Geometric Invariant Theory and 3-Tensors

Benjamin John Leedom

May 2020

A thesis submitted for the degree of Bachelor of Science (Advanced)
of the Australian National University

NATURAM PRIMUM COGNOSCERE RERUM

Australian National University

For John Leedom, Megan Passey, Frankie and Susie Boyle, Stephen Hawkins, Jacqui Lyon and Melinda Smotlak. Sometimes the longest of journeys can start with the smallest of pushes; or in my case, a lot more.

Declaration

The work in this thesis is my own except where otherwise stated.

Benjamin John Leedom

Acknowledgements

Thank you to my parents, whose endless capacity for understanding and support helped push me through what was a fairly tough 18 months. Their generosity and willingness to listen is invaluable, and something that I'm extremely grateful for.

A further thank you must be directed towards my friends - both here in Canberra, in Australia and abroad. The advice, support, companionship have been both a massive help for, and welcome distraction from the work I've done here and elsewhere.

It would also be remiss of me not to mention the wonderful teaching support I've received at ANU, with special thanks to Joan Licata, Vigleik Angeltveit and Jim Borger whose excellent lecturing skills and overall enthusiasm for what they taught helped to both spur my own curiosity, and maintain my enthusiasm for learning the mathematics I am and have been interested in.

Thank you also to Tony Licata, for agreeing to co-supervise this thesis in its early stages, and stepping in where necessary when Anand was on paternity leave. Your ability to interrogate my understanding of core concepts was extremely helpful, and was instrumental in creating a more polished, well rounded thesis.

Finally, thank you to my supervisor, Anand Deopurkar. Without you, this thesis would be a far weaker piece of work. Your endless patience in discussions we have had in the last 18 months speaks to your willingness to invest time in your students, and I cannot thank you enough for continuing to help me during your paternity leave and also agreeing to continue to supervise me as I moved to part-time work this year.

Abstract

This thesis provides an exposition for the basic building blocks of Geometric Invariant Theory. We develop the theory to explain and prove the Hilbert-Mumford Numerical Criterion for the stability of points. We then explain how this Criterion can be manipulated into a convex geometry problem, and discuss an implementation of this problem into code in the Sage Programming Language. Utilising this code, we then find unstable points in some elementary problems before turning to the problem of the action of SL_n^3 on $n \times n \times n$ cubic matrices, for which we provide a potential generic solution for all n.

Contents

Notation and terminology

This section details some basic notation for the thesis:

Notation

k	Refers to an algebraically closed field where the characteristic is zero unless otherwise stated
$\mathcal{O}(X)$	refers to the ring of regular functions for an algebraic variety, although $R(X)$ may also be used explicitly when the variety is projective.
$O(n)$	Used in chapter three and refers to the computational complexity of a piece of code
SL_n	refers to the special linear group of size n, the field k being implicit.
$\mathbb{A}^n$	refers to Affine n-space over a field k which is again implicit
$\mathbb{P}^n$	refers to Projective n-space over k.
X	refers to an algebraic variety, unless otherwise stated.

Chapter 1

Introduction

Consider some kind of geometric object X. It might be a topological space, an algebraic variety or a manifold. Take a group G that acts on X. If we consider the orbit space X/G, it would make sense to try to grant the orbit space similar geometric structures to that of X. However, this is more challenging than it may first appear.
Consider the algebraic variety affine 2-space, or $\mathbb{A}^2$. The group of units k^* for its base field k, acts on $\mathbb{A}^2$ by scalar multiplication. That is, for $\lambda \in k^*$, $(a, b) \in \mathbb{A}^2$, $\lambda(a, b) = (\lambda a, \lambda b)$. Consider the orbit space of $\mathbb{A}^2$ under this action. This does not have the structure of an algebraic variety. We see that the point representing the orbit containing $(1, 1)$ also contains the point $(0, 0)$ in its closure, even though $(0, 0)$ is not contained in the orbit of $(1, 1)$.

Thus, it is clear that the orbit space is not a sufficient structure to be entertained as a quotient. This is one simple example where the orbit space does not preserve the "categorical" geometry of the original object. Hence, a new concept of what the quotient should be is required. Via utilising the Yoneda lemma, we can arrive at the categorical quotient, but this comes with its own difficulties.

The restrictive case of a reductive group acting on an affine or projective variety, elucidates one such problem. Namely, that while the categorical quotient will be contained in the category of algebraic varieties, its geometry can be vastly different from the original object, and much of the structure can be lost. In the case of our example where k^* acts on $\mathbb{A}^2$, $X//G$ is only a single point, 0.

Now, consider $X = \mathbb{A}^2 \setminus 0$. Let the group and action be as before. That is, k^* acting by scalar multiplication. Here $X//G = \mathbb{P}^1$, which is the same as the set of orbits. Thus, we see that removing zero turns $X//G$ from a geometrically unsastifying object, into the orbit space. This is because 0 is an "unstable point".

The finding and algebro-geometric classification of such unstable points is one of the main purposes of Geometric Invariant Theory, or GIT. It is also, in specific cases, the main subject of this thesis. In particular, a potential solution is presented for the problem of the unstable points for the action of SL_n^3 on $n \times n \times n$ tensors.

Chapter Two is dedicated to setting up the foundations for the GIT quotient and stable points. It is strongly based in Hoskins work in [6], and utilising the theory of Lie algebras and affine algebraic groups from Humphreys in [7] to describe the affine and projective GIT quotients. With the help of some basic concepts in category theory, we then justify why they are a better definition than the orbit space, following which we build up and discuss the concept of the stability of points. This idea of stability allows us to understand the level of similarity between the GIT quotients and the orbit space. The chapter finishes by using the theory we have extablished, along with some results from linear algebra due to Nagata and Cohen [4] and algebraic completions of curves to prove the Hilbert Mumford Numerical Criterion, a strong theorem that allows for algorithmic identification of point stability.

In Chapter Three, we will use this criterion along with some convex geometry to create a program which we can use to identify which points are unstable in our varieties. We then justify the use of this particular algorithm over a more naive approach within the context of our main example discussed in Chapter Four. Chapter Three is rounded out with the use of this program on some basic examples of unstable point identification relating to homogeneous polynomials.

In Chapter Four, we move onto our main example - the action of SL_n on cubal $n \times n \times n$ matrices. We use the program, and work done by Bhargava and Ho [2] [3] to explore some basis-free descriptions of the unstable points before providing a conjecture for integer generality, of which one direction is proven (points that share this property are unstable).

This thesis should be accessible to anyone with basic understanding of Lie algebras, affine algebraic groups, algebraic varieties, representation theory. The discussion of the complexity of the program in Chapter three may require some basic computer science knowledge.

Chapter 2

GIT Quotients and Stable Points

2.1 Algebraic Groups, Lie Algebras and Reductivity

2.1.1 Algebraic Groups

Before we discuss some actual GIT theory such as the GIT quotient or stable points, we first need to build up a strong background in some algebraic group theory, as well as a brief discussion of Lie Algebras, since they will be useful in proving that in characteristic zero the main group used in this thesis, SL_n, is linearly reductive.

Definition 2.1. An *algebraic group* is an algebriac variety X along with regular maps:

- $\pi : X \times X \to X$
- $\iota : X \to X$

and an element e in X such that

- π satisfies the properties of group multiplication
- ι satisfies the properties of an inverse
- e satsifies the properties of the identity

Examples 2.2. A very simple example is the complex numbers. As a variety it is $\mathbb{A}^1$ where the field k it is defined over is $\mathbb{C}$. π is standard complex number addition, ι is the map $x \to -x$ and the identity is 0.

More importantly for our exploration into GIT, various matrix groups under multiplication are also algebraic. The $n \times n$ matrices are the variety $\mathbb{A}^{n^2}$. Although they are not an algebraic group, GL_n is. Matrix multiplication is a regular map since it is polynomial in the matrix entries, and GL_n is the subvariety of the $n \times n$ matrices with the zariski open condition that the determinant is nonzero. This allows matrix inversion to also be a regular map, and so GL_n is an algebraic group.

Lastly, the main object of study in this thesis, SL_n, is clearly a variety. It is a subvariety of GL_n. The elements that satisfy the zariski closed condition that the determinant of the matrix is 1. Again, the regularity of inversion and multiplication is inherited from the GL_n example.

It is useful to think of GL_n as a fundamental example in this thesis. The thesis exclusively deals with affine algebraic groups. That is, groups that are affine varieties. A standard theorem (as seen in [7]) states that any affine algebraic group, G, is isomorphic to a closed subgroup of some GL_n. Thus, in many circumstances, we can prove properties in GL_n, which is often easier. From here, we can then simply restrict to subgroups, as long as restriction maintains the property that we're proving.

2.1.2 Tangent Spaces

We will see in our discussion of Lie Algebras that the Lie Algebra of an algebraic group G is isomorphic as a vector space to the tangent space of G at the identity. Thus, we prepend our discussion of Lie Algebras with a brief discussion on the tangent space. We will provide three different formulations, each useful for its own purpose, either in proving the existence of the isomorphism with Lie Algebras, or in computing the derivations (that is, the corresponding lie algebra homomorphisms) of various homomorphisms on G. The first formulation in its affine form will be used to compute the Lie Algebra of GL_n and SL_n, whilst its more general form as a local ring construction and the formulation of the tangent space as the set of point derivations will be used to prove the vector space isomorphism with Lie Algebras. We will introduce a final formulation of the tangent space in terms of the Dual Numbers in order to make the computation of the derivation of some group homomorphisms into Lie Algebra homomorphisms simpler.

Definition 2.3. For an affine variety $X \subset \mathbb{A}^n$ defined by polynomials $f_i(t_1, \cdots, t_n)$, the *tangent space* $\mathrm{Tan}_x(X)$ at a point $x = (x_1, \cdots, x_n) \in X$ is the affine linear subspace defined as the zero locus of the polynomials

$$d_x f_i = \sum_{j=1}^{n} \frac{df_i}{dt_j}(x)(t_j - x_j)$$

Note that this definition views the tangent space as a vector space with the zero vector being identified with x. We have additive closure since the $d_x f_i$ are all linear maps, and so if u and v are both in the zero loci of the $d_x f_i$, so is $u + v$. If we want to view this tangent space more geometrically, with zero as the zero vector, we simply take the shift map $t_i \to (t_i - x_i)$.

Note that the definition above will only suffice for affine varieties. For varieties more generally, the tangent space is defined via a local ring construction. We will use the above definition to compute the Lie Algebra of SL_n, as SL_n is an affine algebraic group. However, to prove that the traditional definition of the Lie Algebra is equivalent to the definition via the tangent space, we will need to explore this local ring construction somewhat.

Assume X is a variety, let $R = \mathcal{O}(X)$ and let M be the maximal ideal in R consisting of functions that vanish at x. Let $\mathcal{O}_x$ be the ring of functions in R that are non-vanishing at x.

Definition 2.4. For an arbitrary variety X, we can define the *tangent space of X at some x* to be the dual vector space $(m_x/m_x^2)^*$ over $k = \mathcal{O}_x/m_x$ where $m_x = M\mathcal{O}_x$.

Proposition 2.5. *The local ring construction of the tangent space as described above is the same as Definition 2.3 for affine varieties*

Proof. Let X be an affine variety, and let R and M be deefined as before. We can identify R/M with k, and so we can view M/M^2 as a vector space over k, since it is an R/M-module. Since we can think of $d_x f$ (for any functions in $\mathcal{O}(\mathbb{A}^n)$) as a linear function on $\mathbb{A}^n$, and $\mathrm{Tan}(X)_x$ is a vector subspace of $\mathbb{A}^n$, we can also think of $d_x f$ as a linear function on the tangent space. Note that if f is in M, it vanishes on the tangent space, and so $d_x f$ is determined by where f gets sent to in the quotient of $R = \mathcal{O}(\mathbb{A}^n)/I(X)$, $I(X)$ being the polynomial radical defining X. Thus, d_x is a k-linear map from R to the dual space of the tangent space at x. Next, since as a vector space we can write $R = k + M$, and d_x of a constant function is zero, we can think of d_x as a map from M to the dual space of the tangent space. It is then not hard to show that the kernel of d_x is M^2. From here, we note that $\mathcal{O}_x$ is R localised at M, and so we can induced from the inclusion of R into R_M a canocial isomorphism between M/M^2 and m_x/m_x^2. Thus, we arrive at the more general tangent space definition. □

From here, we can move toward an equivalent way to view the tangent space (therefore another construction of the Lie Algebra). We view the tangent space as the set of point derivations $\delta : \mathcal{O}_x \to k$. Viewing the tangent space as the set of point derivations will be useful in proving that the tangent space is indeed equivalent to the Lie Algebra.

Definition 2.6. A *point derivation* $\delta : \mathcal{O}_x \to k$ is a k-linear map satisfying the following equation:

$$\delta(fg) = \delta(f) \cdot g(x) + f(x) \cdot \delta(g)$$

We now prove that the set of point derivations is indeed equivalent to the tangent space.

Proposition 2.7. *The set of point derivations at x are naturally isomorphic to the tangent space at x.*

Proof. It is clear that the set of point derivations form a vector space over k. Now, if f is constant or in m_x^2 it is zero by the above equation (let g be the identity map in the dual space), and so δ is determined by what it does to maps in m_x/m_x^2. Thus, we have an injection from the point derivations into the tangent space. In the other direction, we can take a map $m_x/m_x^2 \to k$ to a map from $m_x \to k$ by composing with the projection map $m_x \to m_x/m_x^2$. We can then extend this map from $m_x \to K$ to a map from $\mathcal{O}_x$ to K by sending constants to zero. It then remains to show that this map satisfies the equation found in the definition of the point derivation.

For $f \in \mathcal{O}_x$, we can write $f = c + g$ where c is a constant function and $g(x) = 0$. Note that we extended $\delta : m_x \to K$ to $\mathcal{O}_x$ by setting $\delta(\text{const}) = 0$, and since δ is k-linear, we have that

$$\begin{aligned}
\delta(f) &= \delta(c + g) \\
&= \delta(c) + \delta(g) \\
&= \delta(g)
\end{aligned}$$

Now:

$$\begin{aligned}\delta(f_1f_2) &= \delta((c_1+g_1)\cdot(c_2+g_2))\\ &= \delta(c_1c_2)+\delta(c_1g_2)+\delta(c_2g_1)+\delta(g_1g_2)\\ &= c_1\delta(g_2)+c_2\delta(g_1)+\delta(g_1g_2)\end{aligned}$$

However, $g_1, g_2 \in m_x$ so $g_1g_2 \in m_x^2$, and so

$$\begin{aligned}\delta(f_1f_2) &= c_1\delta(g_2)+c_2\delta(g_1)\\ &= \delta(f_2)f_1(x)+\delta(f_1)f_2(x)\end{aligned}$$

□

Thus, we have our first three equivalent formulations of the tangent space. Finally, we turn to the definition of the tangent space in terms of the dual numbers:

Definition 2.8. The *Dual Numbers* are the algebra $k[t]/t^2$ but will be referred to notationally as $k[\epsilon]$. That is, elements are $a+b\epsilon$, such that $a, b \in k$, and $\epsilon^2 = 0$.

The key idea here is that we can identify the tangent space at x with the set of $k[\epsilon]$-valued points that satisfy some reasonable conditions. To understand what this means, let X be affine, with $R = \mathcal{O}(X)$. Let f_i be the functions that generate the ideal I whereby $\mathcal{O}(X) = k[t_1, \cdots, t_n]/I$. A point of X, say $x = (c_1, \cdots, c_n)$, $f_i(x) = 0$ has the property that $f_i(x) = 0$ for all f_i. Thus, from x we can derive a map $\alpha_x : R \to k$, where $\alpha_x(t_i) = c_i$. We call x a k-valued point. With this information, we see another way to identify the tangent space as a subset of the $k[\epsilon]$-valued points. In particular, the $k[\epsilon]$-valued points that lift the map $\alpha_x : R \to k$. That is, k-algebra homomorphisms t such that:

$$\begin{tikzcd} R \arrow[r, "t"] \arrow[dr, "x\mapsto c"'] & k[\epsilon] \arrow[d, "\epsilon\mapsto 0"] \\ & k \end{tikzcd}$$

commutes.

