1 + \cdots + \mathbb{Z}v_d \subset N_\sigma]$$  \hspace{1cm} (5.13)$$

where $[G : H]$ denotes the index of a subgroup $H$ in a group $G$. In practice we shall employ Lemma 5.3.2 to compute $\text{mult}(\sigma)$.

Lemma 5.3.2 will allow us to compute the multiplicity of a simplicial cone. Since we only consider complete simplicial toric varieties in Algorithms 5.3.1 and 5.3.2 this lemma may be used to compute the multiplicity in all cases appearing in the algorithms. Lemma 5.3.2 is a modified version of Proposition 11.1.8. of Cox, Little, Schenck [2], we have slightly altered the statement of the result to explicitly show how we will compute these multiplicities in practice.

**Lemma 5.3.2** (Modified version of Proposition 11.1.8. of Cox, Little, Schenck [2]).

Let $N = \mathbb{Z}^n$ be an integer lattice. For a simplicial cone $\sigma = \rho_1 + \cdots + \rho_d \subset N$ let $\mathcal{M}_\sigma$ be the matrix with columns specified by the generating vectors of the rays $\rho_1, \ldots, \rho_d$ which define the cone $\sigma$; we have

$$\text{mult}(\sigma) = |\det(\text{Herm}(\mathcal{M}_\sigma))|$$  \hspace{1cm} (5.14)$$

where $\text{Herm}(\mathcal{M}_\sigma)$ denotes the Hermite normal form of matrix $\mathcal{M}_\sigma$ with all zero rows and/or zero columns removed.

*Further* $\text{mult}(\sigma) = 1$ if and only if $U_\sigma$ is smooth and if $\tau$ is a face of $\sigma$ $\text{mult}(\tau) \leq \text{mult}(\sigma)$.

**Proof.** Suppose $\rho_1 = \langle u_1 \rangle, \ldots, \rho_d = \langle u_d \rangle$ so that we can write $\sigma = \langle u_1, \ldots, u_d \rangle$. In Proposition 11.1.8. of Cox, Little, Schenck [2] it is shown that if $e_1, \ldots, e_d$ is a basis for $N_\sigma$ (see (5.12)) and $u_i = \sum_{j=1}^d a_{i,j} e_j = E[a_{i,j}]$ (where $E$ is the $n \times d$ matrix with columns $e_1, \ldots, e_d$) then we have that

$$\text{mult}(\sigma) = \left| \det \left( [a_{i,j}] \right) \right|.$$  \hspace{1cm} (5.15)$$
The matrix $M_\sigma$ defined by the rays $\rho_1, \ldots, \rho_d$ is the $n \times d$ matrix with columns given by the vectors $u_1, \ldots, u_d$. Note that $M_\sigma$ has rank $d$. Choose $e_1, \ldots, e_d$ to be a basis of $N_\sigma$ so that the matrix $E$ with columns $e_1, \ldots, e_d$ has the form

$$E = \begin{bmatrix} \tilde{E} \\ 0 \end{bmatrix}$$

with $\det(\tilde{E}) = 1$. Now since $M_\sigma$ has rank $d$ we may write

$$M_\sigma = \begin{bmatrix} \operatorname{Herm}(M_\sigma) \\ 0 \end{bmatrix} T$$

for $\operatorname{Herm}(M_\sigma)$ the $d \times d$ matrix obtained from the Hermite normal form of $M_\sigma$ with the zero rows removed and $T$ a $d \times d$ unimodular matrix. Then we have that

$$\begin{bmatrix} \tilde{E} \\ 0 \end{bmatrix} [a_{i,j}] = \begin{bmatrix} \operatorname{Herm}(M_\sigma) \\ 0 \end{bmatrix} T,$$

and hence $\tilde{E}[a_{i,j}] = \operatorname{Herm}(M_\sigma)T$. Note that $\det(\tilde{E}) = \det(T) = 1$, this gives that $\det([a_{i,j}]) = \det(\operatorname{Herm}(M_\sigma))$ as claimed.

The remaining statements are given in the form stated above in Proposition 11.1.8. of Cox, Little, Schenck [2].

To compute the classes $[V(\sigma)]$ appearing in (5.11) we will employ the following proposition combined with Proposition 5.2.2.

**Proposition 5.3.3** (Theorem 12.5.2. of Cox, Little, Schenck [2]). Assume that $X_\Sigma$ is complete and simplicial. If $\rho_1, \ldots, \rho_d \in \Sigma(1)$ are distinct and if $\sigma = \rho_1 + \cdots + \rho_d \in \Sigma$ then in $A^*(X_\Sigma)$ we have the following:

$$[V(\sigma)] = \operatorname{mult}(\sigma)[V(\rho_1)] \cdot [V(\rho_2)] \cdots [V(\rho_d)].$$

(5.16)

Here $\operatorname{mult}(\sigma)$ will be calculated using Lemma 5.3.2.

In Algorithm 5.3.1 we present an algorithm to compute $c_{SM}(X_\Sigma)$ for a complete,
simplicial toric variety $X_{\Sigma}$ defined by a fan $\Sigma$. Note that we represent $[V(\rho_j)]$ as $x_j$ using the isomorphism in Proposition 5.2.2.

Let $\omega_1, \ldots, \omega_t$ be a basis for $A_0(X_{\Sigma})$ (this can be computed either using Proposition 5.2.1 or by finding a monomial basis of the quotient ring presentation of Proposition 5.2.2 using standard methods). Then we may write the dimension zero component of $c_{SM}(X_{\Sigma})$ as

$$(c_{SM}(X_{\Sigma}))_0 = b_1 \omega_1 + \cdots + b_t \omega_t,$$

hence we have that the Euler characteristic of $X_{\Sigma}$ is given by

$$\chi(X_{\Sigma}) = \int c_{SM}(X_{\Sigma}) = b_1 + \cdots + b_t. \quad (5.17)$$

Algorithm 5.3.1. **Input:** A complete, simplicial toric variety $X_{\Sigma}$ defined by a fan $\Sigma$ with $\Sigma(1) = \{\rho_1, \ldots, \rho_r\}$.

**Output:** $c_{SM}(X_{\Sigma})$ in $A^*(X_{\Sigma}) \cong \mathbb{Q}[x_1, \ldots, x_r]/(I + J)$ and/or the Euler characteristic $\chi(X_{\Sigma})$.

- Compute the rational Chow ring $A^*(X_{\Sigma}) = \mathbb{Q}[x_1, \ldots, x_r]/(I + J)$ using Proposition 5.2.2.

- $csm = 0$.

- For $i = 1$ to $\dim(X_{\Sigma})$:  
  - orbits = all subsets of $\Sigma(1) = \{\rho_1, \ldots, \rho_r\}$ containing $i$ elements.
  - total = 0.

- For $\rho_{j_1}, \ldots, \rho_{j_i}$ in orbits:
  - $\sigma = \rho_{j_1} + \cdots + \rho_{j_i}$.
  - $\triangleright$ Find $w = \text{mult}(\sigma)$ using Lemma 5.3.2.
  - $[V(\sigma)] = \text{mult}(\sigma)[V(\rho_{j_1})] \cdots [V(\rho_{j_i})] = w \cdot x_{i_1} \cdots x_{i_s}$. 


\[ \text{total} = \text{total} + [V(\sigma)]. \]

- \[ \text{csm} = \text{csm} + \text{total}. \]

- Set \( c_{SM}(X_{\Sigma}) = \text{csm}. \)
- Set \( \chi(X_{\Sigma}) = \int c_{SM}(X_{\Sigma}). \)
- Return \( c_{SM}(X_{\Sigma}) \) and/or \( \chi(X_{\Sigma}). \)

Note that, due to the structure of Algorithm 5.3.1, if only the Euler characteristic is desired we could instead compute only the zero dimensional component of the \( c_{SM} \) class, \( (c_{SM}(X_{\Sigma}))_0 \). This procedure is described below in Algorithm 5.3.2. Algorithm 5.3.2 will offer reduced running time in comparison to Algorithm 5.3.1 when one only wishes to compute the Euler characteristic as no unnecessary computations will be performed.

**Algorithm 5.3.2.**

**Input:** A complete, simplicial toric variety \( X_{\Sigma} \) defined by a fan \( \Sigma \) with \( \Sigma(1) = \{\rho_1, \ldots, \rho_r\} \).

**Output:** The Euler characteristic \( \chi(X_{\Sigma}) \).

- Compute the rational Chow ring \( A^*(X_{\Sigma})_{\mathbb{Q}} \cong \mathbb{Q}[x_1, \ldots, x_r]/(I + J) \) using Proposition 5.2.2.
- orbits = all subsets of \( \Sigma(1) = \{\rho_1, \ldots, \rho_r\} \) containing \( \dim(X_{\Sigma}) \) elements.
- total = 0.

**For** \( \rho_j, \ldots, \rho_j \) **in** orbits:

- \( \sigma = \rho_{j_1} + \cdots + \rho_{j_s} \).

- Find \( w = \text{mult}(\sigma) \) using Lemma 5.3.2.

- \[ [V(\sigma)] = \text{mult}(\sigma)[V(\rho_{i_1})] \cdots [V(\rho_{i_t})] = w \cdot x_{i_1} \cdots x_{i_t}. \]

- total = total + \[ [V(\sigma)]. \]
- Set \( (c_{SM}(X_{\Sigma}))_0 = \text{total}. \)
• Set $\chi(X_\Sigma)$ equal to the sum of the coefficients of the monomials in $(c_{SM}(X_\Sigma))_0$.

• Return $\chi(X_\Sigma)$.

We note that both Algorithm 5.3.1 and Algorithm 5.3.2 are strictly combinatorial; hence the runtime of each algorithm depends only on the combinatorics of the fan $\Sigma$ defining the toric variety $X_\Sigma$.

We now give an example of the output and input to Algorithm 5.3.1 and Algorithm 5.3.2 for a singular complete simplicial toric variety.

**Example 5.3.4.** Assume that our implementation of Algorithm 5.3.1 is called “CSM-Toric”, that our implementation of Algorithm 5.3.2 is called “EulerToric” and that each algorithm takes as input a normal toric variety in Macaulay2 [4]. In this example we show the results of using Algorithm 5.3.1 to compute the $c_{SM}$ class and Euler characteristic and of using Algorithm 5.3.2 to compute only the Euler characteristic of the singular complete simplicial toric variety $X = X_\Sigma$ defined below.

Rho = {{-1,-1,1},{3,-1,1},{0,0,1},{1,0,1},{0,1,1},{-1,3,1},{0,0,-1}};
Sigma = {{0,1,3},{0,1,6},{0,2,3},{0,2,5},{0,5,6},{1,3,4},
        {1,4,5},{1,5,6},{2,3,4},{2,4,5}};
X = normalToricVariety(Rho,Sigma);

isSmooth X = false
isComplete X = true
isSimplicial X = true

The list Rho represents the list of generating rays $\Sigma(1)$, the fan $\Sigma$ is defined by taking each list in Sigma to represent the collection of rays indexed by the list, i.e. the list $\{0,1,3\}$ defines the cone $\sigma = \rho_0 + \rho_1 + \rho_3$ where $\rho_0 = \langle(-1,-1,1)\rangle$, $\rho_1 = \langle(3,-1,1)\rangle$, and $\rho_3 = \langle(1,0,1)\rangle$. The fan $\Sigma$ is then defined by the 10 cones listed in Sigma and all of their faces. Now we may compute the $c_{SM}$ class
CSMToric X;

giving

\[ c_{SM}(X_\Sigma) = 6x_4 + 88x_1x_5 - 52x_3^2 + 2x_2^3 + \frac{47}{3}x_2x_5 - 8x_1 + \frac{1}{3}x_2^2 + 16x_5 + 2x_2 + 1 \]

in the rational Chow ring of \( X_\Sigma \) which is given by

\[ A'(X_\Sigma)_\mathbb{Q} = \mathbb{Q}[x_0, x_1, x_2, x_3, x_4, x_5, x_6] / \langle x_1x_2, x_0x_4, x_0x_1x_5, x_3x_5, x_2x_6, x_3x_6, x_4x_6, -x_0 + x_3 + 3x_1 - x_5, -x_0 + x_4 - x_1 + 3x_5, -x_6 + x_3 + x_4 + x_1 + x_5 + x_2 \rangle \]

where \( x_0 \) corresponds to the ray \( p_0 = (-1, -1, 1) \) and so on under the isomorphism given in Proposition 5.2.2. If we wanted only the find the Euler characteristic we could instead run

EulerToric X;

=2

to find that \( \chi(X_\Sigma) = 2 \).

Having already found the \( c_{SM} \) class we could have, alternatively, used Proposition 5.2.1 and the presentation of the Chow ring given above to find that \( \{ x_3^2 \} \) forms a basis for the dimension zero Chow group \( A_0(X_\Sigma) \). Thus from our expression for \( c_{SM}(X_\Sigma) \) we obtain that the Euler characteristic of \( X_\Sigma \) is

\[ \chi(X_\Sigma) = \int c_{SM}(X_\Sigma) = 2, \]

that is the coefficient of \( x_3^2 \) in \( c_{SM}(X_\Sigma) \).
5.4 Performance

In this section we give the run times for Algorithm 5.3.1 and Algorithm 5.3.2 applied to a variety of examples. Since the goal here is to see how long the algorithm takes on larger examples we will primarily use complete simplicial toric varieties $X_\Sigma$ which can be constructed using built-in methods in the “NormalToricVarieties” Macaulay2 [4] package. Because of this nearly all the examples considered will be smooth (meaning we are just computing Chern classes).

Consider a complete simplicial toric variety $X_\Sigma$. We give two alternate implementations of Algorithms 5.3.1 and 5.3.2 to reflect what we can expect the timings to be in both the smooth cases and singular cases. Specifically the running times in Table 5.1 for Algorithm 5.3.1 and Algorithm 5.3.2 which are marked with a † check the input to see if the given fan $\Sigma$ defines a smooth toric variety, if it does these implementations use the fact that $\text{mult}(\sigma) = 1$ for all $\sigma \in \Sigma$ and hence do not compute the Hermite normal forms and their determinates in Lemma 5.3.2. However to show how the algorithms would perform on a singular input of a similar size and complexity we also give running times for an implementation which always computes the Hermite forms and their determinates in Lemma 5.3.2. Hence the running time for a given example would be very similar to that of a singular toric variety with a similar number and dimension of cones to those considered in the examples in Table 5.1. In this way we see in a precise manner what the extra cost associated to computing the $c_{S,M}$ class and Euler characteristic of a singular toric variety would be in comparison to the cost of computing a smooth toric variety defined by a fan having similar combinatorial structure.

By default the implementation of Algorithms 5.3.1 and 5.3.2 in our “CharToric” package checks if the input defines a smooth toric variety, i.e. performs the procedure of the implementations marked with †. As such the performance of the package methods on smooth cases can be expected to be that of Algorithm 5.3.1 † and Algorithm 5.3.2 † in Table 5.1 below.

We also remark that the extra cost in the singular case (or in the case where we don’t
check to see if the input is singular) comes entirely from performing linear algebra with integer matrices. As such the running times in these cases could perhaps be somewhat reduced by using a specialized integer linear algebra package to compute the Hermite normal forms and determinates appearing in Lemma 5.3.2. To give a rough quantification of what performance improvement one might expect from using such a linear algebra package we performed some testing using LinBox [5] and PARI [7] via Sage [6] on linear systems of similar size and structure to those arising in the examples in Table 5.1. In this testing we found that the specialized algorithms available through Sage [6] seemed to be around two to three times faster than the linear algebra methods used by our implementation in the “CharToric” package, however this testing is by no means conclusive. In any case it seems reasonable to conclude that some performance increase could be expected, for singular examples, if one used a specialized, fast integer linear algebra package to compute the Hermite forms and determinates arising in Algorithms 5.3.1 and 5.3.2.

### Table 5.1: Running times for Algorithm 5.3.1 ($c_{SM}$ and Euler) and Algorithm 5.3.2 (only Euler), our algorithms to compute the $c_{SM}$ class and Euler characteristic of a complete simplicial toric variety. The † denotes that for these versions of the algorithms the input is checked for smoothness. Meaning that, in the † versions, if the input is found to be smooth we know $\text{mult}(\sigma) = 1$ for all cones $\sigma \in \Sigma$ and hence we do not compute Hermite normal forms and determinates, see Lemma 5.3.2. In this table we present the time to compute the Chow ring separately from the time required for the other computations, as such the total run time for each algorithm will be the time listed in its column plus the time to compute the Chow ring if the Chow ring is not already known. Computations were performed using Macaulay2 [4] on a computer with a 2.9GHz Intel Core i7-3520M CPU and 8 GB of RAM.

<table>
<thead>
<tr>
<th>Input</th>
<th>Alg. 5.3.1 †</th>
<th>Alg. 5.3.2 †</th>
<th>Alg. 5.3.1</th>
<th>Alg. 5.3.2</th>
<th>Chow Ring (Prop. 5.2.2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 5.3.4</td>
<td>0.0s</td>
<td>0.0s</td>
<td>0.0s</td>
<td>0.0s</td>
<td>0.1 s</td>
</tr>
<tr>
<td>$\mathbb{P}^6$</td>
<td>0.0s</td>
<td>0.0s</td>
<td>0.0s</td>
<td>0.0s</td>
<td>0.1 s</td>
</tr>
<tr>
<td>$\mathbb{P}^{12}$</td>
<td>0.2s</td>
<td>0.0s</td>
<td>3.8s</td>
<td>0.0s</td>
<td>0.3 s</td>
</tr>
<tr>
<td>$\mathbb{P}^{16}$</td>
<td>5.3s</td>
<td>0.0s</td>
<td>85.4s</td>
<td>0.0s</td>
<td>0.7 s</td>
</tr>
<tr>
<td>$\mathbb{P}^5 \times \mathbb{P}^6$</td>
<td>0.3s</td>
<td>0.0s</td>
<td>3.7s</td>
<td>0.0s</td>
<td>1.2 s</td>
</tr>
<tr>
<td>$\mathbb{P}^5 \times \mathbb{P}^8$</td>
<td>1.1s</td>
<td>0.0s</td>
<td>16.8s</td>
<td>0.1s</td>
<td>2.1 s</td>
</tr>
<tr>
<td>$\mathbb{P}^8 \times \mathbb{P}^8$</td>
<td>12.0s</td>
<td>0.1s</td>
<td>168.5s</td>
<td>0.1s</td>
<td>4.5 s</td>
</tr>
<tr>
<td>$\mathbb{P}^5 \times \mathbb{P}^5 \times \mathbb{P}^5$</td>
<td>12.8s</td>
<td>0.2s</td>
<td>156.7s</td>
<td>0.6s</td>
<td>11.8 s</td>
</tr>
<tr>
<td>$\mathbb{P}^5 \times \mathbb{P}^5 \times \mathbb{P}^6$</td>
<td>28.4s</td>
<td>0.3s</td>
<td>387.1s</td>
<td>0.8s</td>
<td>17.0 s</td>
</tr>
<tr>
<td>Fano sixfold 123</td>
<td>0.3s</td>
<td>0.0s</td>
<td>1.0s</td>
<td>0.4s</td>
<td>1.1 s</td>
</tr>
<tr>
<td>Fano sixfold 423</td>
<td>0.3s</td>
<td>0.0s</td>
<td>0.7s</td>
<td>0.2s</td>
<td>0.8 s</td>
</tr>
<tr>
<td>Fano sixfold 1007</td>
<td>0.4s</td>
<td>0.1s</td>
<td>1.0s</td>
<td>0.1s</td>
<td>1.8 s</td>
</tr>
</tbody>
</table>
Bibliography


Chapter 6

Bézout Type results in Multi-Projective Space For Application to Automated Polynomial System Solving

In this chapter we prove several Bézout-like results in multi-projective space and investigate their application to giving degree bounds on systems of polynomial equations with a natural multi-projective structure. These degree bounds are of interest for finding refined running time bounds on algorithms working with systems of polynomial equations with a natural multi-projective structure.

The first such result for bi-projective space was proved in the 1920’s by van der Waerden [15], however this result considered only the case of a hypersurface transversally intersecting a variety. In 1985 another result of this type, in the multi-projective setting, was shown by Masser and Wustholz in [11], again this result only considered the intersection of a variety with a hypersurface, and had certain assumptions on the degree of the polynomial defining the hypersurface.

Following this, Philippon [14, Proposition 3.3] and Brownawell [2] were able to prove another Bézout-like bound for multi-projective space which, in geometric terms, considers a scheme $Y$ intersecting $V(f_1, \ldots, f_k)$ where $f_1, \ldots, f_k$ are multi-
homogeneous polynomials. The results of Philippon and Brownawell do not use Chow rings and intersection products to describe their bounds, and instead phrase the results in terms of degrees of ideals. In [12] Nakamaye shows an additional Bézout type result in multi-projective space phrased in the language of schemes. Nakamaye also translates the multi-projective Bézout type bound of Philippon to the language of schemes and shows that the Bézout type result given in [12] is, in fact, equivalent to the earlier result of Philippon [14, Proposition 3.3] which was also proved by Brownawell [2]. We state the result of Nakamaye [12] in Theorem 6.1.4 below.

We note that each of these results is structured in such a way as to be useful for the specific applications considered by the authors. In this chapter we present several results in a similar vein, but with a different application in mind. Namely the goal of this chapter is to give several Bézout type bounds in multi-projective space that can be easily used to bound the degree of a polynomial system which has a natural multi-projective structure such as may arise, for example, when trying to find the critical locus of an algebraic variety using the method of Lagrange multipliers. Additionally we structure the result in such a way so as to make it easy to apply in combination with known running time bounds for polynomial system solving. In particular these results could allow one to obtain a refined degree bound for polynomial system with a natural multi-projective structure with the refined bound being sharper than a bound obtained by ignoring the multi-projective structure and using the regular Bézout bound in projective space.

Recall that the motivating example for the work in this chapter was illustrated in §1.3.4 where we considered the system of equations investigated by Safey El Din and Trébuchet [5] as part of their construction of an algorithm to compute at least one point in each connected component of a smooth real algebraic set. This system is given in (1.18) and consists of equations having degree at most $D$ and defining an affine variety $V \subset \mathbb{A}^{n+m}$. Suppose $W_1, \ldots, W_t$ are the irreducible components of $V$. In this example the bi-projective Bézout-like bound of Theorem 6.3.7 gives us the
degree bound
\[ \sum \ell(O_{W_i,V}) \deg(W_i) \leq D^n \binom{n+m-1}{n-1}, \]
which is sharper (at least for large \( D \)) than the usual Bézout bound in \( \mathbb{P}^{n+m} \) which would give \( \sum \ell(O_{W_i,V}) \deg(W_i) \leq D^{n+m} \).

### 6.1 Review

In this section we review some notation and results that will be used to prove the Bézout-like bounds in §6.2. The primary references for this section are Fulton [7, 6], Eisenbud and Harris [4] and Gathmann [8].

In this chapter a variety will be assumed to be a reduced and irreducible scheme. A subvariety of a scheme will be taken to be a reduced and irreducible subscheme.

Any subvariety \( V \) of a scheme \( Y \) corresponds to a prime ideal in the coordinate ring of any affine open set meeting \( V \). We will let \( O_{V,Y} \) denote the local ring of \( Y \) along \( V \), that is \( O_{V,Y} \) is the localization of the coordinate ring of any affine open set meeting \( V \) at the corresponding prime ideal. In the notation of Grothendieck \( O_{V,Y} \) would be considered the stalk of the structure sheaf \( O_Y \) of \( Y \) at the generic point of \( V \).

Alternatively \( O_{V,Y} \) can be defined as the set of equivalence classes \( \langle U, f \rangle \) where \( U \subseteq Y \) is open, \( U \cap V \neq \emptyset \), and \( f \) is regular on \( U \). In this construction we consider \( \langle U_1, f_1 \rangle \) equivalent to \( \langle U_2, f_2 \rangle \) if \( f_1 = f_2 \) on \( U_1 \cap U_2 \). One may show that \( O_{V,Y} \) is a local ring of dimension \( \dim Y - \dim V \) with maximal ideal
\[ \mathcal{M}_{V,Y} = \{ f \in O_{V,Y} \mid f(x) = 0 \ \forall x \in V \}. \]

If \( X \) is a subvariety of \( k^n = \text{Spec}(k[x_1, \ldots, x_n]) \) for \( k \) an algebraically closed field and if \( V \) is a subscheme of \( X \) with irreducible component \( W \) then
\[ O_{W,V} = O_{W,X}/I(V)O_{W,X}, \] (6.1)
where $I(V)$ is the ideal of $V$ and $O_{W,X} = (k[x_1, \ldots, x_n]/I(X))_{I(W)}$ is the localization of the coordinate ring of $X$ at the prime ideal $I(W)$. In particular if $X = k^n$ then

$$O_{W,V} = (k[x_1, \ldots, x_n]/I(V))_{I(W)}.$$ 

We will let $\ell_A(M)$ denote the length of the $A$-module $M$; for a local ring $O$ we will write $\ell(O) = \ell_O(O)$ for the length of $O$ as an $O$-module. Recall from §1.3.4 that for $W$ an irreducible component of a scheme $V$ we refer to $\ell(O_{W,V})$ as the geometric multiplicity of $W$ in $V$. See Example 1.3.5 for an example of the computation of the geometric multiplicity.