Proposition 2.9. *The tangent space at a point x is in bijection with the $k[\epsilon]$-valued points that lift α_x*

Proof. We know that the tangent space at a point X is the vector space $\mathrm{Hom}(m/m^2, k)$. Take a homomorphism in this space, say, h. Now, we need a k-algebra homomorphism from $R \to k[\epsilon]$ that is a lift of α_x. As a k-vector space, we can write our ring of regular functions R as $m \oplus k$. In particular, for any $r \in R$, we have $(r - r(x), r(x))$. Note $r - r(x) \in m$, since $(r - r(x))(x) = 0$. Thus, given h, we take the homomorphism that sends r to $r(x) + h(r - r(x))\epsilon$, where we understand h here to really be h composed with the projection from $m \to m/m^2$. This is clearly a lift of α_x, since if we send ϵ to 0, we get $r(x) = \alpha_x(r)$. It remains to show this is a k-algebra homomorphism. Let $r, s \in R$. Then:

$$\begin{aligned}
r + s \mapsto& (r+s)(x) + h(r + s - (r+s)(x))\epsilon \\
&= r(x) + s(x) + h(r + s - r(x) - s(x))\epsilon \\
&= r(x) + h(r - r(x))\epsilon + s(x) + h(s - s(x))\epsilon
\end{aligned}$$

$$\begin{aligned}
r \cdot s \mapsto& (r(x) + h(r - r(x))\epsilon) \cdot (s(x) + h(s - s(x))\epsilon) \\
&= r(x)s(x) + r(x)h(s - s(x))\epsilon + s(x)h(r - r(x))\epsilon \\
&= rs(x) + h(r(x)s - r(x)s(x))\epsilon + h(s(x)r - r(x)s(x))\epsilon \text{ since h is k-linear} \\
&= rs(x) + h(r(x)s + s(x)r - 2rs(x))\epsilon
\end{aligned}$$

We want $h(r(x)s + s(x)r - 2rs(x)) = h(rs - rs(x))$, so subtracting them should give us zero.

$$\begin{aligned}
h(r(x)s + s(x)r - 2rs(x) - rs + rs(x)) &= h(r(x)s + s(x)r - rs - rs(x)) \\
&= h((r - r(x)) \cdot (s - s(x))) \\
&= h(0)
\end{aligned}$$

since $(r - r(x)) \cdot (s - s(x)) \in m^2$. Thus, we have a k-algebra homomorphism for h that lifts α_x.

For the reverse direction, let $f : R \to k[\epsilon]$ be a k-algebra homomorphism. For some $c \in m/m^2$, choose a lift of c, say $\tilde{c} \in m$. Now, $m \subset R$, so we have $f(\tilde{c}) = \alpha_x(c) + h(\tilde{c})\epsilon$. This h will be our homomorphism from m/m^2 to k. To ensure this however, there are two things we need to check:

1. h is zero on elements in m^2

2. This process is independent of our choice of lift.

For the first, let $c' = cd \in m^2$, $c, d \in m$. Now, $f(c') = f(c)f(d) = h(c)\epsilon h(d)\epsilon = h(c')\epsilon^2 = 0$. However, $h(c')\epsilon = f(c')$, so $h(c') = 0$.
For the second, we note that if we have two separate lifts of c, say $\tilde{c}$ and $\tilde{c}'$, that since both map to c in the quotient m/m^2, their difference $\tilde{c} - \tilde{c'}$ must be contained in m^2. Then:

$$\begin{aligned} f(\tilde{c} - \tilde{c}') &= h(\tilde{c} - \tilde{c}')\epsilon \\ &= 0 \end{aligned}$$

Checking that each compose with each other to give the identity is clear. $\square$

However, the above proof only demonstrates there is a bijection as sets. Ideally, for us to be able to think of this dual numbers construction as a tangent space, it should also have the structure of a vector space. Since k-algebra homomorphisms are k-linear, we see that defining the additive structure as

$$(f_1 + f_2)(r) = r(x) + h_1(r - r(x))\epsilon + h_2(r - r(x))\epsilon$$

where f_1 and f_2 are lifts of α_x derived from h_1 and h_2 and defining scalar multiplication as

$$kf_1(r) = kr(x) + kh_1(r - r(x))$$

satisfy the conditions of a vector space. Thus:

Proposition 2.10. *The numbers identification of the tangent space is a vector space*

With these formulations of the tangent space in hand, we move on to discussing Lie Algebras.

2.1.3 Lie Algebras

Lie Algebras have a broad and rich strucutre useful over many disciplines. However, the purpose of the discussion of Lie Algebras in this thesis is to prove that SL_n has no nontrivial normal subgroups. This is the main fact we will use to prove SL_n is linearly reductive, which is necessary to discuss the GIT examples in this thesis. To expose this structure, we will look at its Lie Algebra, and in particular the relation between subgroups of an algebraic group G and ideals of its Lie Algebra $\mathfrak{g}$. First, we need to understand what a Lie Algebra is, and in particular, how to think about the Lie Algebra of a specific group. Humphreys fantastic book [7] is a fantastic reference and will be adhered to closely in this discussion.

Definition 2.11. A *Lie Algebra* $\mathfrak{g}$ over a field k is a vector space together with a binary operation called the *Lie Bracket*: $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ that satisfies the following properties:

- Bilinearity. That is, $[ax+by, z] = a[x,z]+b[y,z]$ and $[z, ax+by] = a[z,x]+b[z,y]$
- Alternativity. That is $[x,x] = 0$.
- Anticommutativity. That is $[x,y] = -[y,x]$
- The Jacobi Identity. That is $[x,[y,z]] + [z,[x,y]] + [y,[z,x]]$.

For matrix groups, this Lie Bracket will take the form $[x,y] = xy - yx$, and we will assume this is the Lie Bracket operation from here on unless otherwise stated. It will also be useful to understand what homomorphisms between Lie Algebras look like:

Definition 2.12. A *Lie Algebra homomorphism* $\phi : \mathfrak{g} \to \mathfrak{g}'$ is a vector space homomorphism that is compatible with the Lie Bracket. That is,

$$\phi([x,y]) = [\phi(x), \phi(y)]$$

One can assign to every algebraic group G a Lie Algebra $\mathfrak{g}$. The problem with the above definition is that it can be used to identify Lie Algebras, but it doesn't tell us how to make this assignment. We begin by looking at the traditional construction for the Lie Algebra of an algebraic group: the set of left invariant derivations. To understand this construction, we must first understand derivations.

Definition 2.13. Consider an algebra A over a field k. A *derivation* of A is a vector space homomorphism $\delta : A \to A$ such that $\delta(ab) = \delta(a)b + a\delta(b)$. We call the space of derivations Der A.

Now, to define the Lie Algebra of an algebraic group G, we need to note two more important facts. The first is that we can see that we have a left action λ_g of G on the ring of regular functions $\mathcal{O}(G)$: $(g \cdot f)(y) = f(g^{-1}y)$. Respectively, we have a right action ρ_g: $(f \cdot g)(y) = f(yg)$. Thus, we can define the Lie Algebra of G:

Definition 2.14. The *Lie Algebra* of an algebraic group G is the subspace of Der A that are left invariant. That is, the algebra is the set:

$$\{\delta \in \text{Der } A \mid \delta\lambda_g = \lambda_g\delta \text{ for all } g \in G\}$$

with Lie bracket $[\delta, \epsilon] = \delta \circ \epsilon - \epsilon \circ \delta$.

For this to be well-defined, we need that the Lie Bracket of two left-invariant derivations to again be a left-invariant derivation. However, this can be easily checked.

While the construction of the Lie Algebra in terms of derivations provides an easy to understand building block structure, it is difficult to directly understand the structure of the derivation space. Therefore, it is much more convenient to identify it with a tangent space as defined in the previous section:

Proposition 2.15. *The Lie Algebra $\mathfrak{g}$ of G is isomorphic as a vector space to the tangent space $Tan_e(G)$*

Proof. Within this proof, we will consider the tangent space in two ways: the local ring construction, and the related definition in terms of point derivations. To prove the theorem, we construct the map sending a tangent vector x to a derivation $*\mathrm{x}$ which we define in the following way:

$$(f * \mathrm{x})(x) = \mathrm{x}(\lambda_{x^{-1}} f)$$

We first check that it is a derivation

$$\begin{aligned}
(fg * \mathrm{x})(x) &= \mathrm{x}(\lambda_{x^{-1}}(fg)) \\
&= \mathrm{x}((\lambda_{x^{-1}} f)(\lambda_{x^{-1}} g)) \text{ but x is a point derivation} \\
&= \mathrm{x}(\lambda_{x^{-1}} f) g(x) + f(x)\mathrm{x}(\lambda_{x^{-1}} g) \\
&= ((f * \mathrm{x})g + f(g * \mathrm{x}))(x)
\end{aligned}$$

We then need to make sure it is left invariant

$$\begin{aligned}
\lambda_y(f * \mathrm{x})(x) &= (f * \mathrm{x})(y^{-1}x) \\
&= \mathrm{x}(\lambda_{x^{-1}y} f) \\
&= \mathrm{x}(\lambda_{x^{-1}}(\lambda_y f)) \\
&= (\lambda_y f) * \mathrm{x}(x)
\end{aligned}$$

Thus, we have a well defined map in one direction. We now need an inverse map. We again turn to our definition of the tangent space in terms of point derivations. Since as previously discussed, the derivations are determined by their effect on $\mathcal{O}(G)$, we can define our inverse map θ as

$$(\theta\delta)(f) = (\delta f)(e)$$

for δ a derivation and f a function in $\mathcal{O}(G)$. We can then check that the composites are the identity

$$\begin{aligned} f * \theta(\delta)(x) &= \theta(\delta)(\lambda_{x^{-1}} f) \\ &= (\delta\lambda_{x^{-1}} f)(e) \text{ but we know the action commutes with derivations so} \\ &= \lambda_{x^{-1}}(\delta f)(e) \\ &= (\delta f)(x) \end{aligned}$$

Then we just need to show the other composition also gives the identity:

$$\begin{aligned} \theta(*\mathrm{x})(f) &= f * \mathrm{x}(e) \\ &= \mathrm{x}(\lambda_{e^{-1}} f) \\ &= \mathrm{x}(f) \end{aligned}$$

□

We must also note that differentiation also operates on functions in the following way:

Proposition 2.16. *If $\phi : G \to G'$ is a morphism of algebraic groups, $d\phi_e : \mathfrak{g} \to \mathfrak{g}'$ is a homomorphism of Lie Algebras*

Proof. Let $\mathrm{x}' = d_\phi(\mathrm{x})$, $\mathrm{y}' = d_\phi(\mathrm{y})$, $f = \phi^* f'$. Now

$$\begin{aligned} [\mathrm{x}', \mathrm{y}'](f') &= (\mathrm{x}'\mathrm{y}' - \mathrm{y}'\mathrm{x}')(f') \\ &= (f' * \mathrm{y}' * \mathrm{x}')(e) - (f' * \mathrm{x}' * \mathrm{y}')(e) \\ &= \mathrm{x}'(f' * \mathrm{y}') - \mathrm{y}'(f' * \mathrm{x}') \\ &= \mathrm{x}(\phi^*(f' * \mathrm{y}')) - \mathrm{y}(\phi^*(f' * \mathrm{x}')) \end{aligned}$$

In a similar way we have that

$$d_\phi[x,y] = \mathrm{x}(f * \mathrm{y}) - \mathrm{y}(f * \mathrm{x})$$

and so all it remains to show is that $\phi^*(f' * \mathrm{x}') = (f * \mathrm{x})$. That is, that $(\phi^* f') * \mathrm{x} = \phi^*(f' * d_\phi \mathrm{x})$. Let's evaluate each side at $x \in G$.

$$\begin{aligned}
(\phi^* f') * \mathrm{x}(x) &= x(\lambda_{x^{-1}} \phi^* f') \\
\phi^*(f' * d_\phi \mathrm{x})(x) &= (f' * d_\phi \mathrm{x})\phi(x) \\
&= d_\phi \mathrm{x}(\lambda_{\phi(x)^{-1}} f') \\
&= \mathrm{x}(\phi^*(\lambda_{\phi(x)^{-1}} f'))
\end{aligned}$$

So we need to show that

$$\lambda_{x^{-1}} \phi^* f' = \phi^*(\lambda_{\phi(x)^{-1}} f') \qquad (\star)$$

We evaluate at some $y \in G$.

$$\begin{aligned}
(\lambda_{x^{-1}} \phi^* f')(y) &= \phi^* f'(xy) \\
&= f' \phi(xy)
\end{aligned}$$

We now evaluate the right hand side of $\star$

$$\begin{aligned}
\phi^*(\lambda_{\phi(x)^{-1}} f')(y) &= (\lambda_{\phi(x)^{-1}} f')\phi(y) \\
&= f'(\phi(x)\phi(y)) \\
&= f'(\phi(xy))
\end{aligned}$$

as required. □

Thus, we have shown that there is a simpler way to find the Lie Algebra of an algebraic group G by using the tangent space and so we can calculate the Lie Algebras of the groups we are fundamentally interested in: GL_n and SL_n. However, we still need to find a way to relate the subgroups of the algebraic groups to the ideals of the Lie Algebras. This relation is derived directly from an important representation that relates G and $\mathfrak{g}$, the Adjoint Representation Ad.

Definition 2.17. The *Inner Automorphism* $\mathrm{Int}\, g : G \to G$ is the group automorphism $\mathrm{Int}\, g(h) = ghg^{-1}$.

From the Inner Automorphism, we derive the Adjoint Representation.

Definition 2.18. The *Adjoint Representation* Ad $g : G \to GL(\mathfrak{g})$ is defined to be the differential of the inner autormorphism. That is, if we think of the Lie Algebra as the space of left derivations, and $x \in G$, Ad $x(\delta) = \rho_x \delta \rho_{x^{-1}}$

Remark 2.19. We see that the Adjoint Representation is indeed a representation, as it maps into the general linear group of the lie algebra

Before we prove some basic properties of the Adjoint representation, we need to know what the Lie Algebra of the general linear group is:

Proposition 2.20. *The Lie Algebra $\mathfrak{gl}_n$ of GL_n is the set of $n \times n$ matrices, with Lie Bracket $[x, y] = xy - yx$ defined by matrix multiplication and addition.*

Proof. Since the ring of regular functions of an affine open variety, embedded in $\mathbb{A}^{n^2}$, say, is just the ring of regular functions on $\mathbb{A}^{n^2}$, we see that the tangent space at e of GL_n has canonical basis $\delta/\delta T_{i,j}$, evaluated at e, and so we can write any tangent vector x as a set of numbers defined by what it does to each of the $T_{i,j}$. That is, x is defined by the $x_{i,j} = \mathrm{x}(T_{i,j})$ where we arrange the $x_{i,j}$ in a square matrix. One can check that the multiplication of these tangent vector matches the matrix multiplication and so we can identify the tangent vectors as a subset of $M_n(k)$. This map is injective, since x gets sent to zero if it kills all the $T_{i,j}$. That is, it is the zero matrix. It is surjective since the dimensions on either side are the same, and so we can indeed identify the Lie Algebra of GL_n with $M_n(k)$. □

Now we can prove some basic properties about the Adjoint Representation in relation to GL_n. These will be useful since, as previously discussed, any affine algebraic group is a closed subgroup of GL_n and will therefore inherit any properties we can prove about GL_n (we will discuss this more rigorously below). In particular, they will inherit the structure on the differential of Ad, which will be paramount in comparing structure between a group and its Lie Algebra. Denote the coordinate functions on GL_n by $T_{i,j}$ and let T be the matrix whose (i, j) entry is $T_{i,j}$.