For $\mathbb{P}^n = \text{Proj}(k[x_0, \ldots, x_n])$ an $n$ dimensional projective space over an algebraically closed field $k$ denote by $O_{\mathbb{P}^n}(1)$ the Serre twisting sheaf on $\mathbb{P}^n$ (referred to as the canonical line bundle in Fulton [7] and as the tautological line bundle in Eisenbud and Harris [4]). Also let $O_{\mathbb{P}^n}(d) = \bigotimes_{i=1}^d O_{\mathbb{P}^n}(1)$ for $d \in \mathbb{Z}$. Recall that the global sections of $O_{\mathbb{P}^n}(d)$, $\Gamma(O_{\mathbb{P}^n}(d))$, are the polynomials in $k[x_0, \ldots, x_n]$ having degree $d$.

Consider the product of $m$ projective spaces $\mathbb{P} = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ and let $h_i = c_1(O_{\mathbb{P}^{n_i}}(1))$ be the class of a hyperplane on $\mathbb{P}^{n_i}$ for each $i = 1, \ldots, m$. Recall that

$$A^*(\mathbb{P}) \cong \mathbb{Z}[h_1, \ldots, h_m]/(h_1^{n_1+1}, \ldots, h_m^{n_m+1}).$$ 

(6.2)

Also recall that for a smooth variety $M$, elements of the dimension $j$ Chow group $A_j(M)$ will be referred to as dimension $j$-cycles and elements of the codimension $c$ Chow group $A^c(M)$ will be referred to as codimension $c$-cycles. See §2.1.1 for more details.

**Definition 6.1.1** (Fulton [7] Chapter 12, and Nakamaye [12]). For a dimension $j$-cycle $\alpha \in A^*(\mathbb{P})$ and a vector of integers $d = (d_1, \ldots, d_m)$ we let $\deg_{\alpha}(d)$ denote the degree of $\alpha$ taken with respect to the line bundle

$$O_{\mathbb{P}}(d) = O_{\mathbb{P}}(d_1, \ldots, d_m) = \pi_1^* O_{\mathbb{P}^{n_1}}(d_1) \otimes \cdots \otimes \pi_m^* O_{\mathbb{P}^{n_m}}(d_m),$$
where $\pi_1 : \mathbb{P} \to \mathbb{P}^{n_1}, \ldots, \pi_m : \mathbb{P} \to \mathbb{P}^{n_m}$ are projection maps. That is
\[
\deg_d(\alpha) = \int c_1(O_{\mathbb{P}}(d_1, \ldots, d_m))^j \cdot \alpha.
\] (6.3)

This definition can be made more explicit, specifically we have that
\[
\deg_d(\alpha) = \int c_1(O_{\mathbb{P}}(d))^j \cdot \alpha = \int (d_1h_1 + \cdots + d_mh_m)^j \cdot \alpha.
\] (6.4)

Recall from §2.1.1 that the notation $\int \alpha$ for $\alpha \in A^*(\mathbb{P})$ denotes the degree of the zero dimensional component of $\alpha$ in $A_0(\mathbb{P})$, i.e. $\int \alpha$ is the coefficient of $h_1^{n_1} \cdots h_m^{n_m}$ in the polynomial expression of $\alpha$.

We now give an example of the computation of $\deg_d$ for a hypersurface in $\mathbb{P}^2 \times \mathbb{P}^3$.

**Example 6.1.2.** Consider a hypersurface $V = V(5x_0^3x_2^2l_0l_3 - l_2^2x_1^5)$ in $\mathbb{P}^2 \times \mathbb{P}^3 = \text{Proj}(k[x_0, x_1, x_2]) \times \text{Proj}(k[l_0, l_1, l_2, l_3])$. The Chow ring is given by

\[
A^*(\mathbb{P}^2 \times \mathbb{P}^3) \cong \mathbb{Z}[h_1, h_2]/(h_1^{2+1}, h_2^{2+1}) = \mathbb{Z}[h_1, h_2]/(h_1^2, h_2^2),
\]

and $[V] = 5h_1 + 2h_2$. In this case
\[
\deg_{(1,1)}(V) = \int (h_1 + h_2)^4(5h_1 + 2h_2)
= \int 32s^2t^3.
= 32.
\]
More generally a hypersurface $W$ with $[W] = ah_1 + bh_2$ would have

$$\deg_{(d_1, d_2)}(W) = \int (d_1h_1 + d_2h_2)^4(ah_1 + bh_2)$$

$$= \int \left( \left( \frac{4}{2} \right) d_1^2d_2^2b + \left( \frac{4}{3} \right) d_1d_2^3a \right) h_1^2h_2^3$$

$$= \left( \frac{4}{2} \right) d_1^2d_2^2b + \left( \frac{4}{3} \right) d_1d_2^3a.$$

For a non-pure dimensional scheme

$$X = \bigcup_{i=0}^{\dim X} X_i,$$

where the $X_i$ are the union of the irreducible components of $X$ with dimension $i$, set

$$\deg_d(X) = \sum_{i=1}^{\dim X} \deg_d(X_i).$$

We now state some previous results of Patil and Vogel [13] and of Nakamaye [12] which are used in Section 6.2 below.

**Theorem 6.1.3** (Theorem 2.1 [13]). Let $V = V(F_1, \ldots, F_r)$ be any subscheme of $\mathbb{P}^n = \text{Proj}(k[x_0, \ldots, x_n])$ given (scheme-theoretically) by the intersection of $r \geq 1$ hypersurfaces defined by homogenous polynomials $F_1, \ldots, F_r$ of degrees $d_1, \ldots, d_r$ respectively. Then we have

$$\sum_{C \text{ irr. comp. of } V} \ell \left( \frac{k[x_0, \ldots, x_n]}{(F_1, \ldots, F_r)}_{(I(C))} \right) \cdot \deg(C) \leq \prod_{i=1}^{r} d_i$$

where $C$ runs through all the irreducible components of $V$ and $I(C)$ is the prime ideal defining $C$.

The results in Section 6.2 below are obtained primarily by modifying the proof of the following theorem of Nakamaye [12].

**Theorem 6.1.4** (Theorem 1.1 [12]). Let $S \subset \Gamma(\mathcal{O}_\mathbb{P}(d_1, \ldots, d_m))$ be a collection of
homogeneous forms of multihomogeneous forms of multi-degree \( d = (d_1, \ldots, d_m) \) and let \( J = (S) \) be the multihomogeneous ideal generated by \( S \). Let \( X \) be a pure dimensional subscheme of \( \mathbb{P} \) and let \( Y_j \) be the irreducible components of \( X \cap V(J) \). Then

\[
\sum_{\ell(O_{Y_j, X}) \text{ is CM}} \ell(O_{Y_j, X}) \cdot \deg_d(Y_j) \leq \deg_d(X),
\]

where the sum is over those \( Y_j \) such that \( O_{Y_j, X} \) is Cohen-Macaulay.

### 6.2 Bézout-like Results

In this section we give the main results of this chapter, namely Theorem 6.2.1 and Theorem 6.2.2. These results give Bézout-like bounds on the degree of a subscheme of multi-projective space with respect to a certain line bundle.

Let \( \mathbb{P} = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \) denote multi-projective space and let \( \deg_d \) be as defined in Definition 6.1.1, recall this degree may be computed using (6.4).

Using Theorem 2.1 of Patil and Vogel [13] (Theorem 6.1.3 above) and the proof techniques used by Nakamaye [12] to prove Theorem 6.1.4 above we establish the following:

**Theorem 6.2.1.** Let \( V_1, \ldots, V_r \) be hypersurfaces in \( \mathbb{P} = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \) generated by the multi-homogeneous polynomials \( F_1, \ldots, F_r \) respectively, with the \( F_1, \ldots, F_{r-1} \) of multi-degree less than \( d = (d_1, \ldots, d_m) \) (that is if the multi-degree of \( F_i \) is \( (j_1, \ldots, j_m) \) then \( j_l \leq d_l \) for all \( l \) and \( i = 1, \ldots, r - 1 \)). Also let \( J = (F_1, \ldots, F_{r-1}) \) and let \( W_i \) be the irreducible components of \( V = V(F_1) \cap \cdots \cap V(F_r) \), then we have

\[
\sum \ell(O_{W_i, V}) \deg_d(W_i) \leq \deg_d(V_r).
\]

Here \( \ell(O_{W_i, V}) \) denotes the geometric multiplicity, see §6.1 or Example 1.3.5 for an example of its computation.

**Proof.** First we reduce to the case where all generators of \( J \) have multi-degree \( d \). If
we multiply each $F_i$ generating $J = (F_1, \ldots, F_{r-1})$ by a set of forms of degree $d - \deg(F_i)$ which generate a projectively irrelevant ideal we get a new set of generators $P_1, \ldots, P_{r-1}$ for $J$ which have multi-degree $d$. Now let $X = V(F_r)$. Following the technique of Nakamaye [12] we take

$$t = \max_j (\text{codim}(W_j, X))$$

and construct a set of polynomials $Q_1, \ldots, Q_t$ in $J$, by taking generic linear combinations of the generators of $J$, with the property that each $W_i$ is an irreducible component of $V(J) \cap X$. Let $I = (Q_1, \ldots, Q_t)$, by construction we have $I \subset J$ so

$$\ell(O_{W_i, V(J) \cap X}) \leq \ell(O_{W_i, V(I) \cap X}).$$

We wish to apply Theorem 2.1 of Patil and Vogel [13] (Theorem 6.1.3 above), however this result is for hypersurfaces in $\mathbb{P}^N$, hence to apply the result we will first consider a Segre embedding $i : \mathbb{P} \to \mathbb{P}^N$ with basis determined by the global sections of $O_{\mathbb{P}}(d_1, \ldots, d_m)$, so that $\deg_d(Y) = \deg(i(Y))$ for any scheme $Y$ in $\mathbb{P}$. Here $\deg$ denotes the degree in $\mathbb{P}^N$ computed with respect to $O_{\mathbb{P}^N}(1)$.

Let $X' = i(X)$ and $W_i' = i(W_i)$. Choose linear forms $L_i \in \Gamma(O_{\mathbb{P}^N}(1))$ such that $i^*(L_i) = Q_i$ for each $i = 1, \ldots, t$. Applying Theorem 2.1 of [13] to the hypersurfaces $V(L_1), \ldots, V(L_t), X'$ in $\mathbb{P}^N$ we have:

$$\sum \ell(O_{W_i', X' \cap V(L_1) \cap \cdots \cap V(L_t)}) \deg(W_i') \leq \deg(X') \cdot \prod_{i=1}^t \deg(V(L_i)) = \deg(X').$$

(6.5)

Note that, since $i^*L_i = Q_i$, we have $\ell(O_{W_i', X' \cap V(L_1) \cap \cdots \cap V(L_t)}) = \ell(O_{W_i, V_i \cap \cdots \cap V_i})$ and also note that $\deg(i(Y)) = \deg_d(Y)$ for any $Y \in \mathbb{P}$. Hence we obtain

$$\sum \ell(O_{W_i, V}) \deg_d(W_i) \leq \deg_d(V_r).$$

□

If we know a priori that the scheme $V(F_1, \ldots, F_r)$ is pure dimensional and Cohen-
Macaulay (for example if it is a complete intersection or contains only regular points) then we may state a modified version of Theorem 6.2.1 which may potentially be stronger in certain cases. The proof of the theorem below is identical to that of Theorem 6.2.1 except instead of using the result of Theorem 2.1 of Patil and Vogel [13] (Theorem 6.1.3 above) we would instead use the group of results given in Theorem 1.2, Theorem 2.3 and Corollary 3.4 of Patil and Vogel [13].