Lemma 2.21. *For $x \in GL_n$ $x \in \mathfrak{gl}_n$, $\rho_x T_{ij}$ is the (i, j) entry of Tx. Further, $(T_{i,j} * x)$ is the i, j entry of Tx.*

Proof. We want to show that $\rho_x T_{i,j}(y) = (Tx)_{i,j}(y)$. From evaluating the group action we have that $\rho_x T_{i,j}(y) = T_{i,j}(xy)$, which we can think of as just the i, j entry of the matrix xy. This is something we know from matrix multiplication laws - it is $\sum_k y_{i,k}x_{k,j}$ but the i,k entry of y is the coordinate matrix $T_{i,k}$ evaluated on y, so we actually have $\sum_k T_{i,k}(y)x_{k,j}$, which if we roll back our matrix multiplication we have $(Tx)_{i,j}(y)$. In a similar vein we can prove the result for the convolution when applied to some arbitrary y:

$$\begin{aligned}(T_{i,j} * \mathrm{x})(y) &= \mathrm{x}(\lambda_{y^{-1}}T_{i,j}) \\ &= \mathrm{x}(\sum_k y_{i,k}T_{k,j}) \\ &= \mathrm{x}(\sum_k T_{i,k}(y)T_{k,j}) \\ &= (T\mathrm{x})_{i,j}(y)\end{aligned}$$

□

Lemma 2.22. *Let $x \in GL_n$, $\mathrm{x} \in \mathfrak{gl}_n$. Then $Ad\ x(\mathrm{x}) = x\mathrm{x}x^{-1}$.*

Proof. Since the coordinate functions act as a basis for $\mathfrak{gl}_n$, it suffices to show the result when applied to one of the $T_{i,j}$. Within this proof we will apply both sides of the previous lemma, as well as the same splitting through matrix multiplication step used in the previous proof.

$$\begin{aligned}\mathrm{Ad}\ x(\mathrm{x})(T_{i,j}) &= \rho_x(*\mathrm{x})\rho_{x^{-1}}(T_{i,j}) \\ &= \rho_x(*\mathrm{x})(Tx^{-1})_{i,j}) \\ &= \rho_x(*\mathrm{x})\sum_k T_{i,k}x^{-1}_{k,j} \\ &= \rho_x\sum_k (T\mathrm{x})_{i,k}x^{-1}_{k,j} \\ &= \rho_x\sum_k\sum_l T_{i,l}\mathrm{x}_{l,k}x^{-1}_{k,j} \\ &= \sum_k\sum_l\sum_m T_{i,m}x_{m,l}\mathrm{x}_{l,k}x^{-1}_{k,j} \\ &= (Tx\mathrm{x}x^{-1})_{i,j} \\ &= x\mathrm{x}x^{-1}(T_{i,j})\end{aligned}$$

□

We can then conclude that Ad : $GL_n \to GL_{n^2}$ is a morphism of algebraic groups and since restriction commutes with differentiation. Then, based on this, and our isomorphism between G and a closed subgroup of GL_n, we can show

Proposition 2.23. *$Ad : G \to GL(\mathfrak{g})$ is a morphism of algebraic groups. When G is a closed subgroup of GL_n, Ad x is conjugation by x for $x \in G$.*

Proof. Firstly, by the proposition that all affine algebraic groups are contained in GL_n we have that $G \subseteq GL_n$, and so we have that Int_G is the restriction of Int_{GL_n} and that $\mathfrak{g} \subseteq \mathfrak{gl}_n$. Then, since restriction commutes with differentiation, we have that Ad_G is the restriction of Ad_{GL_n}, and so by the above conclusion about Ad : $GL_n \to Gl_{n^2}$, taking the restriction gives us the morphism of algebraic groups we require. $\square$

The last piece of Lie Algebra theory we need before a technical proof about the Lie Algebra of SL_n is some way to relate structure between the group and its Lie Algebra. We will do this by looking at the differential of Ad. To do so we must first compute the differentials of some more simple maps:

We first look at the multiplication map $\mu : G \times G \to G$. It is not hard to show that the tangent space of $G \times G$ at (e, e): $\mathcal{T}(G \times G)_{(e,e)}$ is isomorphic to $\mathcal{T}(G)_e \oplus \mathcal{T}(G)_e$ (take the direct sum of the projection maps), and so we can therefore think about the differential of this multiplication map as $d\mu_{(e,e)}(\mathrm{x}, \mathrm{y})$

Proposition 2.24. *$d\mu_{(e,e)}(x, y) = x + y$*

Proof. For $f \in \mathcal{O}_G$, $\mu^*(f) = \sum f_i \otimes g_i$, and so we have $f(xy) = \sum f_i(x)g_i(y)$, and in particular $f = \sum f_i(e)g_i = \sum f_i g_i(e)$ Now

$$\begin{aligned} d\mu_{(e,e)}(\mathrm{x}, \mathrm{y})(f) &= (\mathrm{x}, \mathrm{y})(\mu^* f) \\ &= (\mathrm{x}, \mathrm{y})(\sum f_i \otimes g_i) \\ &= \sum \mathrm{x}(f_i)g_i(e) + \sum f_i(e)\mathrm{y}(g_i) \end{aligned}$$

where we get the last line since $(\mathrm{x}, \mathrm{y}) = (\mathrm{x}, e)+(e, \mathrm{y})$. Now, considering $(\mathrm{x}+\mathrm{y})(f)$:

$$\begin{aligned} (\mathrm{x} + \mathrm{y})(f) &= \mathrm{x}(f) + \mathrm{y}(f) \\ &= \mathrm{x}(\sum f_i g_i(e)) + \mathrm{y}(\sum g_i f_i(e)) \\ &= \sum \mathrm{x}(f_i)g_i(e) + \sum f_i(e)\mathrm{y}(g_i) \end{aligned}$$

$\square$

We now consider the differential of the inverse map: ι.

Proposition 2.25. $d\iota(x) = -x$

Proof. We know that $d(\text{id}) = \text{id}$ and that $d(\text{id}, \iota) = (d(\text{id}), d(\iota))$. Further, we also know that $0 = d(\mu \circ (\text{id}, \iota))(x)$ and so:

$$\begin{aligned} d(\mu \circ (\text{id}, \iota))(\text{x}) &= d(\mu d(\text{id}, \iota)(\text{x})) \\ &= d\mu(\text{x}, d(\iota)(\text{x})) \\ &= \text{x} + d\iota(\text{x}) \end{aligned}$$

so $d\iota\text{x} = -\text{x}$. □

To compute the differential of Ad we turn to our final definition for the tangent space relating to the Dual Numbers. Recall that:

Remark 2.26. The tangent space at a point x are in bijection with the $k[\epsilon]$-valued points that lift α_x

In the case of the tangent space at the identity to GL_n, we actually have an even more concrete way to think about the lifts. We know that the tangent space at the identity for GL_n are homorphisms $\phi : R \to k[\epsilon]$ that lift the map sending x to the identity. That is, the set of $k[\epsilon]$ points whereby they are sent to the identity matrix when ϵ is sent to 0. Thus, $T_e(GL_n)$ is the set of matrices in GL_n (b_{ij}) where $b_{ii} = 1 + a_{ii}\epsilon$ and $b_{ij} = a_{ij}\epsilon$. That is, we can think of the matrix (b_{ij}) as $I + (a_{ij})\epsilon$.

Finally, we can prove that:

Theorem 2.27. *The differential of Ad is ad where ad* $x(y) = [x, y]$

Proof. As in the case of the computation of Ad, 2.23, we will restrict to $\mathfrak{gl}_n$ and GL_n. Ad maps GL_n to GL_{n^2}, so the differential of Ad, ad : $T_e(GL_n) \to T_e(GL_{n^2})$. That is, we send $x = \text{id} + M\epsilon$ in the tangent space to the map sending $y \to xyx^{-1}$. We know from the addition law on the tangent space, along with the formula for the differential of the inverse map that this is the map that sends

$$\begin{aligned} y &\to (\text{id} + M\epsilon)y(\text{id} - M\epsilon) \\ &= y + My\epsilon - ym\epsilon \\ &= y + (My - yM)\epsilon \end{aligned}$$

But this is the map $\mathrm{id} + [M, -]\epsilon$ applied to y. Therefore, $d\mathrm{Ad} = \mathrm{ad}$ sends the tangent vector $\mathrm{id}+M\epsilon$ to the map $\mathrm{id}+[M, -]\epsilon$. That is, $\mathrm{ad}\ x$ is the lie bracket. □

We will then use this to show some greater structure on $\mathfrak{h}$ corresponding to H in the case where H is normal. First we show what H corresponds to as a closed subgroup:

Lemma 2.28. *For H a closed subgroup of G, I the ideal of $\mathcal{O}(G)$ vanishing on H, the corresponding lie algebra $\mathfrak{h}$ is the subalgebra $\mathfrak{h} = \{x \in \mathfrak{g} \mid I * x \subset I\}$*

Proof. Suppose $\mathrm{x} \in \mathfrak{h}$, $f \in I$, $x \in H$. Then $(f * \mathrm{x})(x) = \mathrm{x}(\lambda_{x^{-1}} f)$, but $\lambda_{x^{-1}} f$ still vanishes on H since $(xy) \in H$ if $y \in H$ so $\mathrm{x}(\lambda_{-1} f) = 0$, and so $I * \mathrm{x} \subset I$.

Now suppose $\mathrm{x} \in \mathfrak{g}$ such that $I * \mathrm{x} \subset I$. That is, we have that for all $f \in I$,

$$\begin{aligned}(f * \mathrm{x})(e) &= \mathrm{x}(\lambda_{e^{-1}} f)\\ &= \mathrm{x}(f)\end{aligned}$$

That is, $\mathrm{x}f \in I$, so $\mathrm{x}(f)$ vanishes on H for all $f \in I$, so $\mathrm{x} \in \mathfrak{h}$ □

Then, when H is normal, $\mathfrak{h}$ becomes an ideal:

Lemma 2.29. *For H a closed normal subgroup of G, $\mathfrak{h}$ is an ideal of $\mathfrak{g}$. That is, for $x \in \mathfrak{g}$, $y \in \mathfrak{h}$, $[x, y] \in \mathfrak{h}$.*

Proof. If H is normal, it is invariant under Int, and so its corresponding lie subalgebra $\mathfrak{h}$ is invariant under Ad. In particular, if we have a basis for $\mathfrak{g}$ extended from a basis for $\mathfrak{h}$, we see that Ad will have the matrix form $\begin{bmatrix} * & * \\ 0 & * \end{bmatrix}$, where the first block is the first m columns that make up the basis of $\mathfrak{h}$. Differentiating will not change this form, and so ad also fixes $\mathfrak{h}$, which precisely gives us that $\mathfrak{h}$ is an ideal, since $\mathrm{ad}\ x(\mathfrak{h}) = [x, \mathfrak{h}] = \mathfrak{h}$. □

Thus, we have a correspondence between normal subgroups H of G and ideals $\mathfrak{h}$ in $\mathfrak{g}$. In particular, if we have no nontrivial ideals $\mathfrak{h}$ in $\mathfrak{g}$; that is, $\mathfrak{g}$ is *simple*, G has no nontrivial normal subgroups. It is this fact that we will use to prove that SL_n is linearly reductive. That is, we will prove the simplicity of the Lie Algebra of SL_n, the trace zero matrices:

Lemma 2.30. *The Lie Algebra $\mathfrak{sl}_n$ of SL_n are the $n \times n$ matrices of trace zero.*

Proof. We prove this using our first definition of the tangent space, and note that we identify the lie algebra with the vector space with the zero matrix in $\mathbb{A}^{n^2}$ as the zero vector. That is, following the computation of the tangent space, we apply the shift map $t_{i,j} \to t_{i,j} - e_{i,j} = t'_{i,j}$, and so we see that, since SL_n is defined by the function $f(x) = \det(x) - 1$, the tangent space is defined on the $t'_{i,j}$ with vanishing polynomial

$$\sum \frac{\delta f}{\delta t_{i,j}}(e)t'_{i,j}$$

Thus, all that remains is to compute the partial derivatives of the determinant at e. We can write the determinant using the cofactor expansion:

$$\det(T) = t_{1,1}\det(T_{1,1}) \pm t_{1,2}\det(T_{1,2}) \pm \cdots \pm t_{1,n}\det(T_{1,n})$$

where $T_{i,j}$ are the minors with the ith row and jth column of M omitted. We see that differentiating at $t_{1,1}$ gives determinant of $T_{1,1}$ which is the identitity matrix of size $n-1 \times n-1$, and so is 1. For the remainder of the $t_{1,i}$, since in the minors the first row of T is omitted but the first column is not, the first column is all zeroes, and as such the determinant of the minors are zero, and thus the partial derivative is also. For the remainder of the $t_{i,j}$ their partial differentials will have elements in each component of the sum except for the component $t_{1,j}\det(T_{1,j})$, and therefore will be written as some poly in the other $t_{i,j}$:

$$t_{1,1}g_{1,1} \pm \cdots \pm t_{1,j-1}g_{1,j-1}t_{1,j+1}g_{1,j+1} \pm \cdots \pm t_{1,n}g_{1,n}$$

However, at e all the $t_{1,j}$ where $j \neq 1$ are zero, so the partial derivative is fully contained in the summand $t_{1,1}\det(T_{1,1})$ and $t_{1,1} = 1$, so we may as well consider the partial derivative in the minor. However, the fact we're considering the minor which is the identity matrix one dimension lower allows us to apply the above reasoning inductively, and we therefore find that $\frac{\delta f}{\delta t_{i,j}} = 1$ if $i = j$ and 0 otherwise. Thus, we find that the tangent space is defined by the polynomial

$$\sum_{i=1}^{n} t'_{i,i} = 0$$

That is, the trace zero matrices. □

To prove that this algebra is simple, we will utilise a proof idea explored by Yung [10], and will first need to prove a short technical lemma about eigenspaces:

Lemma 2.31. *Suppose V is a finite dimensional vector space and $T : V \to V$ is a diagonalisable linear map. If Λ is the set of eigenvalues of T and V_λ is the* *eigenspace of T associated with an eigenvalue λ, and W is a T-invariant subspace, then*

$$W = \bigoplus_{\lambda\in\Lambda}(W \cap V_\lambda)$$

Proof. Since T is diagonalisable, we have an eigenbasis for V, and therefore we can write $V = \bigoplus_{\lambda\in\Lambda} V_\lambda$. Thus, for any vector, and in particular vectors $w \in W$, we can write

$$w = \sum_{w_\lambda \in V_\lambda} w_\lambda$$

In particular, if we look at the λ_i such that $w_\lambda \neq 0$, we write

$$w = w_{\lambda_1} + \cdots + w_{\lambda_m}$$

. That is, we get that

$$\begin{aligned} w &= w_{\lambda_1} + \cdots + w_{\lambda_m} \\ Tw &= \lambda_1 w_{\lambda_1} + \cdots + \lambda_m w_{\lambda_m} \\ &\vdots \\ T^{m-1}w &= \lambda_1^{m-1} w_{\lambda_1} + \cdots + \lambda_m^{m-1} w_{\lambda_m} \end{aligned}$$

Taking $x = (w_{\lambda_1}, \cdots, w_{\lambda_m})$, $b = (w, Tw, \cdots, T^{m-1}w)$, we can write a coefficient matrix

$$A = \begin{bmatrix} 1 & \cdots & 1 \\ \lambda_1 & \cdots & \lambda_m \\ \vdots & \ddots & \vdots \\ \lambda_1^{m-1} & \cdots & \lambda_m^{m-1} \end{bmatrix}$$

and so if A is invertible, we can then write the w_{λ_i} as a linear combination of the $T^j w$, but the $T^j w$ are in W since W is T-invariant, and so we would be done. Thus, all it remains to show is that the determinant of A is nonzero. However, A is a square vandermonde matrix, so we have that

$$\det(A) = \prod_{i<j} \lambda_j - \lambda_i$$

But, since T is diagonalisable, the eigevalues are distinct and so the determinant is nonzero. $\square$

Finally, we can prove the simplicity of $\mathfrak{sl}_n$. That is, that $\mathfrak{sl}_n$ has no nontrivial ideals:

Theorem 2.32. $\mathfrak{sl}_n$ *is simple.*

Proof. Consider an ideal $J \subseteq \mathfrak{sl}_n$. Since $\mathfrak{gl}_n = \mathfrak{sl}_n \oplus (k \cdot I)$, where I is the identity and $k \cdot I$ is contained in the center of $\mathfrak{gl}_n$, any ideal of $\mathfrak{sl}_n$ is also an ideal of $\mathfrak{gl}_n$.

Now, consider the matrix $s = \sum_{k=1}^{n} 2^k E_{kk}$ where the E_{ij} are the standard basis for $n \times n$ matrices. We see that for any of the E_{ij}:

$$\text{ad } s(E_{ij}) = [s, E_{ij}] = (2^i - 2^j)E_{ij}$$

and hence ad s is a diagonal matrix in $GL(\mathfrak{gl}_n)$, and therefore (obviously) is diagonalizable. Now, since J is an ideal of $\mathfrak{gl}_n$, it must be ad$s$ invariant, and so we can apply the previous technical lemma. That is, $J = \bigoplus_{\lambda \in \Lambda} V_\lambda \cap J$. In this instance, since the eigenvalues are 0 and $2^i - 2^j$ $i \neq j$, this means that

$$J = J \cap (V_0 \bigoplus_{i \neq j} V_{2^i - 2^j})$$

Since V_0 is the diagonal matrices, we see that if $J \cap V_0 \neq \{0\}$, then J contains a diagonal matrix of trace zero. In particular that means there exists a matrix $a = \sum_{i=1}^{n} a_i E_{ii}$ where $a_i \neq a_j$ for some i and j. But this means that $[a, E_{ij}] \in J$, but $b = [a, E_{ij}] = (a^i - a^j)E_{ij}$, and so for $a' = \frac{2}{a^i - a^j} E_{ii} + \frac{1}{a^i - a^j} E_{jj}$ $[a', b] = E_{ij}$, and so $E_{ij} \in J$.

Similarly, if $J \cap V_0 = \{0\}$, then there exists some $i \neq j$ such that $J \cap V_{2^i - 2^j}$ is nonempty,but the eigenspace $V_{2^i - 2^j}$ is the span of some E_{ij}, and so $E_{ij} \in J$.

Now, since

$$[E_{jk}, E_{ij}] = -E_{ik}$$

for $k \neq i$ and

$$[E_{ki}, E_{ij}] = E_{kj}$$

for $k \neq j$, if $l \neq i$, then we have $E_{il} \in J$, and so by the second bracket for $k \neq l$, we have $E_{kl} \in J$. Similarly, if $k \neq j$, then we have that $E_{kj} \in J$ and so by the second bracket for $k \neq l$ we have $E_{kl} \in J$. This covers all basis matrices where $i \neq j$ except for E_{ji}. We can get E_{ji} by getting some E_{ik} $k \neq i, j$ in the following way

$$[E_{jk}, E_{ij}] = -E_{ik}$$
$$[E_{ji}, E_{ik}] = E_{jk}$$
$$[E_{ki}, E_{jk}] = -E_{ji}$$

where the last step is allowed since $i \neq j$. Thus, we have all E_{kl} for $k \neq l$. Thus, we also have

$$[E_{kl}, E_{lk}] = E_{kk} - E_{ll}$$

for all $k \neq l$, but this means we have all $\mathfrak{sl}_n$, and so $I = \mathfrak{sl}_n$, as required. □

2.1.4 Reductivity

To understand reductivity, we first need to understand unipotency and semisimplicity. We begin by defining unipotency and semisimplicity for elements of GL_n.

Definition 2.33. An element $g \in GL_n$ is *unipotent* if all of its eigenvalues are 1. An element $g \in GL_n$ is *semisimple* if it is diagonalisable.

Remark 2.34. Equivalently, an element g in GL_n is unipotent if $g - I$ is nilpotent.

With respect to this definition, we can define unipotency and semisimplicity for a general element $g \in G$.

Definition 2.35. An element $g \in G$ is *unipotent* if there is a faithful linear representation $\rho : G \to GL_n$ such that $\rho(g)$ is unipotent. Similarly, an element $g \in G$ is *semisimple* if $\rho(g)$ is.

Example 2.36. The matrices whose diagonals are all 1, and are upper or lower triangular are unipotent, since $M - I$ has zeroes along the diagonal,is triangular and therefore nilpotent.

From here, we can define the unipotent radical, and reductivity:

Definition 2.37. The *unipotent radical* $R_u(G)$ is the unique closed maximal connected normal subgroup consisting of unipotent elements. A group G is *reductive* if this radical is trivial.

From here, we will define linear reductivity, and quote an important theorem due to many mathematicians that will finally allow us to demonstrate that our main group, SL_n, is linearly reductive

Definition 2.38. An algebraic group G is said to be *linearly reductive* if every finite dimensional linear representation $\rho : G \to GL(V)$ decomposes as a sum of irreducibles.

Example 2.39. A basic example is $\mathbb{G}_m^k$. We see that from [1], that for diagonalizable linear algebraic groups D, which Tori are, every representation of D splits into a finite direct sum of one-dimensional representations. In particular, for $\mathbb{G}_m^k$, these one dimensional representations $\phi : \mathbb{G}_m^k \to GL_1 = \mathbb{G}_m$ are monomial. Let $(t_1, \cdots, t_k) \in \mathbb{G}_m^k$, and let $\phi : \mathbb{G}_m^k \to \mathbb{G}_m$. Then:

$$\phi(t_1, \cdots, t_k) = t_1^{a_1} \cdots t_k^{a_k}$$

Example 2.40. A non-example of a linearly reductive group is the complex numbers under addition. To see this, consider the representation $a \to \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}$. This representation is reducible but indecomposable. It is reducible since it has an invariant subspace in elements $(b, 0)$. However, this is the only invariant subspace, so the representation is indecomposable.

Whilst we're discussing reductivity, it will be helpful later to understand a related concept: geometric reductivity.

Definition 2.41. A group G is *geometrically reductive* for every finite dimensional linear representation $\rho : G \to GL(V)$ and every non-zero G-invariant point $v \in V$, there is a G-invariant non-constant homogeneous polynomial $F \in \mathcal{O}(V)$ such that $f(v) \neq 0$.

Examples 2.42. As we are about to see, every linearly reductive group is geometrically reductive. A further example is SL_n for a field of any characteristic. Thus we see that not every geometrically reductive group is linearly reductive.

We now state one of the most important theorems surrounding reductivity. It is due to many people - Nagata, Mumford, Weyl and Haboush

Theorem 2.43. *Every linearly reductive group is geometrically reductive. Every reductive group is geometrically reductive and vice versa. In characteristic zero fields, all three types of reductivity are equivalent.*

We will prove below that every linearly reductive group is geometrically reductive, and so will provide some indication as to why some of these statements might be true - but otherwise would require far more work and background than would be appropriate. It is however, useful for our purposes since it allows us to prove that SL_n is linearly reductive in characteristic zero.

Theorem 2.44. *SL_n is linearly reductive in characteristic zero fields*

Proof. We have seen already (2.32) that in characteristic zero, the lie algebra of SL_n, $\mathfrak{sl}_n$ is simple. Further, we know that since normal subgroups correspond to ideals in the lie algebra, that since we have no ideals in the lie algebra, we have no normal subgroups. In particular therefore, we have no normal solvable subgroups, and so SL_n is reductive and therefore linearly reductive. □

Remark 2.45. Since the representation theory of SL_n is fully understood, one could prove that SL_n is linearly reductive explicitly using its representation theory, but this is an arduous process deserving of its own thesis.

For the remainder of this section as well as the next, we will follow the work of Hoskins in [6] as well as a few other contributers I will mention where relevant. The proofs are based heavily upon her work although the understanding is my own. Mukai [8] was my first exposure to some of the theory in this section, and although his work is not followed here in the same way Hoskins' is, he still influenced my understanding on the subject.

Proposition 2.46. *Every linearly reductive group is geometrically reductive.*

Proof. We will prove the implication with a chain of implications pertaining to the following facts. That is, for the following:

1. G is linearly reductive
2. For any finite dimensional linear representation $\rho : G \to GL(V)$, any G-invariant subspace V' admits a G-stable complement.
3. For any surjection of finite dimensional G-representations $\phi : V \to W$, the induced map on G-invariants is surjective
4. For any finite dimensional linear representation $\rho : G \to GL(V)$ and every nonzero G-invariant point v, there is a G-invariant linear form $f : V \to k$ such that $f(v) \neq 0$

We will prove that $1 \implies 2$, $2 \implies 3$ and $3 \implies 4$. Furthermore, note that a group that has property 4 is geometrically reductive.

For $1 \implies 2$, since G is linearly reductive, and V' is G-invariant, both V' and V decompose as a direct sum of irreducibles. However, since $V' \hookrightarrow V$, $\bigoplus_{\text{irred}} V' \hookrightarrow \bigoplus_{\text{irred}} V$ and so letting $V'' = V/V'$ provides a G-stable complement.

For $2 \implies 3$, let $V' = \ker(f)$. Then $V'' \cong W$ and since both are G-invariant $V^G = V'^G \oplus V''^G$. That is, we have a surjective map from V^G to $V''^G \cong W^G$.

For 3 $\implies$ 4, pick a nonzero G-invariant vector v. This determines a G-invariant linear form $\phi : V^\vee \to k$. If we let G act trivially on k, ϕ is a surjection of G-representations and by 3 then, we have a surjection $(V^\vee)^G \to k^G = k$. Thus, taking the preimage of 1, there will be an f in the preimage such that $f(v) = 1$. □

2.2 The Categorical Quotient

Consider an algebraic variety X. Suppose further that there is a group G that acts on X. That is, there exists $\phi : G \times X \to X$, where $\phi(g, x) = g \cdot x$. A reasonable structure one might want to consider is some kind of quotient of X by G. What might initially spring to mind is the set of orbits. That is, given the equivalence relation $x \sim y$ if $y = g \cdot x$ for some $g \in G$, identify points that are equal. However, the set of orbits may not actually be a variety. Recall the example given in the introduction: the action of k on $\mathbb{A}^2$. Since 0 is contained in the closure of any other point, we see that the set of orbits cannot be a variety. Therefore, an alternative definition for the quotient of X by G needs to be found. We will call such a quotient a categorical quotient. We can then ascribe further definitions to understand how well behaved such a quotient is. A "good quotient" has several nice properties that make it desirable whilst a "geometric quotient" tells us that, in fact, the categorical quotient has lined up with our orbit space.

We begin by defining the categorical quotient. In particular, we have the Yoneda Lemma for locally small categories, which we recall:

Theorem 2.47. *For F a functor from $\mathcal{C}$ to **Set**, natural transformations from the hom-functor of $A \in \mathcal{C}$ to F are in a one-to-one correspondence with the elements of $F(A)$.*

Corollary 2.48. *Suppose for some functor F, $F = Mor(Z, -)$. Then Z is unique up to isomorphism.*

Proof. Suppose $\mathrm{Mor}(A, -)$ and $\mathrm{Mor}(B, -)$ are isomorphic. Then the Yoneda Lemma and the natural transformation from $\mathrm{Mor}(A, -)$ to $\mathrm{Mor}(B, -)$ gives us an element of $\mathrm{Mor}(B, A)$ whilst the inverse transformation gives us an element of $\mathrm{Mor}(A, B)$. Then, the fact that the natural transformations are indeed inverses provides that the composition of the two resultant maps are also the identity. That is, $A \cong B$.

$\square$

The Yoneda Lemma is important in our case for our definition of our categorical formulation of the quotient. Recall that we defined ϕ to be the action of the group G on a variety X. Now, we define the functor $\mathrm{Fun}_{X/G}$. That is, for an object y, and for $\pi : G \times X \to X$ projection onto the second factor,

$$\mathrm{Fun}_{X/G}(Y) = \{\psi : X \to Y | \psi \circ \phi = \psi \circ \pi\}.$$

From this, we can see why the Yoneda Lemma matters. We can define the categorical quotient X/G indirectly here as the algebraic variety Z (if it exists) such that $\mathrm{Fun}_{X/G} = \mathrm{Mor}(Z, -)$ and importantly, from the corollary, such a Z is unique up to isomorphism.

Thus, we see that the categorical quotient is the Z such that $\mathrm{Fun}_{X/G} = \mathrm{Mor}(Z, -)$. We can write this more intuitively, as in the following definition:

Definition 2.49. A *categorical quotient* for the action of a group G on an algebraic variety X is a G-invariant regular map $\phi : X \to Y$ such that for every other G-invariant regular map $f : X \to Z$ there is a unique map $h : Y \to Z$ such that $f = h \circ \phi$.

2.3 The Affine GIT Quotient

Now that we have this definition, it would be reasonable to ask when such a Z exists. Indeed, there is a set of conditions encapsulated in the following theorem:

Theorem 2.50. *Let G be linearly reductive and X affine. If $\mathcal{O}(X) = A$, (that is $X = spec(A)$) then A^G (the functions of A fixed by G) is finitely generated and reduced. Furthermore, for $Z = spec(A^G)$, Z is the categorical quotient of X by G. In otherwords, $spec(A^G)$ is the Z for which $Fun_{X/G} = Mor(Z, -)$*

Note that there are two parts to this theorem. The first is that A^G is finitely generated (thereby proving that Z is a variety), and the second is that $\mathrm{spec}(A^G)$ is the categorical quotient. We will begin by proving that A^G is finitely generated. To do so, we introduce the concept of a Reynolds operator, and prove some simple lemmas:

Definition 2.51. For a group G acting on a k-algebra A, a linear map $R_A : A \to A^G$ is called a *Reynolds Operator* if it is a projection onto A^G and, for $a \in A^G$ and $b \in A$, we have $R_A(ab) = aR_A(b)$.

Example 2.52. A simple example is to take $\mathbb{Q}$ (under multiplication) acting on $\mathbb{C}$ in the following way: $g \cdot (a + bi) = a + gbi$. Then the invariant ring is all real numbers A, and we have a reynolds operator taking $a + bi$ to a. To see this note that $R(c(a + bi)) = R(ca + cbi) = ca = cR(a + bi)$.

To tie this into the present discussion, and to prove the lemma, we define what a *rational action* is:

Definition 2.53. A *rational action* of G on X is an action such that every element of X is contained in a finite dimensional G-invariant linear subspace of X.

Now we state and prove a lemma and corollary that will prove useful:

Lemma 2.54. *Let G be a linearly reductive group acting rationally on a finitely generated k-algebra A. Then, there exists a Reynolds operator $R_A : A \to A^G$.*

Proof. A is finitely generated. As such, it has a countable basis b_i, $i \in I$. Consider any element that is a j-fold product ($1 \leq j \leq n$) of the first n basis elements. Since the action of G is rational, it is contained in a finite dimensional G-invariant subspace. Further, since the action of a particular $g \in G$ must be a k-algebra homomorphism, we have that the direct sum of any such subspaces are G-invariant, as $g(a_1 + a_2) = ga_1 + ga_2$. Therefore, let B_n be the direct sum of all such invariant subspaces for the up to j-fold products of the first n basis elements. We note that $B_n \subset B_m$ if $m > n$, and that $A = \bigcup_{n \geq 0} B_n$. We know from our proof that linearly reductive implies geometrically reductive that for linearly reductive groups, any G-invariant subspace V' admits a G-stable complement. Thus, we can write $B_n = B_n' \oplus B_n^G$ where B_n' is a direct sum of irreducible representations of G. Thus, we have a surjection $R_n : B_n \to B_n^G$. To make this into a potential candidate for a Reynolds operator from $A \to A^G$, we need to show the maps for each A_n are compatible. Note that for $m > n$ we have

$$\begin{tikzcd} B_n \arrow[r, "R_n"] \arrow[d, hook] & B_n^G \arrow[d, hook] \\ B_m \arrow[r, "R_m"] & B_m^G \end{tikzcd}$$

as, since $B_n \subset B_m$, $B'_n \subset B'_m$ and $B_n^G \subset B_m^G$. Thus our projections are compatible. It remains to show that $R_A(ab) = aR_a(b)$ for $a \in A^G$, $b \in A$. Take an n such that a and b are contained in B_n, and let $m \geq n$ be such that $a(B_n) \subset B_m$. We can write B_n as $B_n^G + B'_n$ where B'_n is a direct sum of non-trivial irreducible representations - say $W_1, \cdots, W_k$. Consider l_a, left multiplication by a. By Schur's Lemma, $l_a(W_i)$ must either be trivial or an isomorphism, and therefore $l_a(B'_n) \subset B'_m$. Thus, writing $b = b^G + b'$, we have that $l_a(b) = l_a(b^G) + l_a(b')$. Furthermore, $a \in A^G$, so $ab^G \in A^G$. Thus, we have that $ab = l_a(b)$ can be written as a sum of a G-fixed component and otherwise in B_m. Thus, $R_A(l_a(b)) = R_A(ab^G + ab') = ab^G = aR_a(b)$ as required. □

Corollary 2.55. *Let A and B be k-algebras with a rational action of a linearly reductive group G which have Reynolds operators R_A and R_B. Then any G-equivariant homomorphism $h : A \to B$ of these k-algebras commutes with the Reynolds operators. That is, $R_B \circ h = h \circ R_A$.*

Finally, we require a lemma about noetherianness:

Lemma 2.56. *Let A be a k-algebra with a rational G-action and suppose that A has a Reynolds operator $R_A : A \to A^G$. Then, for any ideal $I \subset A^G$, we have $IA \cap A^G = I$. More generally, if $\{I_j\}_{j\in J}$ are a set of ideals in A^G, then we have*

$$(\sum_{j\in J} I_j A) \cap A^G = \sum_{j\in J} I_j$$

Furthermore, if A is Noetherian, then so is A^G.