**Theorem 6.2.2.** Let \( V_1, \ldots, V_r \) be hypersurfaces in \( \mathbb{P} = \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \) generated by the multi-homogeneous polynomials \( F_1, \ldots, F_r \) respectively, with the \( F_{\rho+1}, \ldots, F_r \) of multi-degree less than \( d = (d_1, \ldots, d_m) \) (that is if the multi-degree of \( F_i \) is \( (j_1, \ldots, j_m) \) then \( j_l \leq d_l \) for all \( l \) and all \( i = \rho + 1, \ldots, r \). Additionally assume that \( V(F_1, \ldots, F_\rho) \) is pure dimensional and Cohen-Macaulay. Also let \( W_i \) be the irreducible components of \( V = V(F_1) \cap \cdots \cap V(F_r) \), then we have

\[
\sum \ell(\mathcal{O}_{W_i,V}) \deg_d(W_i) \leq \deg_d(V(F_1, \ldots, F_\rho)).
\]

### 6.3 Applications

In this section we explore some applications of the results of §6.2 above to problems involving run time bounds of algorithms for solving polynomial systems.

In §6.3.1 we establish that the notion of geometric multiplicity from §1.3.4 and §6.1 is equivalent to a notion of multiplicity used by several others in the context of algorithms in polynomial systems. In §6.3.2 we show how the Bézout-like results of §6.2 can be applied to a subscheme of affine space with some natural bi-projective structure. This yields Theorem 6.3.7 which is then applied to give the corollaries used when considering the system discussed in §1.3.4 which motivated the results of this chapter.
6.3.1 Relations Between the Geometric Multiplicity and Other Multiplicity Functions

This section shows that the geometric multiplicity, that is the multiplicity in terms of lengths of local rings used in §6.2 and discussed in §6.1 and §1.3.4, is the same as the multiplicity used in several complexity results, in particular those of Lecerf [10].

Also note that the definition of multiplicity below used by Giusti, Lecerf and Salvy [9], Lecerf [10], and others, is equivalent to the notion of multiplicity often used in numerical algebraic geometry; see for example the definition of multiplicity used by Bates and Sommese in [1]. This equivalence is established in the proof of Lemma 6.3.3 below.

Definition 6.3.1. [9, 10]. Let \((f_1, \ldots, f_m) \in k[x_1, \ldots, x_N]\) and let \(W\) be an \(r\)-dimensional irreducible component of the \(k\)-dimensional subscheme \(V = V(f_1, \ldots, f_m)\) of the affine space \(\mathbb{A}^N\). A geometric resolution of \(W\) is a data structure used to store and manipulate \(W\) for computational purposes. Let \(K = k(y_1, \ldots, y_r)\). A geometric resolution of \(W\) consists of the following:

- **\(M\)**, an invertible \(N \times N\) matrix over \(k\) constructed so that the coordinates \(y = M^{-1}x\) are such that \(y_1, \ldots, y_r\) are free with respect to \(W\), i.e. \(I(W) \cap k[y_1, \ldots, y_r] = (0)\).

- A field extension \(L = \mathcal{K}(u)\) obtained by adjoining \(u = \lambda_{r+1}y_{r+1} + \cdots + \lambda_Ny_N\) to \(\mathcal{K}\), with the field extension \(L\) having minimal polynomial \(q(T)\) so that

\[
L = \mathcal{K}(u) \cong \mathcal{K}[T]/q(T).
\]

- The parametrization of \(W\) by the zeros of \(q\), specified by polynomials

\[
v_{r+1}(T), \ldots, v_N(T) \in \mathcal{K}[T]
\]

such that \(y_j = v_j(u)\) in \(\mathcal{K}\) for \(r + 1 \leq j \leq N\) and \(\deg(v_j(T)) < \deg(q(T))\).
Using the structures of the above definition, Lecerf defines the multiplicity of \( W \) in \((f_1, \ldots, f_m)\) as

\[
\text{mul}(W, (f_1, \ldots, f_m)) = \dim_L L[[y_{r+1} - v_{r+1}, \ldots, y_N - v_N]]/((f_1, \ldots, f_m) \circ M),
\]

where \( L = k(y_1, \ldots, y_r)[T]/q(y_1, \ldots, y_r, T) \). Given \( f_1, \ldots, f_s \in k[x_1, \ldots, x_N] \) let

\[
\mathcal{V}_i := \{ z \in \overline{k}^n \mid f_1(z) = \cdots = f_i(z) = 0 \}, \quad i = 0, \ldots, s
\]

and let \( \mathcal{V}_i = \bigcup W_j^{(i)} \) be the irreducible decomposition of \( \mathcal{V}_i \). Define

\[
\delta^a := \max_{i=0, \ldots, m} \delta^a_i
\]

where

\[
\delta^a_i := \sum_j \text{mul}(W_j^{(i)}, (f_1, \ldots, f_i)) \deg(W_j^{(i)}).
\]

Lecerf \cite{10} gives the following result.

**Theorem 6.3.2.** \cite[Theorem 1]{10}. Let \( k \) be a field of characteristic zero. There exists a probabilistic algorithm taking as input a sequence \( f_1, \ldots, f_s, g \) of polynomials in \( k[x_1, \ldots, x_N] \) of degree at most \( d \) and let \( S \) be an upper bound on the complexity of evaluating the system \( f_1, \ldots, f_s \). The output is the equidimensional decomposition of the Zariski closure of the system

\[
f_1 = \cdots = f_s = 0, \quad g \neq 0.
\]

In case of success, the procedure requires

\[
O(s \log(d)N^4(NS + N^4)U(d\delta^a)^3)
\]

arithmetic operations in \( k \). The probability of success of the algorithm depends on the choice of a point in \( k^n \): there exists a Zariski open set of points that yield a correct answer. Here \( U(z) \) denotes a function which dominates the complexity of basic arithmetic operations (multiplication, division, gcd) for univariate polynomi-
als of degree less than or equal to z.

We note that if the system being considered has some natural multi-projective structure then, after translating the multiplicity of Lecerf [10] to something in terms of the geometric multiplicities used in §6.2, we will be able to use the Bézoiut-like bounds of §6.2 to give a bound for the $\delta_a$ appearing in Theorem 6.3.2 above.

Since the multiplicity of Lecerf is given in terms of the geometric resolution we need to convert this to a notion of multiplicity which fits into the framework of Section 6.2. In the process of proving this lemma we will also show that the geometric multiplicity we use in this chapter is equivalent to a notion of multiplicity often used in numerical algebraic geometry, see for example Bates and Sommese [1].

**Lemma 6.3.3.** Let $R = k[x_1, \ldots, x_N]$ and let $V = \text{Spec}(R/(f_1, \ldots, f_m))$ be the subscheme of $\text{Spec}(R)$ corresponding to the ideal $(f_1, \ldots, f_m)$. Take $W$ to be an irreducible component of $V$, we have

$$\text{mul}(W, (f_1, \ldots, f_m)) = \ell(O_{W,V}).$$

Here $\text{mul}(W, (f_1, \ldots, f_m))$ is the multiplicity of Lecerf defined above (see Equation (6.7)).

**Proof.** We will prove this result in three steps. First we show that $\text{mul}(W, (f_1, \ldots, f_m))$ is equal to the multiplicity of $p$-primary component of an ideal $I$ at $p$, denoted $\text{mul}(p, I)$ defined by Bates and Sommese [1], where $I = (f_1, \ldots, f_m)$ and $p = I(W)$.

Second we show that $\text{mul}(p, I)$ of [1] is in fact given by the length of a certain primary component of an ideal $I$ in the sense of Patil and Vogel [16]. Last we show that $\ell(O_{W,V})$ is equal to the definition of the length given by Patil and Vogel [16].

Work in the generic coordinates $y_1, \ldots, y_N$ so $\tilde{f}_i = f_i(y_1, \ldots, y_N) = f_i \circ M$. Let $(w_1, \ldots, w_N) \in W$ be a generic point in $W$ and let $L$ be the field extension (6.6) with minimal polynomial $q(T)$. First show that $\dim_L \mathcal{L}[[y_{r+1}-v_{r+1}, \ldots, y_N-v_N]]/(\tilde{f}_1, \ldots, \tilde{f}_m)$ is equal to $\dim_k k[[y_{r+1}-w_{r+1}, \ldots, y_N-w_N]]/(\tilde{f}_1, \ldots, \tilde{f}_m)$. Take $O_P := k(y_1, \ldots, y_r)_{(w_1, \ldots, w_r)}$ to be the localization of $k(y_1, \ldots, y_r)$ at the point $P = (w_1, \ldots, w_r)$. Let $u = \sum_{j=1} \lambda_j w_j$, then since $(w_1, \ldots, w_N)$ is generic we have a field extension $O_P(u) =$
$O_P[T]/q(T)$ with minimal polynomial $q(T)$ (the same minimal polynomial as $\mathcal{L}$). Define a ring morphism

$$\xi: O_P[T]/q(T) \to \bar{k}$$

$$\sum_i g_i T^i \mapsto \sum_i g_i(\bar{w}_1, \ldots, \bar{w}_r) u^i,$$

where $\bar{k}$ is the algebraic closure of $k$. Since $q(\bar{w}_1, \ldots, \bar{w}_r, u) = 0$ this map is well defined and extends to a map $\xi: (O_P[T]/q(T))[y_{r+1}, \ldots, y_N - v_N] \to \bar{k}[y_{r+1} - w_{r+1}, \ldots, y_N - w_N]$. Take $F_j = f_j(y_1, \ldots, y_r, y_{r+1} - v_{r+1}, \ldots, y_N - v_N)$ and $f^*_j = f_j(y_1, \ldots, y_r, y_{r+1} - w_{r+1}, \ldots, y_N - w_N)$, then $\xi(F_j) = f^*_j$.

We may apply a standard basis algorithm to compute

$$\Delta_{\mathcal{L}} := \dim \mathcal{L}[[y_{r+1}, \ldots, y_N]]/(F_1, \ldots, F_m).$$

Let $\Gamma_{\mathcal{L}}$ denote the coefficients that appear in the steps of the standard basis algorithm to compute the dimension above. Because $(\bar{w}_1, \ldots, \bar{w}_N) \in \mathcal{W}$ is generic we have, first, that the coefficients $\Gamma_{\mathcal{L}}$ are well defined, i.e. are in $(O_P[T]/q(T))$ and second that none of the $\Gamma_{\mathcal{L}}$ vanish at $P$. As the coefficients $\Gamma_{\mathcal{L}}$ do not vanish at $P$, the map $\xi$ applied to these coefficients will give the corresponding coefficients of a standard basis algorithm applied to compute $\Delta_k := \dim \bar{k}[[y_{r+1}, \ldots, y_N]]/(f^*_1, \ldots, f^*_m)$. Note that $\Delta_k$ is equal to $\dim \bar{k}[[y_{r+1} - w_{r+1}, \ldots, y_N - w_N]]/(\bar{f}_1, \ldots, \bar{f}_m)$ and $\Delta_{\mathcal{L}}$ is equal to $\dim \mathcal{L}[[y_{r+1} - v_{r+1}, \ldots, y_N - v_N]]/(\bar{f}_1, \ldots, \bar{f}_m)$. Thus $\dim \mathcal{L}[[y_{r+1} - v_{r+1}, \ldots, y_N - v_N]]/(\bar{f}_1, \ldots, \bar{f}_m)$ is equal to $\dim \bar{k}[[y_{r+1} - w_{r+1}, \ldots, y_N - w_N]]/(\bar{f}_1, \ldots, \bar{f}_m)$.

In [1] Bates and Sommese define the multiplicity of a prime ideal $p$ at an ideal $I$ to be $\text{mul}(p, I) = \text{deg}(a)/\text{deg}(p)$ where $a$ is the $p$-primary component of the minimal primary decomposition of $I$. For a generic $w = (w_1, \ldots, w_N) \in \mathcal{W}$ Theorem 17 of [1] gives us that the expression

$$\dim \bar{k}[[y_{r+1} - w_{r+1}, \ldots, y_N - w_N]]/(\bar{f}_1, \ldots, \bar{f}_m)$$

is the same as the multiplicity of the ideal prime ideal $I(W)$ at $I = (f_1, \ldots, f_m)$. 
Hence we have that the Lecerf multiplicity agrees with the multiplicity of Bates and Sommese [1].