Proof. Since $I \subset A^G$, we have that $I \subset IA \cap A^G$. Now, consider $x \in IA \cap A^G$, which we can write as the sum of some $i_j a_j$. Then, since the Reynolds operator fixes elements of A^G and is linear:

$$x = R_A(x) = R_A(\sum i_j a_j) = \sum R_A(i_j a_j) = \sum i_j R_A(a_j) \in I$$

Now suppose A is Noetherian. Then for any ascending chain of ideals $I_1 \subset I_2 \subset \cdots$, in A^G $I_1 A \subset I_2 A \subset \cdots$ is also an ascending chain of ideals in A and must terminate with $I_n A$. However, $I_n A \cap A^G = I_n$ and so the chain in A^G must also terminate. □

Now we can prove a theorem fundamental to the proof of theorem 2.50:

Theorem 2.57. *[Hilbert, Mumford] For A a finitely generated k-algebra, G a linearly reductive group acting on A, A^G is a finitely generated algebra.*

Proof. Let $a_1, \cdots, a_n$ be the generators of A. Then, since the action of G on A is rational, we have that each a_i is contained in a finite-dimensional G-invariant subspace V_i, with $b_{i,j}$ a basis (the j ranges and i stays fixed). Thus, the subspace $V = \bigoplus_{i=1}^n V_i$ is a finite-dimensional subspace containing the a_i with the $b_{i,j}$ as the basis (now i and j both range). Since the action of G must be a k-algebra homomorphism, we have the action is linear. In particular, $g(v_1 + v_2) = g(v_1) + g(v_2)$, and $g(\lambda v) = \lambda g(v)$, and so we can show V is also G-invariant, since $b_{i,j} = \sum_{i=1}^n \lambda_i a_i$, and so $g(b_{i,j}) = \sum_{i=1}^n \lambda_i g(a_i)$, and the $g(a_i)$ are in V_i. Furthermore, we know that $g(b_{i,j}) = \sum_{k,l} \lambda_{k,l} g(b_{k,l})$, and so the induced action of G on $\mathrm{Sym}^*(V)$ preserves polynomial degree, and in particular, the surjection from $\mathrm{Sym}^*(V) \to A$ is G-equivariant. Thus, by Lemma 2.54 and its corollary, we have a surjection from $\mathrm{Sym}^*(V)^G \to A^G$, and so we can consider the case of $\mathrm{Sym}^*(V)$ instead of A, as finite generation remains true under a quotient.

From here, we assume A is a polynomial algebra with the action G preserving homogeneous degree. Since k is a field, we have that A is Noetherian, and then by the most recent lemma, A^G is Noetherian. Further, since A is graded by homogeneous polynomial degree, and the action of G preserves this degree, we have a natural grading on A^G also by homogeneous polynomial degree. Thus, we have an ideal containing all the polynomials with no degree 0 term: A^G_+. But A^G is Noetherian, so A^G_+ is finitely generated. Therefore, A^G must be. $\square$

Now, we can move on to proving that $\mathrm{spec}(A^G)$ is the categorical quotient. To prove that our $\mathrm{spec}(A^G)$ variety is such a quotient, we will instead prove that it is something slightly stronger: a good quotient.

Definition 2.58. With G and X above, a regular map ϕ is a *good quotient* if the following conditions hold:

1. ϕ is G-invariant
2. ϕ is surjective
3. The image of every G-invariant closed subset $W \subset X$ is closed in Y.
4. For any two disjoint G-invariant closed subsets, their images are disjoint
5. For every affine open subset of Y, its preimage under ϕ is affine

6. For an open subset U of Y, the morphism between the ring of regular functions of U over Y and the ring of regular functions of the preimage of U over X is an isomorphism onto the G-invariant functions.

We can also define exactly what we mean when we say geometric quotient

Definition 2.59. A *geometric quotient* is a quotient whereby the preimage of ecah point is a single orbit. Equivalently, it is a good quotient whereby the action of G on X is closed in the sense that the orbits are closed. If the orbits are closed but the preimage gives two separate orbits, then this contradicts property 4 of the good quotient.

Remark 2.60. With this definition of a geometric quotient in hand, it is not hard to see that in the case of the GIT quotient as defined below that if a GIT quotient is geometric, it is the same as the orbit space.

Lemma 2.61. *If 2 holds, 3 and 4 are equivalent to the statement: "If W_1 and W_2 are disjoint, closed, G-invariant subsets, then the closures of $\phi(W_1)$ and $\phi(W_2)$ are."*

Proof. That 2, 3 and 4 imply the statement is clear. Further, if the closures of $\phi(W_1)$ and $\phi(W_2)$ are disjoint, then so are $\phi(W_1)$ and $\phi(W_2)$. Thus, for the statement implies 3 and 4 if 2 holds direction, it remains to show 3. Without loss of generality, we prove that $\phi(W_1)$ is closed. Take a point p in the closure of $\phi(W_1)$. Since ϕ is surjective, this has a preimage W. Now, since the closures of $\phi(W)$ and $\phi(W_1)$ are not disjoint, W and W_1 are not disjoint. Thus, $p \in \phi(W_1)$ and thus $\phi(W_1)$ is closed. □

Lemma 2.62. *Every good quotient is a categorial quotient. Furthermore, a map satisfying 1, 3, 4, and 6 of the properties above is a categorical quotient.*

Proof. Suppose a map $\phi : X \to Y$ satisfies at least properties 1, 3, 4 and 6. Then, since property 1 gives G-invariance, it remains to show the "universal" or unique factoring property. Consider a map $f : X \to Z$, we will construct the map h as follows:

Take a finite affine open cover V_i of Z and use it to get an open cover U_i of Y, with regular maps $h_i : U_i \to V_i$ in the following way:

Let $W_i = X - f^{-1}(V_i)$. Note this is closed since $f^{-1}(V_i)$ is open and G-invariant, since otherwise f would not be. Thus, by property 3, $\phi(W_i)$ is closed in Y. Now, let $U_i = Y - \phi(W_i)$. To show that this is an open cover, we must prove that $\bigcap_i \phi(W_i) = \emptyset$. Thus, in order to gain a contradiction, suppose othwerise. Then, there exists a point $p \in \bigcap_i \phi(W_i)$. Now, let P denote $\phi^{-1}(p)$, and take a closed orbit W in P. Then, since $\phi(W) \cap \phi(W_i) \neq \emptyset$ by the contrapostive of property 4, we get that $W \cap W_i \neq \emptyset$. But each W_i is G-invariant, which means that $W \subseteq W_i$. This is a contradiction since the V_i are a cover of Z and so the intersection of the W_i must be empty. Thus, we have our open cover. Now, by property 6 and the fact that f is G-invariant we get the following diagram:

$$\begin{tikzcd} \mathcal{O}_Z(V_i) \arrow[r, "h_i^*"] \arrow[d] & \mathcal{O}_Y(U_i) \arrow[d, "\sim"] \\ \mathcal{O}_X(f^{-1}(V_i))^G \arrow[r] & \mathcal{O}_X(\phi^{-1}(U_i))^G \end{tikzcd}$$

and h_i^* is the unique map that makes the diagram commute. Then, since each of the U_i and V_i are affine, the standard categorical equivalence provides us with a map $h_i : U_i \to V_i$, which, by construction, factors f restricted to $\phi^{-1}(U_i)$, and by inputting $U_i \cap U_j$ and similar data into the diagram, we see that h_i and h_j agree on the intersection and thus glue correctly. Therefore, we have the map $h : Y \to Z$ such that $f = h \circ \phi$ as required. $\square$

Now, to prove our theorem, we prove that the map $\phi : X \to X//G$ induced from the inclusion of $\mathcal{O}(X)^G \hookrightarrow \mathcal{O}(X)$ is a good quotient and to do so we require two last lemmas:

Lemma 2.63. *Let G be an affine algebraic group acting on an affine variety X. Then any $f \in \mathcal{O}(X)$ is contained in a finite dimensional G-invariant subspace of $\mathcal{O}(X)$.*

Proof. First, (from Hoskins) we note that the action of G on X gives us a homomorphism of k-algebras $\sigma^* : \mathcal{O}(X) \to \mathcal{O}(G \times X) \cong \mathcal{O}(G) \otimes_k \mathcal{O}(X)$ given by $f \mapsto \sum h_i \otimes f_i$. This, in turn provides a homomorphism from G to the automorphisms on $\mathcal{O}(X)$ where the k-algebra automorphism of $O(X)$ corresponding to each g is given by $f \mapsto \sum h_i(g) f_i$. We can use this automorphism in particular to note that if we take the vector space spanned by the f_i given above, not only does it contain f, but it is also G-invariant. $\square$

Lemma 2.64. *Let G be a geometrically reductive group acting on an affine variety X. if W_1 and W_2 are disjoint G-invariant closed subsets of X, then there is an invariant function $f \in \mathcal{O}(X)^G$ which sepeartes these sets. That is, $f(W_1) = 0$, $f(W_2) = 1$.*

Proof. First note that $(1) = I(\emptyset) = I(W_1) \cap I(W_2) = I(W_1) + I(W_2)$. Thus, we can write $1 = f_1 + f_2$ such that $f_1(W_1) = 0$ and $f_1(W_2) = 1$. Then, since f_1 is contained in a finite dimensional G-invariant subspace $V = \text{Span}(G \cdot f_1)$ of $\mathcal{O}(X)$, we have an injective, G-equivariant map $V \hookrightarrow A$. Then, since the action on V induces an action on $\text{Sym}^*(V) = \mathcal{O}(V^\vee)$, we get a G-equivariant map $\text{Sym}^*(V) \to A$. However, both these spaces are rings of regular functions, and so by the equivalence of categories, this map then induces a G-equivariant map $h : X \to V^\vee$ whereby $W_1 \mapsto 0$ and $W_2 \mapsto p$, $p \neq 0$. Now, since G is geometrically reductive there exist a homogeneous polynomial (non-constant), say F, such that $F(p) \neq 0$ and $F(0) = 0$. Thus, the function $f = cF \circ h$ is a sufficient function, whereby $c = 1/F(p)$. □

Proof of Theorem 2.50. We will prove that the GIT quotient is a categorical quotient by proving that it is a good quotient. To see that the GIT quotient satisfies properties 1 and 3, we note that since G is linearly reductive, then by Theorem 2.57 $\mathcal{O}(X)^G$ is a finitely generated k-algebra. Then since the quotient is constructed from the map on the function rings, it is affine and G-invariant as it is an inclusion from a finitely generated G-invariant k-algebra.

For surjectivity, we need to show that any given point in $X//G$ is mapped to by a point in X. To do this, consider the maximal ideal associated to any point y in $X//G$ and take its generators $f_1, \cdots, f_n$. Then we note that by Lemma 2.56 that

$$(\sum f_i\mathcal{O}(X)) \cap \mathcal{O}(X)^G = \sum f_i\mathcal{O}(X)^G$$

and thus $\sum(f_i\mathcal{O}(X))$ contains an element that corresponds to such an x.

As proved in Lemma 2.61, since we already have surjectivity, 4 and 5 are equivalent to proving that for two disjoint invariant closed subsets have the closures of the images disjoint. By Lemma 2.64, there is a function f in $\mathcal{O}(X)^G$ such that $f(W_1) = 0$ and $f(W_2) = 1$. Note however, that this is a function in $\mathcal{O}(X)^G$ and is thus a regular function on $X//G$. Therefore, we have that $f(\phi(W_1)) = 0$ and $f(\phi(W_2)) = 1$ and therefore the closures of their images must also be disjoint.

It remains to prove 6. Firstly, for all $f \in \mathcal{O}(X)^G$, the subsets $X//G_f$ act as a basis for the open subsets of $X//G$. Thus, it suffices to prove the result simply for these sets. Since f is G-invariant we have:

$$\mathcal{O}_X(\phi^{-1}(X//G_f))^G = \mathcal{O}_X(X_f)^G = (\mathcal{O}(X)^G)_f = \mathcal{O}_Y(Y_f)$$

and thus the GIT quotient is a good quotient. □

For future discussion, we denote the GIT quotient as $X//G$.

Example 2.65. If we take the trivial action of an algebraic group on itself, we see that $G//G = G$. One can also end up with the trivial variety. For example consider the action of $\mathbb{Z}/2\mathbb{Z}$ on $\mathbb{C}$ via multiplication by 1 on the identity and -1 not on the identity. The ring of regular functions is $\mathbb{C}[x]$ but the invariant ring is $\mathbb{C}$ and thus our variety $\mathbb{C}//\mathbb{Z}/2\mathbb{Z} = 0$. Further examples appear in the next chapter.

2.4 Stable Points

Thus, we have a geometric object that one could reasonably describe as a quotient space, at least for affine varieties in the affine GIT quotient. Recall however, that we have another quotient set, the set of orbits. Clearly then, the next reasonable question to address is the "sameness" of these two formulations of a quotient space. The answer, as one might expect, is fairly different (and indeed from a geometric context extremely different), and this is best illustrated with an example.

Example 2.66. Consider the action of GL_2 on the set of 2×2 matrices by conjugation. Our ring of regular functions is just polynomials in 4 variables and we have the following fact about the invariant ring:

Remark 2.67. The invariant ring is generated by the trace and the determinant

We see then that the matrices $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & t \\ 0 & 0 \end{bmatrix}$ are identified in the GIT quotient, even though they are not in the same orbits

Thus, these two objects are quite different, and this difference is encapsulated in the property of stability; it is the subvariety of a type of point called a stable point X^s for which we get an equivalence between the GIT quotient and geometric quotient. We define a stable point on an affine GIT quotient as follows:

Definition 2.68. A point $x \in X$ is *stable* if its orbit is closed in X and $\dim(G_x) = 0$. The set of all stable points of X is denoted by X^s.

Example 2.69. Returning to the example above, we see that since the Jordan normal form of a matrix $M = PAP^{-1}$, and is obtained by conjugation, we can find which points are stable by just considering Jordan normal forms. For any diagonal matrix, we see that the stabilizer is all of the other diagonal matrices - this stabilizer is 2-dimensional and so no diagonal matrix is stable. Scalar matrices have stabilizer GL_2 and so their stabilizer is also not zero dimensional, and thus scalars are unstable. What remains then are non-diagonalizable matrices. That is, matrices with linearly dependent eigenvectors. These are matrices of the form $\begin{pmatrix} a & 1 \\ 0 & a \end{pmatrix}$. The stabilizer of such matrices is the scalar matrices - which is a 1 dimensional subgroup. Thus no points are stable

Example 2.70. An example for which the stable points are not the emptyset is the action of $\mathbb{G}_m$ on $\mathbb{A}^2$ by $t \cdot (x, y) = (tx, t^{-1}y)$. Note that every point except the origin has a zero dimensional stabiliser, so the stable points are the points in $\mathbb{A}^2 \setminus \{(0, 0)\}$ that have a closed orbit. Note that no points on the axis have a closed orbit, since limiting t to 0 or infinity gives that the origin is in the closure, while the orbit of the origin is itself. However, every other point is stable. To see this, note that for a point (a, b), t cuts out the curve, $xy = ab$. This is Zariski closed, as it is a polynomial equation. Thus $X^s = \{(a, b) \mid a, b \neq 0\}$

Proposition 2.71. *For a linearly reductive group G acting on an affine variety X and with $\phi : X \to Y := X//G$ the affine GIT quotient, X^s is open and G-invariant. $Y^s = \phi(X^s)$ is open in Y and $X^s = \phi^{-1}(Y^s)$. Furthermore, $\phi : X^s \to Y^s$ is a geometric quotient*

To prove this proposition, we need three facts which we will not prove. They can be found in Hoskins but require some mathematics beyond the scope of this thesis:

Proposition 2.72. *Let G be an affine algebraic group acting group acting on a variety X. Then, for every n, the sets*

$$\{x \in X \mid dim\ G_x \geq n\} \qquad \{x \in X \mid dim\ (G \cdot x) \leq n\}$$

are closed in X. Moreover, $dim(G) = dim(G_x) + dim(G \cdot x)$.

Lemma 2.73. *For $G \cdot x$ an orbit of G in X, the boundary of the orbit $\overline{G \cdot x} \setminus G \cdot x$ is a union of strictly lower dimensional orbits.*

Proof of Proposition 2.71. Suppose x is stable. Then the dimension of its stabiliser is zero. Since we know that the set $\{x \mid \dim(G_x) > 0\}$ is closed, if we can show it is G-invariant then, since G is linearly (and therefore geometrically) reductive, there exists a G-invariant function f such that

$$f(G \cdot x) = 1, \qquad f(\{x \mid \dim(G_x) > 0\}) = 0$$

However, the set is G-invariant, since if $y = g \cdot X$, then $G \cdot y = G \cdot x$, and in particular their dimensions are the same, so the dimensions of their stabilisers are the same.

Thus, we have a function f such that $x \in X_f$ where X_f is open. Now, suppose $y \in X_f$. The dimension of the stabiliser of y must be zero, since otherwise $f(y) = 0$. Furthermore, since the boundary of an orbit is a union of strictly lower dimensional orbits, and elements of the boundary of the orbit of y must be in X_f as X_f is an open set, the stabiliser of elements on the boundary of y also have dimension zero, and so their orbits have the same dimension as the orbit of y, and so the orbit of y must be closed. Hence $y \in X^s$ and X_f is open in X^s and so x has a neighbourhood of points around it also in X^s. This is true for any $x \in X^s$ however, and so X^s must be open.

We note from this that these X_f will cover X^s, and so since $\phi(X_f) = Y_f$, and Y_f is open, then Y^s must be open. In a similar vein therefore $\phi^{-1}(Y^s) = X^s$, since $\phi^{-1}(Y_f) = X_f$.

It remains to show that $\phi^s : X^s \to Y^s$ is a geometric quotient. However, it is not hard to check that the restriction of a good quotient to open sets remains a good quotient, and from there since we know that the orbits of elements in X^s are closed, we can conclude that ϕ^s is a geometric quotient. □

Thus, if X is an affine variety, we can tell whether a point is stable, and if one could find a condition on points in X that implied stability, we could describe X^s, as well as compare it to the Affine GIT quotient on X. However, this has two major problems. Firstly, this process only works for Affine varities, which is a relatively small number of varieties. The second, more major problem is that finding this condition on points is often extremely difficult. To solve these issues, we will define the projective GIT quotient, and using it, we will find a numerical criterion for stability that is much easier to work with.

To define the projective GIT quotient, we first need to know what a linear G-action on a projective variety means:

Definition 2.74. Let X be a projective variety with an action of an affine algebraic group G. A *linear G-equivariant projective embedding* of X is a group homomorphism $G \to GL_{n+1}$ and a G-equivariant projective embedding $X \hookrightarrow \mathbb{P}^n$. We will reduce this to saying that the G-action on X is *linear* to mean that we have a linear G-equivariant projective embedding of X.

So now suppose we have such a linear action of a linearly reductive group G on some X. Since X is projective, it lifts to the affine cone of X, $\tilde{X}$. Further, since we have an embedding for X in $\mathbb{P}^n$, $\tilde{X}$ is contained in the lift of the affine cone of $\mathbb{P}^n$: $\mathbb{A}^{n+1}$. The action of G lifts to $\mathbb{A}^{n+1}$ and in particular, since the embedding of X is G-equivariant, we get an induced action of G on $\tilde{X}$. That is, if X associates to the homogeneous ideal $I(X)$, then $\tilde{X} = \mathrm{Spec}R(X)$ where $R(X) = k[x_0, \cdots, x_n]/I(X)$. Note that since $R(X)$ is graded by homogeneous degree and the G-action is linear, we get a grading on $R(X)^G$ by the homogeneous degree. Note further since G is linearly redutive, $R(X)^G$ is finitely generated and thus the inclusion $R(X)^G \hookrightarrow R(X)$ induces a rational map of projective varieties

$$X \dashrightarrow \mathrm{Proj}R(X)^G.$$

The fact this is only a rational map motivates the following definition:

Definition 2.75. For a linear action of a reductive group G on a projective variety X, we define the *nullcone* to be the closed subvariety of X defined by $R(X)^G_+$. Furthermore, we define the *semistable set* as the open set $X^{ss} = X - N$. That is, $x \in X$ is *semistable* if there exists a G-invariant homogeneous function $f \in R(X)^G_r$ for $r > 0$ such that $f(x) \neq 0$. Also, we see that via this construction, the semistable set is the set for which the rational map mentioned earlier is defined.

Thus, the morphism $X^{ss} \to X//G = \mathrm{Proj}R(X)^G$ is the *projective GIT quotient*

This is all well and good, but we want to be sure like its affine counterpart that it is a good quotient

Theorem 2.76. *For a linear action of a reductive group G on a projective variety X, the GIT quotient $\phi : X^{ss} \to X//G$ is a good quotient (and therefore a categorical quotient).*

Proof. To prove that the projective GIT quotient is a good quotient, note that being a good quotient is a local property. Then, for $f \in R(X)^G_+$, note that (if our GIT quotient maps $X^{ss} \to Y$) the sets Y_f form a basis of open sets on Y. Furthermore, we have that $\phi^{-1}(Y_f) = X_f$. Then, if we consider the affine cones of X_f and Y_f, say $\tilde{X}_f$ and $\tilde{Y}_f$. Then we have that

$$\mathcal{O}(Y_f) \cong \mathcal{O}(\tilde{Y}(f))_0 \cong ((R(X)^G)_f)_0 \cong ((R(X)_f)_0)^G \cong (\mathcal{O}(\tilde{X}_f)_0)^G \cong \mathcal{O}(X_f)^G$$

Thus, the localisation of the quotient to the affine schemes $\phi_f : X_f \to Y_f$ is an affine GIT quotient, and is therefore a good quotient. Then, as mentioned earlier, since being a good quotient is a local property and we can get to our original quotient ϕ by gluing all the ϕ_f morpisms, ϕ is also a good quotient. □

However, we were trying to understand how different the GIT quotient was from the geometric quotient. With semistability in hand we can define stability for projective varieties:

Definition 2.77. For a linear action of a reductive group G on a closed projective variety X, a point $x \in X$ is

1. *stable* if the dimenstion of its stabiliser $\dim G_x = 0$ and there is a G-invariant homogeneous polynomial $f \in R(X)^G_+$ such that $x \in X_f$ and the action of G on X_f is closed
2. *unstable* if it is not semistable

Thus, all that remains to prove is that on the stable points, the GIT quotient restricts to the geometric one, and to do that we need the following lemma:

Lemma 2.78. *The stable and semistable sets are open in* X

Proof. Firstly, we note that since we defined X^{ss} as the complement of the Null-cone, it is clearly open. Then, consider the set $X' = \bigcup X_f$ for f in $R(X)^G_+$ where we take the union such that it preserves the closed nature of the action of G on X_f. Then, since X' is a union of open sets, it is an open set in X, and thus if X^s is open in X', we're done. However, proposition 2.72 gives us this openness property since $X' - X^s = \{x \in X' \mid \dim G_x \geq 1\}$ is a closed set whose complement is precisely the set with zero dimensional stabilisers, X^s. □

Theorem 2.79. *For a linear action of a reductive group* G *on a closed projective variety* X*, let* $\phi : X^{ss} \to X//G$ *denote the GIT quotient. Then there is an open subvariety of* $X//G$*:* $(X//G)^s$ *such that* $\phi^{-1}((X//G)^s) = X^s$ *and that the GIT quotient restricts to a geometric quotient*

Proof. First, note that any geometric quotient is also a good quotient. Furthermore, note that for any good (or geometric) quotient $\phi : X \to Y$ and any open subset U of Y, then the restriction of the quotient to the preimage of U is also a good (or geometric) quotient. For a final piece of setup, also note that if we have a open cover of Y, the sets U_i, such that the restriction of ϕ to the preimages of these U_i are good (or set theoretic) quotients then ϕ is a good (or set theoretic) quotient. With this information in hand, consider the set $Y' = \bigcup Y_f$ where $f \in R(X)^G_+$ and that the action of G on X_f is closed. Furthermore, let X' be the preimage under ϕ of Y'. Then the map $\phi : X' \to Y'$ can be constructed via the gluing of the ϕ_f. Note that each ϕ_f is a good quotient since they are the restriction of the image of an open set under a good quotient. Additionally, since the G-action on X_f is closed, we have that this is a geometric quotient. Thus, it remains to show that $\phi(X^s)$ is open. Note that since X^s is a G-invariant subset, $\phi(X' - X^s) = Y' - Y^s$. Remember that by the fact mentioned in Lemma 2.78, $X' - X^s$ is closed, and therefore since ϕ is a good quotient, so is $Y' - Y^s$ via property 5. Thus, Y^s is open in Y' and since Y' is open in Y, so is Y^s. □

2.5 The Numerical Criterion

So we now know how close the GIT quotient is to the geometric one for both affine and projective GIT quotients, but it would be better if we had a way to detect unstable points that was easier to compute with. Such a detection method exists, and is known as the Hilbert-Mumford Numerical Criterion. To prove that it works and makes sense, we need to provide some less nice criteria to build from:

Lemma 2.80. *Let G be a reductive group acting linearly on X. A point x is stable if and only if x is semistable, its orbit is closed in X^{ss} and its stabiliser G_x is zero dimensional*

Proof. First, note that if x is stable, then there exists an $f \in R(X)^G_+$ such that $x \in X_f$ so x is semistable. We also already know that the dimension of the stabiliser G_x is zero. Thus, it remains to show the orbit of x is closed in X^{ss}. Suppose $x' \in \overline{G \cdot x}$. Then, since the quotient ϕ is G-invariant, $\phi(x') = \phi(x)$, and so $x' \in X^s$, and therefore in X^{ss}. Thus, it remains to show x' is in the orbit. However, we know that orbits on the boundary are unions of orbits of strictly lower dimension, but x' is stable, and so the dimension of its stabiliser is zero, and so its orbit has dimension the same as the dimension of x, so x' must be in the orbit of x.

Now, suppose $x \in X^{ss}$, $G \cdot x$ is closed in X^{ss}, and the dimension of the stabiliser G_x is zero. Then, since $x \in X^{ss}$, there exists some $f \in R(X)^G_+$ such that $x \in X_f$. Now, since $G \cdot x$ is closed in X^{ss}, it is closed in X_f. Further, it is disjoint from the set of points

$$Z = \{y \in X_f | \dim(G_y) > 0\}$$

Thus, there exists some $h \in O(X_f)^G$ such that $h(G \cdot x) = 1$ and $h(Z) = 0$. Now, $O(X_f) = O(\tilde{X}_f)_0$ and, in particular, is the quotient of some algebra $A = (k[x_0, \cdots, x_n]_f)_0$, where the kernel I is some ideal. Then, by the lemma below, we have that there exists a positive integer r such that $h^r \in A^G/(I \cap A^G)$, but A is localised at f so in particular we have that $h^r = h'/f^s$, where $h' \in O(X)^G$.

Now, $fg = h^r f^{s+1}$, and so $x \in X_{fg}$. Thus, for $y \in X_f$, $y \in X_{fg}$ only if the stabiliser of y is dimension zero. Therefore, $h' = fg$ is a polynomial in $O(X)^G$ whereby $x \in X_{h'}$ and all orbits of elements in $X_{h'}$ are closed, and so x is stable. □

Lemma 2.81. *Let G be a geometrically reductive group acting rationally on a finitely generated k-algebra A. For a G-invariant ideal I of A and $a \in (A/I)^G$, there is a positive integer r such that $a^r \in A^G/(I \cap A^G)$.*

Proof. Let $b \in A$ be an element in the preimage of a and further assume $a \neq 0$. We have that G acts rationally and so b is contained in a finite dimensional G-invariant vector space V spanned by $g \cdot b$. Since a is nonzero $b \notin V \cap I$ but $g \cdot b - b \in V \cap I$ as $\pi(g \cdot b - b) = \pi(g \cdot b) - \pi(b) = a - a$. Thus, dim $(V) =$ dim $(V \cap I) + 1$ and we can write any $v \in V$ as $\lambda b + b'$ for a scalar λ and an element in $V \cap I$. The linear projection from V to k onto the line spanned by V is G-equivariant, and so, in the dual representation, the projection corresponds to a nonzero fixed point, say l^*. Furthermore, since G is geometrically reductive (and has an induced action on the dual representation), there exists a G-invariant homogeneous function F of positive degree r which doesn't vanish at l^*. To complete the proof, note that first, we have the algebra homomorphism as in the proof of theorem 2.57

$$\mathcal{O}(V^\vee) = \mathrm{Sym}^*(V) \to A$$

and second we can take a basis of V such that the first basis vector corresponds to b. Then, $F = \lambda x_1^r + \cdots$, $\lambda \neq 0$. Then $F - \lambda x_1^r$ is a polynomial in which every monomial contains a power (potentially zero) of x_i, $i \neq 1$ Thus, for b_0 in the image of F in A^G, we have that $b_0 - \lambda b^r \in I$ as the images of the remaining x_i are elements in $V \cap I$ and so $a^r \in A^G/(I \cap A^G)$ □

So with that criterion set up, we can move to a topological criterion.

Proposition 2.82. *Let x be a point of X and choose a nonzero lift into the affine cone $\tilde{x}$. Then*

1. *x is semistable iff $0 \notin \overline{G \cdot \tilde{x}}$*
2. *x is stable if and only if dim $G_{\tilde{x}} = 0$ and $G \cdot \tilde{x}$ is closed in $\tilde{X}$.*

Proof. First, suppose that x is semistable. Then, there exists f such that $f(x) \neq 0$ and so $f(\tilde{x}) \neq 0$. But G-invariant functions are constant on orbits and their closures, so $f(\overline{G \cdot \tilde{x}}) \neq 0$ and so $0 \notin \overline{G \cdot \tilde{x}}$

Suppose $0 \notin \overline{G \cdot \tilde{x}}$. Then there exists f such that $f(\overline{G \cdot \tilde{x}}) \neq 0$ and $f(0) = 0$. We can write $f = \sum f_i$ where the f_i are homogeneous, and the action of G is linear, so the f_i are G-invariant, and at least one must be nonvanishing at $\tilde{x}$ and so is nonvanishing at x. Hence, x is semistable.

Suppose x is stable. Since $G_{\tilde{x}} \subset G_x$, the dimension of $G_{\tilde{x}}$ is zero. There also exists f homogeneous such that $f(x) \neq 0$, and so $f(\tilde{x}) \neq 0$. Let

$$Z = \{z \in \tilde{X} \mid f(z) = f(\tilde{x})\}$$

and let $\pi : Z \to X_f$ be the map obtained from restricting the surjection $\tilde{X} \setminus 0 \to X$. Suppose $y \in \overline{G \cdot \tilde{x}}$. Then $\pi(y) \in G \cdot X$, and so the dimension of $G_{\pi(y)} = 0$, but that means that G_y has dimension 0 and so the orbit of y must have dimension $\dim G$ and so if y isn't in the orbit of $\tilde{x}$, then $G \cdot \tilde{x}$ must have dimension bigger than that of G, which is a contradiction.

For the reverse direction, we know that $0 \notin G \cdot \tilde{x}$, since otherwise $\tilde{x} \in G \cdot 0$. Thus, $0 \notin \overline{G \cdot \tilde{x}}$ since $G \cdot \tilde{x}$ is closed, and so by part 1 of this proposition, x is semistable, and so there exists f such that $f(x) \neq 0$. Let Z be as before , and since f is homogeneous of degree d, say, $Z \cap \pi^{-1}(x)$ is the set of points λx such that $\lambda^d f(x) = 1$. That is, π has finite fibers. Now, for $g \in G_x$, since x has finite preimages say $\tilde{x}, \tilde{x_1}, \cdots, \tilde{x_n}$, $g\tilde{x} = \tilde{x}$ or $\tilde{x_i}$. If we denote $G_i = \{g \mid g\tilde{x} = \tilde{x_i}\}$ and $g_i \in G_i$, then G_i is isomorphic (as a variety) to $G_{\tilde{x}}$ by left multiplication by g_i, and since $G_{\tilde{x}}$ has dimension 0, all the G_i do. But $G_x = G_{\tilde{x}} \sqcup G_1 \sqcup \cdots \sqcup G_n$, so G_x has dimension 0. Finally, pick $y \in \overline{G \cdot x}$ strictly on the boundary. That is, assume the orbit is not closed. Then, since $\pi(G \cdot \tilde{x}) = G \cdot x$, then $\tilde{y} \notin G \cdot \tilde{x}$ That is, there exists h such that $h(\tilde{y}) \neq 0$, but $h(G \cdot \tilde{x}) = 0$. Thus, there exists a homogeneous component h_i for which $h_i(\tilde{y}) \neq h_i(G \cdot \tilde{x})$. That is, $h_i(y) \neq h_i(G \cdot x)$, but y is in the closure, and so this is a contradiction. Hence $G \cdot x$ is closed in X_f, and so $G \cdot x$ is closed in $X^{ss} = \bigcup X_f$ and therefore x is stable by Lemma 2.80. $\square$

Now we can begin to discuss the Hilbert-Mumford criterion but first we need to understand 1-parameter subgroups:

Definition 2.83. A *1-parameter subgroup* of G is a non-trivial group homomorphism $\lambda : \mathbb{G}_m \to G$.

Now, we can state the Hilbert-Mumford Criterion:

Theorem 2.84. *Let G be a (linearly) reductive group acting linearly on a projective variety $X \subset \mathbb{P}^n$. Then, for $x \in X$, we have*

1. $x \in X^{ss} \iff \lim_{t \to 0} \lambda(t) \cdot \tilde{x}$ *is either nonzero or doesn't exist for all* 1*-PSs* λ *of* G
2. $x \in X^{s} \iff \lim_{t \to 0} \lambda(t) \cdot \tilde{x}$ *doesn't exist for all* 1*-PSs* λ *of* G.