Patil and Vogel [16, (1.1)] define the length of a $p$-primary ideal $q$, $\ell(q)$, as the length of Artinian local ring $(k[x_1, \ldots, x_n]/q)_p$, i.e. $\ell(q) = \ell((k[x_1, \ldots, x_n]/q)_p)$. Additionally Patil and Vogel [16, (1.38)] show that $\deg(p)\ell(q) = \deg(q)$, hence

$$\text{mul}(I(W), (f_1, \ldots, f_m)) = \deg(q)/\deg(p) = \ell(q).$$

Let $X = \text{Spec}(R)$. For the last step consider $O_{W,V}$; recall that $V$ is the scheme $\text{Spec}(R/(f_1, \ldots, f_m))$ and that $p = I(W)$ is the prime ideal defining the irreducible component $W$ of $V$. Also let $R_p$ denote the localization of $R$ at the prime ideal $p$. Then we have,

$$O_{W,V} = O_{W,X}/I(V)O_{W,X} \quad \text{(Fulton [6, §2.1])}$$

$$= R_p/I(V)R_p$$

$$= (R/I(V))_p \quad \text{(localization commutes with taking quotients).}$$

$$= (R/q)_p \quad \text{(Eisenbud [3, Theorem 3.10]).}$$

This concludes the proof. \qed

### 6.3.2 Affine Varieties with a Bi-projective Structure

To apply the Bézout-like bounds of Section 6.2 to compute $\delta_s$ (see (6.9)) in the complexity bound of Lecerf above (Theorem 6.3.2) when the polynomials $f_1, \ldots, f_s \in k[x_1, \ldots, x_N]$ possess a natural bi-projective structure, such as in our motivating example from §1.3.4, we first need to establish several additional results.

For the remainder of this section let $\mathbb{P} = \mathbb{P}^n \times \mathbb{P}^m$, with projection maps $\pi_1 : \mathbb{P} \to \mathbb{P}^n$,
\( \pi_2 : \mathbb{P} \to \mathbb{P}^m \), and let \( O_{\mathbb{P}}(1, 1) \) be the line bundle on \( \mathbb{P} \) given by
\[
O_{\mathbb{P}}(1, 1) = \pi_1^* O_{\mathbb{P}_n}(1) \otimes \pi_2^* O_{\mathbb{P}_m}(1).
\]

Write the Chow ring of \( \mathbb{P} \) as \( A^*(\mathbb{P}) = \mathbb{Z}[s, t]/(s^{n+1}, t^{m+1}) \). Suppose that the variables \( x_1, \ldots, x_N \) are split into two blocks so that \( x_1, \ldots, x_N = x_1, \ldots, x_n, l_1, \ldots, l_m \).

**Definition 6.3.4.** Define \( \varphi : k[x_1, \ldots, x_n, l_1, \ldots, l_m] \to k[x_0, \ldots, x_n, l_0, \ldots, l_m] \) to be the homogenization mapping
\[
\varphi : f \mapsto x_0^{\deg(f)} l_0^{\deg(f)} f \left( \frac{x_1}{x_0}, \ldots, \frac{x_n}{x_0}, \frac{l_1}{l_0}, \ldots, \frac{l_m}{l_0} \right).
\]  

Also in the following, for a purely \( r \)-dimensional subscheme \( \hat{V} \) of \( \mathbb{A}_n^m \) we will write \( \deg(\hat{V}) \) to be the number of points of intersection of \( \hat{V} \) with \( N - r \) generic hyperplanes. This is the same as the degree of the projective closure, \( V \subset \mathbb{P}^N \), of \( \hat{V} \), i.e. \( \deg(\hat{V}) = \deg(V) \).

**Lemma 6.3.5.** Let \( \hat{V} = V(I) \) be a purely \( r \)-dimensional subscheme of the affine space \( \mathbb{A}_n^m \) defined by the ideal \( I \). Also let \( V = V(\varphi(I)) \). This is a purely \( r \)-dimensional subscheme of \( \mathbb{P}^n \times \mathbb{P}^m \), and we have
\[
\deg(\hat{V}) \leq \deg_{(1, 1)}(V).
\]

Also note that \( \deg_{(1, 1)}(\alpha) \leq \deg_{(d_1, d_2)}(\alpha) \) for any \( d_1, d_2 \geq 1 \) and any cycle \( \alpha \in A^*(\mathbb{P}) \).

**Proof.** In \( A^*(\mathbb{P}^n \times \mathbb{P}^m) \cong \mathbb{Z}[s, t]/(s^{n+1}, t^{m+1}) \), we have
\[
[V] = \sum_{i+j=r} a_{i, j} s^{n-i} t^{m-j}.
\]

This gives,
\[
\deg_{(1, 1)}(V) = \int_X (s+t) (\sum_{i+j=r} a_{i, j} s^{n-i} t^{m-j}).
\]

Now let \( I^h \) be the homogenization of \( I \) and let \( V' = V(I^h) \in \mathbb{P}^{n+m+1} \). From Fulton [7],
§8.4.4] we have that geometrically $V'$ is given by

$$V' = \{(\lambda a_0 : \lambda a_1 : \cdots : \lambda a_n : \mu b_0 : \mu b_1 : \cdots : \mu b_m) \in \mathbb{P}^{n+m+1} | (a) \times (b) \in V, (\lambda : \mu) \in \mathbb{P}\}.$$ 

[7, §8.4.4] also gives that

$$\text{deg}(V') = \sum_{i+j=r} a_{i,j},$$

hence $\text{deg}(V') \leq \text{deg}_{(1,1)}(V)$ with equality when $r = 0$. Hence we have

$$\text{deg}(\hat{V}) = \text{deg}(V') \leq \text{deg}_{(1,1)}(V).$$

\[\Box\]

**Lemma 6.3.6.** Let $R = k[x_1, \ldots, x_N] = k[x_1, \ldots, x_n, l_1, \ldots, l_m]$. Also let $I \subset R$ be a non-zero ideal and take $\hat{V} = V(\varphi(I)) \subset \text{Proj}(k[x_0, \ldots, x_n]) \times \text{Proj}(k[l_0, \ldots, l_m])$ to be a subscheme of $\mathbb{P}^n \times \mathbb{P}^m$ with irreducible components $W_1, \ldots, W_\gamma$. Take $\hat{V} = \text{Spec}(R/I)$. Then $\hat{V}$ has irreducible components $\hat{W}_1, \ldots, \hat{W}_\gamma$ with $\hat{W}_j = W_j - V(x_0) - V(l_0)$ and

$$\ell(O_{\hat{W}_i, \hat{V}}) = \ell(O_{W_i, V}), \quad \text{for } i = 1, \ldots, \gamma.$$

**Proof.** Now take $W$ an irreducible component of $V$ and let $\hat{W} = W - V(x_0) - V(l_0)$. Say that $\ell(O_{W,V}) = \tau$, this means that there is a composition series for $O_{W,V}$ of length $\tau$

$$0 = M_0 \subsetneq M_1 \subsetneq \cdots \subsetneq M_\tau = O_{W,V}, \quad M_{i+1}/M_i \text{ a simple module.}$$

To complete the proof we evaluate $x_0 = l_0 = 1$ in each of the $M_i$ to obtain a composition series for $O_{\hat{W}, \hat{V}}$ of length $\tau$ as follows. $M_{i+1}/M_i = k[x_0, \ldots, x_n, l_0, \ldots, l_m]|_{\zeta}$ for any $\zeta \neq 0 \in M_{i+1}/M_i$ since $M_{i+1}/M_i$ is a simple $k[x_0, \ldots, x_n, l_0, \ldots, l_m]$-module. Evaluating $M_i, M_{i+1}$ at $x_0 = l_0 = 1$ will give two new $R$-modules $\hat{M}_i, \hat{M}_{i+1}$, respectively. Note that since the elements of the $k[x_0, \ldots, x_n, l_0, \ldots, l_m]$-modules $M_i, M_{i+1}$ are homogeneous, then when we evaluate each element at $x_0 = l_0 = 1$ we will get a new set of objects that have the structure of an $R$ module. Let $\zeta|_{x_0=l_0=1}$ denote $\zeta$ evaluated at $x_0 = l_0 = 1$. If $\hat{M}_i \neq \hat{M}_{i+1}$ then $\hat{M}_{i+1}/\hat{M}_i = R\zeta|_{x_0=l_0=1}$ for any $\zeta \in M_{i+1}/M_i$. 

since $\zeta$ generates $M_{i+1}/M_i$. Additionally since $\zeta \neq 0 \in M_{i+1}/M_i$ then $\zeta$ will be homogeneous in $x$ and $l$ so $\zeta|_{x_0=l_0}=1 \neq 0$. Hence we obtain a composition series for $O_{\hat{W}, \hat{V}}$ of length equal to $\tau$, so $\ell(O_{\hat{W}, \hat{V}}) = \tau$.

For the case of a subscheme of a certain affine space with some natural bi-projective structure we now summarize the results of the bi-projective Bézout-like bounds of Theorem 6.2.1 and Theorem 6.2.2 in Theorem 6.3.7 below.

Note that in light of Lemma 6.3.3 above the multiplicity used in the theorem below is the same as that used by Lecerf [10] and others when giving running time bounds for solving polynomial systems such as in Theorem 6.3.2 above, hence the degree bounds below can be easily applied in these cases. In particular when solving systems with a natural bi-projective structure using the algorithm of Lecerf [10] one may directly apply Theorem 6.3.7 below to obtain a refined bound for the $\delta_a$ (6.9) appearing in the run time bound for the algorithm of Lecerf [10] stated as Theorem 6.3.2 above.

**Theorem 6.3.7.** Let $V = V(f_1, \ldots, f_r)$ be a subscheme of

$$A^{n+m} = \text{Spec}(k[x_1, \ldots, x_n, l_1, \ldots, l_m])$$

and let $W_1, \ldots, W_t$ be the irreducible components of $V$ and let $\varphi$ be the homogenization mapping of Definition 6.3.4. Choose $d = (d_1, d_2)$ such that $\deg_x(\varphi(f_i)) \leq d_1$ and $\deg_l(\varphi(f_i)) \leq d_2$ for all $i$. We have that

$$\sum \ell(O_{W_i, V}) \deg(W_i) \leq \deg_d(V(\varphi(f_{\rho}))),$$

(6.11)

for any $\rho = 1, 2, \ldots, r$. Further if $V(f_1, \ldots, f_{\rho})$ is a complete intersection (for some $\rho$ and some arrangement of the equations defining $V$) then we also have the following, possibly sharper, bound

$$\sum \ell(O_{W_i, V}) \deg(W_i) \leq \deg_d(V(\varphi(f_1, \ldots, f_{\rho}))).$$

(6.12)

Here $\deg_d$ denotes the degree with respect to a certain line bundle, see (6.3) for a definition.
Proof. Starting with \( \sum \ell(\mathcal{O}_{W_i, V}) \deg(W_i) \) we apply Lemma 6.3.5 and Lemma 6.3.6 to convert to an expression in \( \mathbb{P}^n \times \mathbb{P}^m \) we may then apply either Theorem 6.2.1 or Theorem 6.2.2 to obtain the first and second bounds, respectively. \( \square \)

We may now state the corollaries applied in §1.3.4 which give degree bounds which may be used to give running time bounds on the algorithm of Safey El Din and Trebuchet in [5] to compute at least one point in each connected component of a smooth real algebraic set.

We recall the setting considered in §1.3.4. Let \( m < n \) and let \( f_1, \ldots, f_m \in k[x_0, \ldots, x_n] \) be homogeneous polynomials, with \( \deg(f_i) = d_i, d_i \leq D \) for all \( i \). Take

\[
F_j = \begin{cases} 
  f_j & \text{if } j \leq m \\
  l_1 \frac{\partial f_1}{\partial x_{j-m}} + \cdots + l_m \frac{\partial f_m}{\partial x_{j-m}} - l_0 & \text{if } j = m + 1 \\
  l_1 \frac{\partial f_1}{\partial x_{j-m}} + \cdots + l_m \frac{\partial f_m}{\partial x_{j-m}} & \text{if } m + 2 \leq j \leq m + n
\end{cases} 
\]  

(6.13)

We have the following corollaries to Theorem 6.3.7 which could, for example, be applied to bound the \( \delta_a \) in the complexity estimate on the algorithm of Lecerf [10] stated as Theorem 6.3.2 above (when applied to the system (6.13)).