In order to prove the criterion, we must first recall a few definitions. The first is of a local ring at a point:

Definition 2.85. A ring is a *local ring* if it has a unique maximal ideal. The local ring of a variety X at a point P is the ring

$$\{f/g \mid f, g \in \mathcal{O}(X), g(P) \neq 0\}$$

The next is the normalization of a variety

Definition 2.86. A variety is *normal* if at every point $x \in X$ the local ring $\mathcal{O}(X)_x$ is an integrally closed domain. The normalization of a variety is the unique morphism $\phi : Y \to X$ such that Y is normal and for any other normal variety Z and a map $\psi : Z \to X$ dominant, we have a unique morphism $\theta : Z \to \tilde{Y}$ with $\phi = \psi \circ \theta$. Intuitively, this can be thought of (at least for curves) as the removal of singularities.

Lastly, we have the completion of a ring

Definition 2.87. Let A be a ring with a descending filtration

$$A = F^0 A \supset F^1 A \supset \cdots$$

of subrings. The *completion* A' of the ring A is the inverse limit

$$A' = \lim_{\leftarrow}(A/F^n A)$$

Next, we need to prove two lemmas. For each let G be a linearly reductive group acting linearly on $\mathbb{A}^n$ and let $z \in \mathbb{A}^n$ be a k-point. Further, suppose 0 lies in the orbit closure of z and let σ_z be right multiplication of elements of G by z:

Lemma 2.88. *There exists an irreducible curve $C_1 \subset G \cdot z$ that contains the origin in its closure*

Proof. Begin by fixing an embedding of affine n-space into Projective n-space, and let p denote the image of the origin. Let $Y = \overline{G \cdot z} \subset \mathbb{P}^n$, let $Z = Y \setminus G \cdot Z$ and let $d = \dim(Y)$. Note that we can assume that $d > 1$ as if $d = 1$, we have a curve that we can obtain C_1 from via removing points in Z. For $n > 1$, we can use an intersection of hyperplanes to produce the required curve. Given the set of hyperplanes containing p is a codimension 1 subspace $\mathcal{H}_p$, taking $(d-1)$ copies of this subspace that contain hypersurfaces $H_1, \cdots H_{d-1}$ with the following two conditions:

1. $\cap_i H_i \cap Y$ is a curve (that is has dimension 1)
2. $\cap_i H_i \cap Y$ is not entirely contained in Z

Gives us a curve (nonempty open dimension 1 subspace) $C_1' = \cap_i H_i \cap Y$ that isn't entirely contained in Z, since the two conditions are nonempty open conditions, and the dimension of $\mathcal{H}_p^{d-1}$ is greater than zero. Then, we retrieve the required curve C_1 by removing the points in Z. □

Lemma 2.89. *There exists a curve $C_2 \subset G$ which dominates C_1 under the group action morphism σ_z.*

Proof. To prove such a curve exists, use the same technique as in the previous lemma except on the preimage σ_z^{-1} to construct a curve $C_2' \subset \sigma_z^{-1}(C_1)$. Since G acts linearly, dimensions are preserved. Furthermore, open sets are also preserved under preimage since the action morphism is a morphism in the category of varieties. □

Now, we can begin the proof:

Proof. To start with, note that because of Lemma 2.82 the Hilbert Mumford Criterion reduces to the following theorem.

Theorem 2.90. *A point $x \in X$ with lift $\tilde{x}$ is unstable if and only if there exists a one parameter subgroup λ such that*

$$\lim_{t \to 0} \lambda(t) \cdot \tilde{x} = 0$$

The reverse direction is clear. If such a subgroup exists, then zero is clearly in the closure of the orbit, as it is in the closure of the one parameter subgroup. Thus, by Lemma 2.82, the point is unstable. The forward direction requires considerably more work:

To start with, we use Lemma 2.88 to provide us with a curve $C_1 \subset G \cdot z$ that contains zero in its closure.

Then, we use Lemma 2.89, to get the curve C_2. Then, we get a curve C by taking the projective completion of the normalization of C_2, where the normalization is $\tilde{C}_2 \to C_2$. Then, we have a rational map $\rho : C \dashrightarrow G$ defined by the maps $\tilde{C}_2 \to C_2 \to G$, and thus we have a preimage of the origin in C (since the origin is contained in the closure of C_1), say c_0. Then $\lim_{c \to c_0} \rho(c) \cdot z = \lim_{c \to c_0} \sigma_z(\rho(c)) = 0$.

Since C is smooth (we took the normalization), the completion of its local ring at c_0, $\mathcal{O}_{C,c_0}$ is isomorphic to the power series ring $k[[t]]$ by the Cohen Structure Theorem [4]. The field of fractions of the power series ring is the Laurent series ring $k((t))$, since ρ was defined in a punctured neighbourhood of c_0 we get an induced morphism on Spec Frac of the completion of the local ring down:

$$q : \text{Spec } k((t)) \cong \text{SpecFrac}\hat{\mathcal{O}}_{C,c_0} \to \text{SpecFrac}\mathcal{O}_{C,c_0} \xrightarrow{\rho} G$$

Thus $\lim_{t \to 0} q(t) \cdot z = 0$.

Next, let $R = \text{Spec}k[[t]]$, $K = \text{Spec}k((t))$. Note that there is a morphism $K \to R$ induced by the inclusion of the ring of regular functions and so the R-valued points of G form a subgroup of the K-valued ones. Further the limit as $t \to 0$ of the R-valued G-points exists, since the powers of t in R are all positive. Now, for any one parameter system λ, we can define its Laurent series expansion (in $G(K)$) in the following way: there is a natural morphism from $K \to \mathbb{G}_m$ induced from the morphism on their group rings $k[s, s^{-1}] \to k((t))$ by $s \mapsto t$. Define the expansion then to be the composition of λ with this map from $K \to \mathbb{G}_m$

Now, we relate our map q to such a Laurent series expansion using the Cartan-Iwahori decomposition. That is, every double coset in $G(K)$ for $G(R)$ is represented by a Laurent series expansion of a one parameter system $\langle \lambda \rangle$. Following this proof, we will prove this decomposition for $G = SL_n$, the main group of study in this thesis. A more general proof can be found in [9], but for now we will use this decomposition without proof. From it, we see that there exists a Laurent expansion $< \lambda >$ and two $G(R)$ points l_1, l_2^{-1} such that $l_1 q l_2^{-1} =< \lambda >$. That is, $l_1 q =< \lambda > l_2$. Note that since $q \notin G(R)$, $< \lambda >$ cannot be trivial.

Let $g_i = l_i(0) \in G$. Then, we have that

$$0 = g_1 \cdot 0 = \lim_{t \to 0} l_1[q(t) \cdot z] = \lim_{t \to 0} (l_2 < \lambda >)(t) \cdot z$$

Now, the action of λ (not $< \lambda >$) decomposes into weight spaces V_r, $r \in \mathbb{Z}$. Furthermore, since $l_2 \in G(R)$ and $g_2 = \lim_{t \to 0} l_2(t)$, we know that l_2 decomposes into g_2 plus some positive powers of t, say $\epsilon(t)$ which go to zero when we take the limit. Under our weight space decomposition:

$$l_2(t) \cdot z = g_2 \cdot z + \epsilon(t) = \sum_{r \in \mathbb{Z}} (g_2 \cdot z)_r + \epsilon(t)_r$$

However, we have that $\lim_{t \to 0}(l_2 < \lambda >)(t) \cdot z = 0$, and so $(g_2 \cdot z)_r = 0$ for $r \leq 0$. That is, there are no negative powers of t in $g_2 \cdot z$, and so the limit $\lim_{t \to 0} \lambda(t) \cdot g_2 \cdot z = 0$ and so we have a one parameter system $\lambda' = g_2^{-1} \lambda g_2$ such that $\lim_{t \to 0} \lambda'(t) \cdot z = 0$, completing the proof. □

The final element that remains is the proof of the Cartan-Iwahori decomposition in the case where $G = SL_n$. That is,

Theorem 2.91. *Every double coset in $n \times n$ matrices whose entries are Laurent series for $n \times n$ matrices whose entries are power series is represented by a laurent series expansion of a one parameter system. That is, there exist matrices of power series M and N, such that $MKN = K'$ where K' is a diagonal matrix with laurent series entries.*

Proof. First, note that if this is possible in GL_n, say with $\langle \lambda \rangle = l_1 q l_2$, then it is possible in SL_n with $\frac{l_1}{\det(l_1)} q \frac{l_2}{\det(l_2)}$, and furthermore that row and column operations are matrices in GL_n. Next, note that K is the fraction field of R, so entries in elements of $G(K)$ are in fact $\frac{p(x)}{q(x)}$, for p and q power series. Now, power series can all be written in the form $x^\alpha p(x)$ whereby $p(0) \neq 0$. That is, $p(0)$ is invertible. Thus

$$\begin{aligned} f(x) = \frac{p(x)}{q(x)} &= \frac{x^{\alpha_p} p'(x)}{x^{\alpha_q} q'(x)} \\ &= x^{\alpha_p - \alpha_q} \frac{p' x}{q' x} \end{aligned}$$

But $p'(x)$ and $q'(x)$ are nonzero at 0, so $\frac{p'(x)}{q'(x)}$ is a power series $f'(x)$ which is also nonzero at 0. Thus, every entry in an element of $G(K)$ is of the form $x^\alpha p(x)$ where $\alpha \in \mathbb{Z}$ and $p(x)$ is a power series where $p(0) \neq 0$.

Now, using row and column operations, move the entry with the smallest negative power into the top left corner (for notation we label the ij-th entry of the matrix as $x^{\alpha_{ij}} p_{ij}(x)$)) Now for column j, multiply the column by $x^{\alpha_{11} - \alpha_{1j}} p_{1j}(x)^{-1} p_{11}(x)$.

The entry in space $1, j$ then becomes the same as the entry in $1, 1$. Then subtract column 1 from column j. Doing this for all columns j zeroes every element in the first row except the diagonal entry. We can then do the same process to zero the first column, instead working with rows i. This creates a matrix whose first diagonal entry is nonzero, the first column and row are otherwise zero with an $n-1 \times n-1$ minor. We can then repeat this process with the minor provided that the row and column operations preserve the form of the entries. That is, provided the new laurent series in entry i, j, say, still has $p_{ij}(0) \neq 0$.

Without loss of generality, we can check this for a single operation. Consider an entry i, j and assume that the multiplication step has been done, to take $x^{\alpha_{ij}} p_{ij}(x)$ to some $x^\beta q(x)$. Now subtract (again without loss of generality) column 1. We then have the i, j entry is $x^\beta q(x) - x^{\alpha_{i,1}} p_{i,1}(x)$. One of the powers β or $\alpha_{i,1}$ must have larger (or equal) absolute value. Since the only difference is a factor of -1, assume β is the larger absolute value. Then our i, j entry is $x^\beta(q(x) - x^{\alpha_{i,1}-\beta} p_{i,1}(x))$ where the factor inside the bracket is now a power series. Label this power series $q'(x)$. If $|\beta|$ is directly larger, $q'(0) = q(0) \neq 0$, so the only case is where $\beta = \alpha_{i1}$. Then $q'(0) = 0$ only when $q(0) = p_{i,1}(0)$. Thus,

$$
\begin{aligned}
q'(x) &= (a + \sum_{i=1}^{\infty} a_i x^i - (a + \sum_{i=1}^{\infty} b_i(x^i))) \\
&= \sum_{i=1}^{\infty} (a_i - b_i) x^i
\end{aligned}
$$

where $q(0) = p_{i,1}(0) = a$, and the a_i and b_i are coefficients. Notice that if the two power series are completely equal, then in the previous subtraction process the i, j entry became zero, and so we don't need to apply any operations. Thus, there must exist some power of x whereby $a_i \neq b_i$, say m. Then we can write $f(x) = x^{\beta+m}(a_m - b_m + \sum_{i=1}^{\infty}(a_i - b_i)^{x^i}) = x^{\beta+m} q'(x)$. This now has $q'(0) \neq 0$ and so the form of the entries is maintained. $\square$

Chapter 3

Computation and Classification

3.1 Some Basic Convex Geometry

In the last chapter we established a powerful numeric criterion for stability. However, it was a check on individual points, of which there may be infinitely many. Therefore finding the set of unstable points simply using the criterion naively is next to impossible. The section following on from this one will explain a method of translating a numeric problem with limits into a convex geometry problem with halfspaces. For that reason, we will take a brief interlude into some convex geometry, covering some basic definitions and one or two useful facts - in particular, about polyhedra.

For what follows, all sets are contained in $\mathbb{R}^n$. We first define a convex set:

Definition 3.1. A set A is *convex* if, for every two points $x, y \in A$, we have that $\lambda x + \mu y \in A$ for any $\lambda, \mu \geq 0$ and $\lambda + \mu = 1$

Given a set we can also define its convex hull

Definition 3.2. The *convex hull* conv A of a set A is the smallest convex set containing A.

It's important to note that this set will always exist. We can take the intersection of all convex sets containing A, and at least one such set exists, since the ambient space for A, say $\mathbb{R}^n$, is convex.

From here, we can define a polyhderon

Definition 3.3. A *polyhedron* is the intersection of a finite family of closed halfspaces. That is, it is the set of all points $(x_1, \cdots, x_n)$ which satisfy a finite system of linear inequalities. The set of such halfspaces for a polyhedron P is called the *halfspace representation* of P.

Remark 3.4. Both the intersection of convex sets and halfspaces are convex. Thus, polyhedrons are convex objects.

This idea of a halfspace representation is what we will rely on heavily for our computation. We require one last piece of information to explain in greater detail how one moves from the Numerical Criterion to a question about convex sets: the Hahn-Banach Separation Theorem:

Theorem 3.5 (Hahn-Banach). *Let X be a real topological vector space, A a non-empty convex open subset of X, and $x_0 \notin A$. Then there exists a continuous $\lambda \in X^*$ such that $\lambda a - x_0 > 0$ for every $a \in A$*

3.2 The Connection with Convex Geometry

As we established at the end of the previous chapter - we have a numerical criterion for the instability of points of $X = \mathbb{P}V$ under the action of G, the Hilbert-Mumford Numerical Criterion:

Theorem 3.6. *For a point $x \in X$, and a lift $\tilde{x}$ into the affine cone, x is unstable if there exists a one parameter subgroup λ_t such that*

$$\lim_{t \to 0} \lambda_t \tilde{x} = 0$$

To translate this into a problem involving convex geometry, we need the following four facts:

1. G has (at least one) maximal torus T
2. Any two maximal tori are conjugate [7, 21.3]
3. Each one parameter subgroup is contained in some maximal torus T
4. Any representation of T decomposes into a finite direct sum of one-dimensional representations

Note that we discussed the last in Example 2.33 of the previous section, whilst facts one and three are clear. Since every maximal torus is conjugate to any other, we can solve the problem of unstable points in $X = \mathbb{P}V$ up to the G-action by fixing a maximal torus T using the method described below.

Suppose we fix a maximal torus T and this torus is n-dimensional. That is, $T \cong \prod_{i=1}^{n} \mathbb{G}_m$. Let T be parametrised by the tuples $(t_1, \cdots, t_n)$. Now consider any one dimensional representation of this torus. That is, a morphism $\phi : T \to \mathbb{G}_m$. As discussed in Example 2.33, this representation must be monomial. That is

$$\phi(t_1, \cdots, t_n) = t_1^{a_1} \cdots t_n^{a_n}$$

Thus, this representation is specified by an n-tuple of integers $(a_1, \cdots, a_n)$.

Now, since any finite-dimensional representation A decomposes into a finite direct sum of one-dimensional representations, we can write $A = \bigoplus_{i=1}^{m} a^i$, whereby, due to our bijection between representations and $\mathbb{Z}^n$, we can think of a^i as some n tuple of integers $(a_1, \cdots, a_n)$. That is, $a^i = k(a_1, \cdots, a_n)$ where k is the bijection. Furthermore, we have for each of these one dimensional representations a one dimensional invariant subspace. That is, to each a^i we have a subspace of V generated by some w_i. Thus, due to the direct sum decomposition, these w_i form a T-eigenbasis for V and therefore any element $v \in V$ can be written as a sum $v = \sum_{i=1}^{m} c_i w_i$. Thus, if we label the c_i as v_i we see that V is similarly in bijection with a set of n-tuples $(v_1, \cdots, v_m)$.

In practice, since $X = \mathbb{P}V$ for some V, and our action is linear, we have an action of T on V. In other words, we have a representation. Thus, our process for converting into convex geometry will be to first find a T-eigenbasis for V and then use this T-eigenbasis to find a decomposition into one-dimensional representations a^i. These a^i then become points in $\mathbb{R}^n$ on which we can do some convex geometry.