**Corollary 6.3.8.** Let \( W_i \) be the irreducible components of \( V = V(F_1, \ldots, F_{n+m}) \) with the \( F_j \)'s as in (6.13). We have,

\[
\sum \ell(\mathcal{O}_{W, V}) \deg(W_i) \leq D^{n+m-1} \binom{n+m-1}{n-1}.
\]

**Corollary 6.3.9.** Let \( W_i \) be the irreducible components of \( V = V(F_1, \ldots, F_{n+m}) \) with the \( F_j \)'s as in (6.13). Additionally assume that \( V(F_1, \ldots, F_m) \) has pure codimension \( m \) and is Cohen-Macaulay:

\[
\sum \ell(\mathcal{O}_{W_i, V}) \deg(W_i) \leq \binom{n}{n-m} D^n (D-1)^{n-m},
\]

where \( V = V(F_1, \ldots, F_{n+m}) \) and \( W_i \) are the irreducible components of \( V \).
Bibliography


Chapter 7

Conclusion

The main problem considered in this thesis is the computation of characteristics classes of algebraic varieties. Our study of this problem begins with considering the case of subschemes of \( \mathbb{P}^n \) over an algebraically closed field of characteristic zero. In this setting we develop algorithms to compute the Segre class, the Chern-Schwartz-MacPherson (\( c_{SM} \)) class and the Euler characteristic with a computer algebra system using standard tools such as Gröbner basis or polynomial homotopy continuation. These methods could in turn be used to compute other invariants such as the Chern-Fulton class, the Milnor class and the Chern-Fulton-Johnson class. We then extend the methods developed for subschemes of \( \mathbb{P}^n \) to subschemes of \( \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m} \). Additionally we give a combinatorial algorithm to compute the \( c_{SM} \) class and Euler characteristic of complete simplicial toric varieties.

In Chapter 2 we present algorithms to compute the Segre class, and the Chern-Schwartz-MacPherson class and Euler characteristic in \( \mathbb{P}^n \). The key component of our construction of these algorithms is Theorem 2.3.1 in which we give a new expression for the projective degrees of a rational map. This theorem is then applied to give Algorithm 2.3.1 which gives a method to compute the projective degrees of a rational map on a computer algebra system using symbolic or numeric methods. Running time bounds for Algorithm 2.3.1 are given in Proposition 2.4.1.

Algorithm 2.3.1 is then used in conjunction with results of Aluffi [2] to construct
Algorithm 2.3.2 which computes the Segre class \( s(V, \mathbb{P}^n) \). Algorithm 2.3.2 is tested on a variety of examples and is found to perform favourably in comparison to other algorithms in most cases; the results of this testing were presented in Table 2.1. A run time bound for our algorithm to compute the Segre class is given in Corollary 2.4.2. To the best of our knowledge this is the first running time bound given on an algorithm which computes the Segre class of a subscheme of \( \mathbb{P}^n \).

Using the inclusion/exclusion property of \( c_{SM} \) classes, a result of Aluffi [1] and the result of Theorem 2.3.1 in the form of Algorithm 2.3.1 we then construct Algorithm 2.3.3 which computes the Chern-Schwartz-MacPherson class and Euler characteristic of \( V \), a subscheme of \( \mathbb{P}^n \). The running time of this algorithm is compared to those of other algorithms that compute the \( c_{SM} \) class or the Euler characteristic in projective space in Table 2.2; in all cases considered Algorithm 2.3.3 performs favourably. We give a run time bound for our algorithm to compute the \( c_{SM} \) class and Euler characteristic in Corollary 2.4.3, this is the first running time bound on an algorithm to compute these objects in \( \mathbb{P}^n \) which is known to us.

In Chapter 3 we give a method to compute the \( c_{SM} \) class of certain complete intersection subschemes of \( \mathbb{P}^n \) without using inclusion/exclusion, the key result needed for this method is proved in Theorem 3.2.1. More specifically in Theorem 3.2.1, starting from a result of Fullwood [4], we prove an expression for the \( c_{SM} \) class of a global complete intersection subscheme \( V = V(f_0, \ldots, f_r) \) of \( \mathbb{P}^n \) where \( V(f_0, \ldots, f_{r-1}) \) is smooth in terms of the Segre class of the singularity subscheme of \( V \). This leads to a direct algorithm (i.e. without inclusion/exclusion) to compute the \( c_{SM} \) class and Euler characteristic in this case; this is presented in Algorithm 3.2.1. The main computational cost of this algorithm is the computation of the Segre class of the singularity subscheme of \( V \), which is done using Algorithm 2.3.2. Algorithm 3.2.1 is extended to any complete intersection subscheme using a type of inclusion/exclusion which considers only the defining equations of \( V \) which give singular hypersurfaces in Algorithm 3.2.2. These algorithms seem to offer improved performance for many cases where they are applicable, with the biggest performance gains being in cases where a large majority of the equations \( f_0, \ldots, f_r \) define a smooth hypersurface.
In Chapter 4 all the above mentioned algorithms to compute characteristics classes are extended to subschemes of a product of projective spaces $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ over an algebraically closed field of characteristic zero. To extend the algorithms we first prove Theorem 4.2.1 which gives an expression for the Segre class in this setting in terms of the projective multi-degrees. We then prove Theorem 4.2.2 which allows us to compute these projective multi-degrees using a computer algebra system. In Algorithm 4.3.1 we use the result of Theorem 4.2.2 combined with the result of Theorem 4.2.1 to construct an algorithm to compute the Segre classes $s(V, \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})$ for $V$ a subscheme of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$. Running time bounds for this algorithm are given in Proposition 4.4.1. In Table 4.1 we compare Algorithm 4.3.1 to another algorithm applicable in this setting, for all examples tested we find that our algorithm performs favourably and in many cases the difference is quite substantial.

In Algorithm 4.3.2 we give a method to compute $c_{SM}(V)$ and $\chi(V)$ for $V$ a subscheme of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$. This algorithm extends Algorithm 2.3.3 to the multi-projective setting. A running time bound for the multi-projective algorithm is given in Corollary 4.4.2. Since other algorithms which compute the $c_{SM}$ class and Euler characteristic in this setting are not known to us we are not able to offer any comparisons. We also extended Algorithm 3.2.1 to compute $c_{SM}$ classes and Euler characteristics for complete intersection subschemes of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$ in Algorithm 4.3.2. Similar to the situation in $\mathbb{P}^n$ Algorithm 4.3.2 offers improved performance over inclusion/exclusion in many cases, particularly where a large majority of the equations of the subscheme being considered define a smooth hypersurface.

For complete simplicial toric varieties $X_\Sigma$ defined by a fan $\Sigma$ we give a new algorithm (Algorithm 5.3.1) to compute the Chern-Schwartz-MacPherson class $c_{SM}(X_\Sigma)$ and the Euler characteristic $\chi(X_\Sigma)$ and an additional algorithm to compute only the Euler characteristic (Algorithm 5.3.2) in Chapter 5. These algorithms are strictly combinatorial and depend only on the fan $\Sigma$. The algorithms are based on a result of Barthel, Brasselet and Fieseler [3] which gives an expression for the $c_{SM}$ class of a toric variety in terms of torus orbit closures. Algorithm 5.3.2 offers improved performance in comparison to Algorithm 5.3.1 when one wishes only to compute
the Euler characteristic.

Taken together the algorithms presented here offer the capability to effectively compute several important topological invariants in a wide variety of settings. As such we believe that this work will be of utility to a wide audience among both those considering applied problems and those computing test cases while working on problems in pure mathematics.

The Macaulay2 [5] implementations of all algorithms discussed in this thesis can be found at https://github.com/Martin-Helmer/char-class-calc. The Macaulay2 [5] implementation for subschemes of $\mathbb{P}^n$ is also available as part of the “CharacteristicClasses” package in Macaulay2 version 1.7 and above and can be accessed using the option “Algorithm=ProjectiveDegree”, see the Macaulay2 documentation http://www.math.uiuc.edu/Macaulay2/doc/Macaulay2-1.7/share/doc/Macaulay2/CharacteristicClasses/html/ for further details. In the near future we hope to make the other algorithms presented here available in the Macaulay2 [5] “CharacteristicClasses” package as well.

In the last chapter, Chapter 6, of the thesis we present several Bézout-like bounds in multi-projective space. These bounds are structured so that they may be easily used to bound the degree of a polynomial system which has a natural multi-projective structure. This type of structure may arise, for example, when trying to find the critical locus of an algebraic variety using the method of Lagrange multipliers. These results are stated in a manner that makes them easy to apply in combination with known running time bounds for solving polynomial systems. In particular these Bézout-like bounds in multi-projective space could allow one to obtain a refined degree bound for polynomial system with a natural multi-projective structure with the refined bound being sharper (in large degree) than a bound obtained by ignoring the multi-projective structure and using the usual Bézout bound in projective space.
Bibliography


Appendix A

Overview of Implementations and Lists of Examples

In this appendix we give a brief overview of the methods provided by the Macaulay2 [2] implementations of the algorithms described in this thesis. We also list the examples used for testing in Tables 2.1, 2.2, 4.1, 4.2 and 4.3. The examples are given in the form of plain text Macaulay2 [2] input.

A.1 Overview of the Implementation used in Chapter 2 and Chapter 3

In this section we briefly describe our M2 package “CharClassCalc” available at https://github.com/Martin-Helmer/char-class-calc which was used for testing in Chapter 2 and Chapter 3.

Let $k[x_0, \ldots, x_n]$ be the coordinate ring of $\mathbb{P}^n$ and let $I$ be any homogeneous ideal in $k[x_0, \ldots, x_n]$, in practice $k$ will often be a finite field of large prime characteristic. All methods represent the Chow ring of $\mathbb{P}^n$ as

$$A^*(\mathbb{P}^n) \cong \mathbb{Z}[h]/(h^{n+1}).$$

The M2 package “CharClassCalc” provides the following methods:
• Segre
  - Takes as input a homogeneous ideal $I$ in $k[x_0, \ldots, x_n]$, in the form $\text{Segre}(I)$.
  - The method of Algorithm 2.3.2 is used for all computations.
  - Outputs the Segre class $s(V(I), \mathbb{P}^n)$ as an element in the Chow ring $\mathbb{Z}[h]/(h^{n+1})$.

• CSM
  - Takes as input a homogeneous ideal $I$ in $k[x_0, \ldots, x_n]$, in the form $\text{CSM}(I)$.
  - By default the method of Algorithm 2.3.3 which uses inclusion/exclusion is used.
  - Optionally we may use the method of Algorithm 3.2.2 with syntax $\text{CSM}(I, \text{Alg}=> \text{Composite})$,
    when $V(I)$ is a complete intersection. The implementation automatically checks if the input is a complete intersection when the argument $\text{Alg}=> \text{Composite}$ is given.
  - Outputs $c_{SM}(V(I))$ as an element in the Chow ring $\mathbb{Z}[h]/(h^{n+1})$.

• Euler
  - Takes as input a homogeneous ideal $I$ in $k[x_0, \ldots, x_n]$, in the form $\text{Euler}(I)$.
  - By default the method of Algorithm 2.3.3 which uses inclusion/exclusion is used.
  - Optionally we may use the method of Algorithm 3.2.2 with syntax $\text{Euler}(I, \text{Alg}=> \text{Composite})$,
    when $V(I)$ is a complete intersection. The implementation automatically checks if the input is a complete intersection when the argument $\text{Alg}=> \text{Composite}$ is given.
posite is given.

- Outputs the integer $\chi(V(I))$.

All the methods described above accept the optional argument “Method”. By default this is set to Method => VspaceDim, which uses symbolic methods to find the vector space dimension described in Theorem 2.3.1. Alternatively numerical methods may be used (via Bertini [1]) with the option Method => Num. For example to find the Segre class using Bertini [1] we may run

```
Segre(I, Method => Num).
```

Using the argument Method => VspaceDim will run exactly the same method as using no optional arguments, i.e. the default symbolic method is used in either case.

### A.2 Examples From Chapter 2

For the examples from Chapter 2 we assume that the function to compute Segre classes is named Segre and the function to compute $c_{SM}$ classes is named CSM. This is the convention used in our M2 package “CharClassCalc” available at [https://github.com/Martin-Helmer/char-class-calc](https://github.com/Martin-Helmer/char-class-calc).

Below are the examples listed in Table 2.1 which are used for testing the performance of Algorithm 2.3.2, our algorithm for computing the Segre class of a projective variety.

```plaintext
// Segre Examples

needsPackage "CharClassCalc"
TEST ///
-- Rational Normal curve in P^7
n=7; R=ZZ/32749[y_0..y_n];
M = matrix{{y_0..y_n},{y_1..y_n,y_0}};
I=minors(2,M);
time Segre(I)

TEST ///
-- Segre embedding of P^2 x P^3 in P^11
n=11; R=ZZ/32749[x_0..x_n];
M = matrix{{x_0,x_1,x_2,x_3},{x_4,x_5,x_6,x_7},{x_8,x_9,x_10,x_11}};
I=minors(2,M);
time Segre(I)

TEST ///
-- Smooth degree 81 variety in P^7
```
n=7; R=ZZ/32749[y_0..y_n];
I = ideal(random(3,R),random(3,R),random(3,R),random(3,R));
time Segre(I)

///
TEST ///
--Degree 10 variety in P^8
n=8; R=ZZ/32749[y_0..y_n];
M = matrix{{random(1,R),random(1,R),random(1,R),random(1,R)},
{random(1,R),random(1,R),random(1,R),random(1,R)},
{random(1,R),random(1,R),random(1,R),random(1,R)});
I=minors(2,M);
time Segre(I)

///
TEST ///
--Degree 21 variety in P^9
n=9; R=ZZ/32749[x_0..x_n];
I=ideal((4*x_3*x_2*x_4-x_0^3)*x_1^3,x_5^3*(x_0*x_1*x_4-x_2^3),
x_9^3*(x_7^3*x_6^3-x_4^3)-x_7^5*x_0^3,
7*x_1^3*(x_2^2*x_3^2-x_6^3)-3*x_2^3*x_9^3);  
time Segre(I)

///
TEST ///
--Degree 48 variety in P^6
n=6; R=ZZ/32749[x_0..x_n];
M = matrix{{x_1*x_4^2*x_3-x_1^4,random(4,R),random(4,R)},
{random(4,R),x_0^2*x_5^2-x_6^3*x_0,random(4,R)});
I=minors(2,M);
time Segre(I)

Below are the examples listed in Table 2.2 which are used for testing the performance of Algorithm 2.3.3, our algorithm for computing the c_{S,M} class of a projective variety.

--CSM Examples
---

TEST ///
--Twisted Cubic
n=3; R=ZZ/32749[x_0..x_n];
K=ideal(x_1*x_3-x_2^2, x_2*x_0-x_3^2,x_1*x_0-x_2*x_3)  
time CSM K

///
TEST ///
--Segre embedding of P^1xP^2 in P^5
n=5; R=ZZ/32749[y_0..y_n];
I=ideal(y_0*y_4-y_1*y_3,y_0*y_5-y_2*y_3,y_1*y_5-y_4*y_2);  
time CSM I

///
TEST ///
--Smooth degree 8 variety in P^4
n=4; R=ZZ/32749[z_0..z_n];
K=ideal(-11796*z_2^2 + 2761*z_0*z_1 + 10725*z_1^2 - 11900*z_0*z_2 -
11598*z_1^2 + 11286*z_2^2 + 5210*z_0*z_3 - 7485*z_1*z_3 + 11208*z_2*z_3 +
5247*z_3^2 - 4745*z_0*z_4 - 15915*z_1*z_4 + 14229*z_2*z_4 - 11236*z_3*z_4 +
16583*z_4^2, 6934*z_0^2 - 10377*z_0*z_1 + 9604*z_1^2 - 4343*z_0*z_2 - 10848*z_1*z_2 -
16357*z_2^2 + 8747*z_0*z_3 - 13140*z_1*z_3 - 7136*z_2*z_3 + 3115*z_3*z_4 -
3741*z_0*z_4 + 14969*z_1^2*z_4 + 10956*z_2^2*z_4 - 10016*z_3^2*z_4 + 13449*z_4^2,
\begin{align*}
&12153z_0^2 - 4789z_0z_1 - 9183z_1^2 - 15107z_0z_2 - 5045z_1z_2 + \\
&6862z_2^2 - 13665z_0z_3 + 4455z_1z_3 - 3129z_2z_3 + 14146z_3^2 - \\
&1424z_0z_4 + 11305z_1z_4 + 4882z_2z_4 - 14665z_3z_4 - 10270z_4^2
\end{align*}

time CSM(I)

---

**Smooth degree 4 variety in \( \mathbb{P}^{10} \)**

\( n=10; R=\mathbb{Z}/32749[x_0..x_n]; \)

\( I=\text{ideal}(\text{random}(2,R),\text{random}(2,R)); \)

time CSM(I)

---

**Smooth degree 6 variety in \( \mathbb{P}^{7} \)**

\( n=7; R=\mathbb{Z}/32749[y_0..y_n]; \)

\( I=\text{ideal}(2y_0^3+12y_1^3+96y_2^3 + 19y_3^3+12y_4^3+y_6^3+5y_7^3, \text{random}(2,R)); \)

time CSM(I)

---

**Degree 12 hypersurface in \( \mathbb{P}^{3} \)**

\( n=3; R=\mathbb{Z}/32749[x_0..x_n]; \)

\( I=\text{ideal}(x_2^6x_3^6+3x_1^2x_2^4x_3^4x_0^2+3x_1^4x_2^2x_3^2x_0^4-3x_2^4x_3^4x_0^4+ \\
&x_1^6x_0^6+21x_1^2x_2^2x_3^2x_0^6-3x_1^4x_0^8+3x_2^2x_3^2x_0^8+3x_1^2x_0^{10}-x_0^{12}) \)

time CSM(I)

---

**Degree 3 variety in \( \mathbb{P}^{8} \)**

\( n=8; R=\mathbb{Z}/32749[x_0..x_n]; \)

\( M = \text{matrix}([\text{random}(1,R),\text{random}(1,R),\text{random}(1,R)], [\text{random}(1,R),\text{random}(1,R),\text{random}(1,R)]); \)

\( I=\text{minors}(2,M); \)

time CSM(I)

---

**Degree 16 variety in \( \mathbb{P}^{10} \)**

\( n=10; R=\mathbb{Z}/32749[x_0..x_n]; \)

\( M = \text{matrix}([x_0-x_1,22x_3-35x_9-13x_2,x_9-x_7+5x_3], \\
&[x_8+9x_0+4x_1,7x_1-33x_5+23x_6,\text{random}(1,R)]; \)

\( I=\text{minors}(2,M); \)

time CSM(I)

---

**Degree 16 variety in \( \mathbb{P}^{5} \)**

\( n=5; R=\mathbb{Z}/32749[x_0..x_n]; \)

\( I=\text{ideal}(4x_3^3x_2^2x_0^3x_1^5, \text{random}(1,R)); \)

time CSM(I)

---

### A.3 Examples From Chapter 3

Below are the examples listed in Tables 3.1 and 3.2 which are used for testing the performance of Algorithm 3.2.1 and Algorithm 3.2.2, our algorithms for computing the \( c_{SM} \) class described in Chapter 3.
restart
needsPackage "CharClassCalc"
}
n=7;
kk=ZZ/32749;
R=kk[x_0..x_n];
I=ideal(21*x_0^2 + 5*x_1^2 - 24*x_2^2+ 13*x_3^2 + 8*x_4^2 -
106*x_5^2 + 2*x_6^2 +14*x_7^2,x_1^2*x_5-x_0^2*x_4)
time CSM(I)
time CSM(I,Alg=>Composite)
```
needsPackage "CharClassCalc"

n=6;
kk=ZZ/32749;
R=kk[x_0..x_n];
I=ideal(3*x_0°2 + 17*x_1°2 - 47*x_2°2 + 3*x_3°2 + 38*x_4°2 -
727*x_5°2 + 12*x_6°2, x_0°x_6 + x_0°2, 43*x_0°2 + 52*x_0°x_1 + 94*x_1°2 + 5*x_0°x_2 + 13*x_1°x_2 + 2*x_2°2 +
x_0°x_3 + 4*x_1°x_3 + 98*x_2°x_3 + 2*x_3°2 + x_0°x_4 + 74*x_1°x_4 +
13*x_2°x_4 + 71*x_3°x_4 + 23*x_4°2 + 12*x_0°x_5 + 2*x_1°x_5 + x_2°x_5 + 65*x_3°x_5 + 92*x_4°x_5 + 27*x_5°2 + 5*x_0°x_6 + 103*x_1°x_6 +
38*x_2°x_6 + x_3°x_6 + 6*x_4°x_6 + 2*x_5°x_6 + 95*x_6°2);
time CSM(I)
time CSM(I,Alg=>Composite)

TEST ///
--V6
{
  restart
  needsPackage "CharClassCalc"
}n=10;
kk=ZZ/32749;
R=kk[x_0..x_n];
I=ideal(random(2,R), random(2,R), random(2,R));
time CSM(I)
time CSM(I,Alg=>Composite)

TEST ///
--V7
{
  restart
  needsPackage "CharClassCalc"
}n=10;
kk=ZZ/32749;
R=kk[x_0..x_n];
I=ideal(random(2,R), random(2,R), 5*x_0°(x_1-x_0)+17°x_0°2-x_0°x_0);
time CSM(I)
time CSM(I,Alg=>Composite)

TEST ///
--V8
{
  restart
  needsPackage "CharClassCalc"
}n=4;
kk=ZZ/32749;
R=kk[x_0..x_n];
I=ideal(-3*x_0°2 + 4*x_1°2 + x_2°2 + x_3°2 + 7°x_4°2 - 9°x_0°3 +
43°x_1°3 + x_2°3 - 98°x_3°3 + 7°x_4°3, (x_1 - x_0)°x_4, x_1°x_0);
time CSM(I)
time CSM(I,Alg=>Composite)

///
A.4 Overview of the Implementation used in Chapter 4

In this section we briefly describe our M2 package “MultiProjChar” available at https://github.com/Martin-Helmer/char-class-calc which was used for testing in Chapter 4.

Let $R$ denote the graded coordinate ring of $\mathbb{P}^n_1 \times \cdots \times \mathbb{P}^n_m$ and let $I$ be any multi-homogeneous ideal in $R$. The M2 package “MultiProjChar” provides the following methods:

- **ChowRing**
  - Takes as input the graded coordinate $R$ in the form
    
    \[
    \text{ChowRing}(R).
    \]
  - Outputs the Chow ring $A^*(\mathbb{P}^n_1 \times \cdots \times \mathbb{P}^n_m)$ as a quotient ring which is isomorphic to $\mathbb{Z}[h_1, \ldots, h_m]/(h_1^{n_1+1}, \ldots, h_m^{n_m+1})$.

- **Segre**
  - Takes as input a multi-homogeneous ideal $I$ in $R$, in the form
    
    \[
    \text{Segre}(I).
    \]
  - Optionally the Chow ring can be input to allow the user to easily perform further computations with the result.
\text{segregClass} = \text{Segre}(A, I).

In this case segregClass is an element of the input Chow ring $A$.

\begin{itemize}
\item The method of Algorithm 4.3.1 is used for all computations.
\item Outputs the Segre class $s(V(I), \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})$ as an element in the Chow ring $A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})$.
\end{itemize}

\textbf{CSM}

\begin{itemize}
\item Takes as input a multi-homogeneous ideal $I$ in $R$, in the form \text{CSM}(I).
\item Optionally the Chow ring can be input to allow the user to easily perform further computations with the result \text{csmClass} = \text{CSM}(A, I).
\end{itemize}

In this case csmClass is an element of the input Chow ring $A$.

\begin{itemize}
\item By default the method of Algorithm 4.3.2 which uses inclusion/exclusion is used.
\item Optionally we may use the method of Algorithm 4.3.3 with syntax \text{CSM}(I, \text{Method}=> \text{DirectCompleteInt}),
\end{itemize}

when $V(I)$ is a complete intersection which satisfies the assumptions of Theorem 4.2.3. The implementation automatically checks if these assumptions are satisfied by the input, if the input does not satisfy the assumptions the inclusion/exclusion method of 4.3.2 is used.

\begin{itemize}
\item Outputs $c_{SM}(V(I))$ as an element in the Chow ring $A^*(\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m})$.
\end{itemize}

\textbf{Euler}

\begin{itemize}
\item Takes as input a multi-homogeneous ideal $I$ in $R$, in the form \text{Euler}(I).
\item By default the method of Algorithm 4.3.2 which uses inclusion/exclusion is used.
\end{itemize}
• Optionally we may use the method of Algorithm 4.3.3 with syntax

\texttt{Euler(I,Method=> DirectCompleteInt)},

when \( V(I) \) is a complete intersection which satisfies the assumptions of Theorem 4.2.3. The implementation automatically checks if these assumptions are satisfied by the input, if the input does not satisfy the assumptions the inclusion/exclusion method of 4.3.2 is used.

• Outputs the integer \( \chi(V(I)) \).

• Alternatively the method Euler will accept as input a previously computed \( c_{SM} \) class in the form of quotient ring element such as

\texttt{Euler(CSM(I))},

or else

\begin{verbatim}
csm=CSM(I)
EC=Euler(csm)
\end{verbatim}

this method will be much faster if one has already computed the \( c_{SM} \) class.

\section{A.5 Examples From Chapter 4}

For the examples from Chapter 4 we assume that the function to compute Segre classes using Algorithm 4.3.1 is named Segre and the function to compute \( c_{SM} \) classes is named CSM and that when no options are given the inclusion/exclusion method of Algorithm 4.3.2 is used. Further we assume that the method of Algorithm 4.3.3 to compute the \( c_{SM} \) class of certain complete intersections without using inclusion/exclusion is accessed by calling the CSM method with the option Method\( \Rightarrow \) DirectCompleteInt. This is the convention used in our M2 package “MultiProjChar” available at \url{https://github.com/Martin-Helmer/char-class-calc}.

```
-- Multi-projective Segre Examples
-----------------------------------------------
-- For all the examples below we would need to execute the command
needsPackage "MultiProjChar"

-- For brevity we will also use the "NormalToricVarieties" package to build the multi-projective spaces
-- in the manner
```
-- \( P^n \times P^m = \text{projectiveSpace}(n, \text{CoefficientRing} => \mathbb{Z}/32749)^{\times} \text{projectiveSpace}(m, \text{CoefficientRing} => \mathbb{Z}/32749) \)
-- and so on.
-- Hence we also need to run

\begin{verbatim}
needsPackage "NormalToricVarieties"
\end{verbatim}

-- Note that we could equally well construct the graded coordinate ring of each
-- multi-projective space directly and not use the "NormalToricVarieties" package.

Below are the examples listed in Table 4.1 which are used for testing the performance of Algorithm 4.3.1, our algorithm for computing the Segre class of a subscheme of \( P^n \times \cdots \times P^m \).

\begin{verbatim}
TEST ///
-- codimension 3 in \( P^2 \times P^3 \)
kk=\mathbb{Z}/32749;
X=\text{projectiveSpace}(2, \text{CoefficientRing} => kk)\^{\times} \text{projectiveSpace}(3, \text{CoefficientRing} => kk);
R=\text{ring} X;
I = \text{ideal}(\text{random}(2,0) R, R_3 R_5 R_4^2, R_0 R_1);
time \text{Segre}(I)
///
TEST ///
-- codimension 2 in \( P^1 \times P^1 \times P^1 \)
kk=\mathbb{Z}/32749;
X=\text{projectiveSpace}(1, \text{CoefficientRing} => kk)^{\times} \text{projectiveSpace}(1, \text{CoefficientRing} => kk)^{\times} \text{projectiveSpace}(1, \text{CoefficientRing} => kk);
R=\text{ring} X;
I = \text{ideal}(\text{random}(2,1,1) R, R_3 R_5 R_4^2 - R_2 R_5^3);
time \text{Segre}(I)
///
TEST ///
-- codimension 2 in \( P^3 \times P^2 \)
kk=\mathbb{Z}/32749;
X=\text{projectiveSpace}(3, \text{CoefficientRing} => kk)^{\times} \text{projectiveSpace}(2, \text{CoefficientRing} => kk);
R=\text{ring} X;
I = \text{ideal}(R_0 R_1 R_2 - R_2^2 R_3, R_0 R_2 R_1 R_3);
time \text{Segre}(I)
///
TEST ///
-- hypersurface in \( P^5 \times P^3 \)
kk=\mathbb{Z}/32749;
X=\text{projectiveSpace}(5, \text{CoefficientRing} => kk)^{\times} \text{projectiveSpace}(3, \text{CoefficientRing} => kk);
S=\text{ring} X;
I = \text{ideal}(R_0^2 R_7^2 - R_6 R_7 R_1 R_2);
time \text{Segre}(I)
///
TEST ///
-- codimension 2 in \( P^2 \times P^3 \times P^1 \)
kk=\mathbb{Z}/32749;
X=\text{projectiveSpace}(2, \text{CoefficientRing} => kk)^{\times} \text{projectiveSpace}(3, \text{CoefficientRing} => kk)^{\times} \text{projectiveSpace}(1, \text{CoefficientRing} => kk);
R=\text{ring} X;
I = \text{ideal}(R_0 R_1 R_3 - 5 R_4 R_2^2, R_4 R_6);
time \text{Segre}(I)
///
TEST ///
-- codimension 2 in \( P^2 \times P^2 \times P^2 \)
kk=\mathbb{Z}/32749;
X=\text{projectiveSpace}(2, \text{CoefficientRing} => kk)^{\times} \text{projectiveSpace}(2, \text{CoefficientRing} => kk)^{\times} \text{projectiveSpace}(2, \text{CoefficientRing} => kk);
R=\text{ring} X;
I = \text{ideal}(R_0 R_3 R_6, R_5 R_7 - 7 R_4 R_8);
time \text{Segre}(I)
///
TEST ///
-- codimension 2 in \( P^4 \times P^3 \times P^3 \)
kk=\mathbb{Z}/32749;
X=\text{projectiveSpace}(4, \text{CoefficientRing} => kk)^{\times} \text{projectiveSpace}(3, \text{CoefficientRing} => kk)^{\times} \text{projectiveSpace}(3, \text{CoefficientRing} => kk);
R=\text{ring} X;
I = \text{ideal}(R_0 R_5^2 - R_2 R_5 R_6, R_8 R_13 - 6 R_5 R_14);
time \text{Segre}(I)
///
\end{verbatim}
Below are the examples listed in Table 4.2 which are used for testing the performance of Algorithm 4.3.2, our algorithm for computing the $c_{SM}$ class of a subscheme of $\mathbb{P}^n \times \cdots \times \mathbb{P}^m$ using inclusion/exclusion.

\[
\text{---Multi-projective CSM via inclusion/exclusion---}
\]
\[
\text{TEST} \quad \text{---codimension 2 in $\mathbb{P}^2 \times \mathbb{P}^2$} \\
\text{kk=ZZ/32749;} \\
\text{X=projectiveSpace(2,CoefficientRing =>kk)**projectiveSpace(2,CoefficientRing =>kk);} \\
\text{R=ring X;} \\
\text{I=ideal(random({1,1},R), R_0^2*R_5^2-R_1*R_2*R_4*R_5);} \\
\text{time CSM(I)} \quad \text{--- Segre I} \\
\]

\[
\text{TEST} \quad \text{---codimension 2 in $\mathbb{P}^5 \times \mathbb{P}^2$} \\
\text{kk=ZZ/32749;} \\
\text{X=projectiveSpace(5,CoefficientRing =>kk)**projectiveSpace(2,CoefficientRing =>kk);} \\
\text{R=ring X;} \\
\text{I=ideal(4*R_0*R_6-7*R_7*R_2, R_0*R_4*R_8);} \\
\text{time CSM(I)} \quad \text{--- Segre I} \\
\]

\[
\text{TEST} \quad \text{---codimension 2 in $\mathbb{P}^2 \times \mathbb{P}^2 \times \mathbb{P}^3$} \\
\text{kk=ZZ/32749;} \\
\text{X=projectiveSpace(2,CoefficientRing =>kk)**projectiveSpace(2,CoefficientRing =>kk)**projectiveSpace(3,CoefficientRing =>kk);} \\
\text{R=ring X;} \\
\text{I=ideal((R_0*R_1-R_2^2)*R_4, R_5*(R_6^2-R_7*R_6));} \\
\text{time CSM(I)} \quad \text{--- Segre I} \\
\]

\[
\text{TEST} \quad \text{---codimension 3 in $\mathbb{P}^2 \times \mathbb{P}^2 \times \mathbb{P}^3$} \\
\text{kk=ZZ/32749;} \\
\text{X=projectiveSpace(2,CoefficientRing =>kk)**projectiveSpace(2,CoefficientRing =>kk)**projectiveSpace(3,CoefficientRing =>kk);} \\
\text{R=ring X;} \\
\text{I=ideal((R_0*R_1-R_2^2)*R_4, R_5*(R_6^2-R_7*R_6), R_0*R_3^2);} \\
\text{time CSM(I)} \quad \text{--- Segre I} \\
\]

Below are the examples listed in Table 4.3 which are used for testing the performance of Algorithm 4.3.3, our algorithm for computing the $c_{SM}$ class of certain subschemes of $\mathbb{P}^n \times \cdots \times \mathbb{P}^m$ without using inclusion/exclusion.

\[
\text{TEST} \quad \text{---codimension 3 in $\mathbb{P}^2 \times \mathbb{P}^2$} \\
\text{kk=ZZ/32749;} \\
\text{X=projectiveSpace(2,CoefficientRing =>kk)**projectiveSpace(2,CoefficientRing =>kk);} \\
\text{R=ring X;} \\
\text{A=ChowRing(R);} \\
\text{I=ideal(random({1,1},R), random({1,1},R), R_1*R_0*R_3-R_0^2*R_4);} \\
\text{csm=time CSM(A,I);} \\
\text{csm2=time CSM(A,I,Method=>DirectCompleteInt)} \quad \text{--- Segre I} \\
\]
A.6 Overview of the Implementation used in Chapter 5

In this section we briefly describe our M2 package “CharToric” available at https://github.com/Martin-Helmer/char-class-calc which was used for testing in Chapter 5.

Let $X = X_\Sigma$ be the complete simplicial toric variety of a fan $\Sigma$ defined using the “NormalToricVarieties” M2 package [2]. The M2 package “CharToric” provides the following methods:

- **ChowRing**
  - Takes as input the complete simplicial toric variety $X$, in the form
    
    \[
    \text{ChowRing}(X)
    \]
  - Outputs the Chow ring $A^*(X_\Sigma)_{\mathbb{Q}}$.
  - The Chow ring is found by employing the relation in Proposition 5.2.2 and using built-in methods from the “NormalToricVarieties” package to compute the Stanley-Reisner ideal of the fan $\Sigma$.  

```plaintext
/// TEST ///
-- codimension 2 in $\mathbb{P}^2 \times \mathbb{P}^3$
kk=ZZ/32749;
X=projectiveSpace(2,CoefficientRing =>kk)**projectiveSpace(3,CoefficientRing =>kk);
R=ring X;
I=ideal(random({2,1,1},R),R_1*R_0*R_4);
time CSM(I)

/// TEST ///
-- codimension 3 in $\mathbb{P}^2 \times \mathbb{P}^2 \times \mathbb{P}^2$
kk=ZZ/32749;
X=projectiveSpace(2,CoefficientRing =>kk)**projectiveSpace(2,CoefficientRing =>kk)**projectiveSpace(2,CoefficientRing =>kk);
R=ring X;
I=ideal(random({2,1,0},R), random({0,0,1},R), R_2*R_6-7*R_0*R_7);
csm=time CSM(A,I)
csm2=time CSM(A,I,Method=>DirectCompleteInt)

/// TEST ///
-- codimension 2 in $\mathbb{P}^3 \times \mathbb{P}^2 \times \mathbb{P}^2$
kk=ZZ/32749;
X=projectiveSpace(3,CoefficientRing =>kk)**projectiveSpace(2,CoefficientRing =>kk)**projectiveSpace(2,CoefficientRing =>kk);
R=ring X;
I=ideal(random({2,1,1,1},R),R_2*R_5-7*R_0*R_6);
csm=time CSM(I)
csm2=time CSM(I,Method=>DirectCompleteInt)
```
• **CSMToric**
  
  o Takes as input the complete simplicial toric variety $X$, in the form
    
    $\text{CSMToric}(X)$.
  
  o The method CSMToric implements Algorithm 5.3.1.
  
  o Outputs $c_{SM}(X_\Sigma)$ as an element in the quotient ring presentation of the Chow ring $A^*(X_\Sigma)_\mathbb{Q}$ (see Proposition 5.2.2).

• **EulerToric**
  
  o Takes as input the complete simplicial toric variety $X$, in the form
    
    $\text{EulerToric}(X)$.
  
  o The method EulerToric implements Algorithm 5.3.2.
  
  o Outputs the numerical value $\chi(X_\Sigma)$. 

Bibliography


Curriculum Vitae

Name: Martin Helmer

Education:

University of Western Ontario
London, Ontario, Canada
2011-2015 Ph.D.

Queen’s University
Kingston, Ontario, Canada
2009-2011 M.Sc.

University of Ontario Institute of Technology
Oshawa, Ontario, Canada
2005-2009 B.Sc.

Awards:

Ontario Graduate Scholarship (OGS),

NSERC USRA,
2008.

Publications:


Related Work Experience:

Teaching Assistant
University of Western Ontario

Teaching Assistant
Queen’s University
2009-2011.