Now, our one parameter subgroup maps into T as follows:

$$t \mapsto (t^{\lambda_1}, \cdots, t^{\lambda_n})$$

Again, we can write this map as an n-tuple $\lambda = (\lambda_1, \cdots, \lambda_n)$. We therefore have the action of λ on v as

$$\lambda(t).v \mapsto (t^{\lambda \cdot a^1} v_1, \cdots, t^{\lambda \cdot a^m} v_m)$$

Therefore, for the point to be unstable, for each nonzero v_i we need $(\lambda_1, \cdots, \lambda_n) \cdot a^i$ to be positive. Thus, the criterion for instability with respect to the torus T becomes the following:

Theorem 3.7. *Let T be a maximal torus for G and let x be a point in X with lift $\tilde{x}$. Furthermore, let S be the set of a^i corresponding to the nonzero v_i in $\tilde{x}$. A point x is unstable (with respect to T) if and only if the convex hull of S does not contain zero.*

Proof. If a point is unstable then there exists $\lambda \in \mathbb{Z}^n$ such that $(\lambda_1, \cdots, \lambda_n) \cdot a^i$ is positive for all i. That is, for S the set containing the a^i, then there is a halfspace separating S and zero defined by $\lambda = (\lambda_1, \cdots, \lambda_n)$. But halfspaces are convex, and since the convex hull of S is the smallest such convex set containing S, it is contained in the halfspace. Therefore, zero is not contained in the convex hull.

If zero is not contained in the convex hull of the a^i then by Hahn Banach there exists a linear functional such that for all v' in the convex hull of the a^i $\lambda(v' - 0) > 0$. In particular, this is true for the a^i.

However, Hahn-Banach gives us a real valued line, not an integer-valued one. We produce an integer-valued line in the following way: since the rationals are dense within the reals and the number of a^i is finite, we can find a rational line arbitrarily close to a real line. Then this rational line can be turned into an integer one by clearing denominators. □

Thus, we have formed a clear connection between a numerical criterion and a finite problem in convex geometry. If, having fixed a maximal torus T, we can find all the maximal convex sets of points a^i that do not include the origin we have found the sets of points $v = (v_1, \cdots, v_m)$ of X that are destabilised by one parameter subgroups (defined by λ) contained in T. That is, since all tori are equivalent up to conjugation, we have found the unstable points for the action of G on X up to conjugation of the one-parameter subgroups. That is, all of the unstable points up to the G-action. The beauty of this method is that it is a finite search problem - and so we can utilise a computer in order to find the convex sets in complicated, multidimensional cases.

3.3 The Program

In the previous section, we established that there is a clear goal for computing unstable points - find the maximal convex sets which do not contain the origin. That is, for a finite set S of points in $\mathbb{R}^n$, we have to find subsets T whose convex hulls do not contain the origin and are maximal. Here, maximality means that adding an additional point would include the origin in the convex hull. However, this comes with some computational challenges. Since the cardinality of the power set is 2^n for n the cardinality of S (as a set), computing all possible combinations of points manually is very computationally expensive. For this

reason, computing maximal convex sets (not containing zero) via working with polyhedra is the method used in the program that does the computational work within this thesis. This also has the added benefit of being able to utilise a powerful python add-on known as Sage, which provides functions to do some of the more brute-force computation, such as identifying whether the origin was present in a given convex set.
The initial attempt at creating such a program was used with the specific example of GL_3 acting on degree n homogeneous polynomials (where n could be varied). This particular example was simpler, as through a simple affine transformation, one is able to take the points in $\mathbb{R}^3$ and convert them into points in $\mathbb{R}^2$, whereby the algorithm boils down to rotating a hyperplane around $\mathbb{R}^2$. Then, since the points also formed a triangle, one could then compute a minimal increment of rotation for the hyperplane, as well as prevent excess computation due to the rotational symmetry of the points. The code is attached in appendix 3.

The second program was designed to be for any group acting (under the assumed conditions) on any algebraic variety X. Here, a more complicated algorithm was necessary:

1. Begin with a halfspace not containing any of the points that are being considered. Call the set of points S and the intersection with any halfspace H, H_s.

2. Add this halfspace to a list of halfspaces required to be checked - say $L1$

3. Take a halfspace H from the list $L1$, and for each point $x \in X$, add x to the halfspace, and so creating a new polyhedron $H' = \text{conv}(H_s \cup x)$ for each x

4. Take the halfspace representation of H' and for each halfspace check whether zero is contained inside it.

5. If not, add the halfspace to the list of new halfspaces, say $L2$ (note this list is specific to each H)

6. If the list $L2$ of new halfspaces is empty, this means that no extra points could be added without adding the origin, and so we have a maximal convex set. Add this to a list of maximal sets $L3$.

7. If the list $L2$ is not empty, check it against the list of halfspaces $L1$ to check for duplicates and for non-duplicates, add them to the list of halfspaces to check ($L1$).

8. Repeat this for every polyhedron in the list of halfspaces to check ($L1$)

Note this algorithm will eventually terminate, since the number of points is finite, and so the number of halfspaces must also be. Furthermore, by building halfspaces in this way, the ordering of the points does not matter, and so we avoid checking a large number of potential sets. The algorithm works additively - halfspaces that are added to the list contain a convex set (that doesn't contain zero) of the points in X, so although ordering of points doesn't matter, we find every possible convex set (that doesn't contain zero) containing each individual point.

Again, Sage is instrumental here. It allows for the creation of Polyhedron objects via both their hyperplane and vertex representations. Furthermore, Sage contains functions that allow to check whether zero is contained in the polyhedron. The code for this iteration of the program is attached in appendix 2. It is also worth noting that points are read in from a csv file, allowing for simple user input.

Since the primary example we work with is cubal $n \times n \times n$ matrices acted on by SL_n^3, we have symmetries of $S_n \times S_3$ and so some regular expression analysis is used to cut out these additional hyperplanes, dramatically reducing the output data and thereby making the remaining human computation simpler.

3.3.1 Complexity

It is worthwhile, however, to check that this program indeed has lower complexity than the naive algorithm (found in Appendix 2). The naive algorithm operates as follows:

1. For the set of points S, generate the power set of S.
2. Create polyhedra (i.e the convex hull) out of every element in the power set and remove the elements that contain zero in their convex hulls
3. Add the remaining elements to a list L
4. For each element l in the list L, check that the points in l are not contained in any of the other elements m in L.
5. If so, add to a list of maximal sets M, if not remove from L.

For the complexity analysis, we will assume that we have n points in m dimensions. But in particular we know that for k the dimension of SL_k, $n = k^3$, and $m = 3k - 3 \approx k$. Thus, we substitute $n = m^3$ to get complexities and to check if this process is indeed more efficient for the main problem of the thesis. The complexity of the naive algorithm is determined by the containment check described in step 4 of the algorithm:

```
for v in sets_without_zero:
    contained = False
    for w in sets_without_zero:
        if v != w:
            if set(v).issubset(w):
                contained = True
                break
    if contained == False:
        maximal_sets += [v]
```

Since all the sets could possibly not contain zero, this has complexity $O(2^n 2^n) = O(2^{2n})$. But $n = m^3$, so this has complexity $O(4^{m^3})$

For the algorithm utilising hyperplanes, the complexity depends on the dimension m and the number of points n. Since any hyperplane created from the points of S is supported by m points, we see that the maximal possible number of hyperplanes we could check are $\binom{n}{m}$. Note that the point with the greatest possible computational complexity is the main loop.

```
while i < len(check_list):
    new_planes = []
    poly_points = []
    current_half = check_list[i]
    poly_to_check = Polyhedron(ieqs = [current_half])
    for y in data_points:
        if poly_to_check.contains(y):
            poly_points +=[y]
    for y in data_points:
        if not poly_to_check.contains(y):
            new_poly = Polyhedron(vertices = (poly_points + [y]))
            new_halves = new_poly.Hrepresentation()
            j = 0
            while j < len(new_halves):
```

```
                h = new_halves[j]
                if h.is_inequality():
                    check_half = Polyhedron(ieqs = [h])
                    if not check_half.contains(zero):
                        new_planes = new_planes + [h]
                else:
                    new_ineq = Polyhedron(ieqs =
                     ↪  [tuple(h.vector())]).Hrepresentation()
                    new_halves = new_halves + new_ineq
                    new_ineq = Polyhedron(ieqs = [tuple(-1 *
                     ↪  (h.vector()))]).Hrepresentation()
                    new_halves = new_halves + new_ineq
                j += 1
    if new_planes == []:
        maximal_halves = maximal_halves + [current_half]
    for h in new_planes:
        if h not in check_list:
            check_list = check_list + [h]
    i += 1
```

Since the program will search for the largest hyperplane by adding in points as the choice formula above, the maximum possible length of the hyperplane list will be a quadratic form, and so whatever executes inside the main loop will have complexity $\binom{O(n}{m)}$ times the complexity of the code within the loop. The next largest complexity term will come from the second for loop, since it contains another nested loop, and the first has two $O(1)$ complexity steps. Whatever is inside this loop is then multiplied by $O(n)$. To check containment, the program must execute a dot product of vectors, which has $O(m)$ complexity. Then, since the number of faces of a polyhedron is a linear expression in relation to the number of points, we see that the complexity of the loop is the complexity of the embedded loop times $O(n) + O(m)$. The only nonlinear execution time step inside this last loop is the check for zero being contained, which is another $O(m)$ step. Thus, our program's complexity is:

$$O(\binom{n}{m})(O(n)(O(m) + O(nm))) = O(\binom{n}{m}nm) + O(\binom{n}{m}n^2m)$$

This is $O(\binom{n}{m}n^2m)$, but this is equivalent to $O(\binom{m^3}{m}m^7)$. This complexity is not fantastic, but if we now check when $4^{m^3} > \binom{m^3}{m}m^7$ we find that this is the case for all natural m. It is important to note that this analysis works for when $n \approx m^3$, and therefore operates in this particular instance. A priori, it is possible that for large enough m, that the naive algorithm has complexity $O(2^n nm)$ from the power set and containment checks. However, since the regular expression analysis is only valid for our particular example, a general program would not contain it. This is important, since it is possible a priori that the complexity of the regular expression analysis might be greater than the complexity of the main loop. However, since it would not be included we need to check $O(2^n nm) > O(\binom{n}{m} m^7)$, and finding values such that this inequality holds is outside the scope of this thesis.

3.4 A Quick Explainer on Algebraic Curves

For this section, we will mostly follow Fulton's work in [5]. We provide a short, definition heavy exposition of these objects, since computational examples will appear in the GIT examples in the next section. We will be working over algebraically closed fields k

Put simply, algebraic (or plane) curves are the zero loci of equivalence classes of non-constant polynomials. This equivalence is defined in terms of scaling. That is, $F \sim G$ if $F = \lambda G$ for some $\lambda \in k$. These curves can be both affine or projective, but for the properties of such curves that we are concerned with, the projective case reduces to the affine, and so we will start there.

Consider an affine plane curve f in $\mathbb{A}^2$ (the zero locus of a polynomial in x_1, x_2). We have a few definitions associated to such a curve:

Definition 3.8. A point P on the curve f is called *simple* if at least one of the derivatives at P $\frac{df}{dx_i}(P)$ is nonzero. If a point is not simple it is *multiple*. A curve with only simple points is *nonsingular* or *smooth*. A curve with one or more multiple points is called *singular*.

What we are concerned with is the multiplicity (and type) of these multiple points.

Definition 3.9. Write a curve $f = f_m + f_{m+1} + \cdots + f_n$ where f_i is a polynomial of degree i, and f_m is nonzero. We define the *multiplicity* of f at P, the origin, to be m. Notably, if $m = 2$ P is called a double point, and if $m = 3$, P is called a triple point.

To determine the multiplicity of a curve f at a point P not at the origin, simply take the affine transformation that moves the point to the origin and apply it to the curve in the obvious way. Then, use the definition as described above.

Example 3.10. A simple example is the curve $x^2 + y^2 = 0$ (with our field as $\mathbb{C}$). It is 0 at $(0, 0)$ as are both its derivatives. Since the curve is made up of only one quadratic form, we see that f has multiplicity 2 at $(0, 0)$. That is, $(0, 0)$ is a double point.

We can also describe the type of multiple point. We do this using tangent lines.

Definition 3.11. For a curve f with multiplicity m at P the origin, write $F_m = \prod L_i^{r_i}$ we say that P is an *ordinary multiple point* if the r_i are all 1. Otherwise, it is a *non-ordinary multiple point*

This describes the situation with affine curves, but the examples in the next section are all projective. In the instance of a projective plane curve (or indeed for a multihomogeneous curve in m copies of $\mathbb{P}^n$), we simply dehomogenise the curve, setting one (or multiple for the multihomogeneous curve) variable to 1. We can then describe (assume we dehomogenise with respect to the last variable or variables), the multiplicity of the point $[0 : 0 : \cdots : 0 : 1]$, or
$([0 : \cdots : 0 : 1], \cdots , [0 : \cdots : 0 : 1])$.

3.5 Three Elementary Examples

Now we have a program that is capable of providing the unstable points of an action of a group on a variety, we look at three rudimentary examples to check whether the algorithm is working. These examples are simple to do by hand, and so provide a simple check to make sure there are no issues with the program. They also demonstrate that often there will be a clean geometric classifier for the set of unstable points.

3.5.1 SL_2 on 2-variable Homogeneous Polynomials

Consider $X = \mathbb{P}V$ where V is the vector space of degree 3 homogeneous polynomials in two variables. Let $G = SL_2$ act on V in the following way:

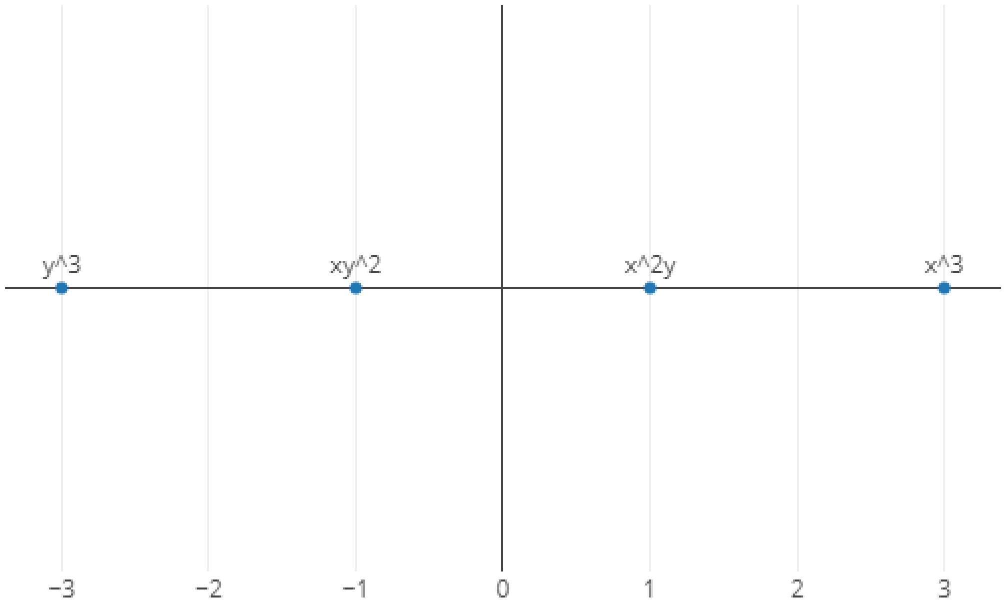
$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot f(x, y) = f(ax + by, cx + dy)$$

Theorem 3.12. *The unstable points for the action of G on X are the points which have at least a double root.*

Proof. First, note that every point is a linear combination of the following monomials: x^3, y^3, x^2y, xy^2. We take T to be the maximal torus

$$\begin{pmatrix} t & 0 \\ 0 & t^{-1} \end{pmatrix}$$

We see that the breakdown into one dimensional representations provides us with the following integers corresponding to each basis element:



Thus searching for maximal convex sets gives us that polynomials conjugate to $ax^3 + bx^2y = x^2(ax + by)$ are unstable. That is, polynomials that contain at least a double point. Note that since the action of G on the curve is $g \cdot P(X,Y) = P(g^{-1} \cdot (X,Y))$ where the action on points of $\mathbb{P}^1$ is induced from the action on polynomials and thus double points will be preserved.

For the reverse direction, consider the polynomial $P(X,Y) = (ax+by)^2(cx+dy)$. If $a = c$, $b = d$ the map $ax + by- > x$ is sufficient to map to an unstable point, and if not take the map $ax + by- > x$, $cx + dy- > y$. $\square$

Thus, we see that for two variable homogeneous cubics, the unstable points are simply the points that have a double root.

Adding an additional variable creates a slightly more subtle classification for unstable points:

3.5.2 SL_3 on 3-variable Homogeneous Polynomials

Consider $X = \mathbb{P}V$ where V is now degree 3 homogeneous polynomials in 3 variables. Consider $G = SL_3$ acting on V in the following way:

$$\begin{pmatrix} a & b & c \\ d & g & h \\ j & k & l \end{pmatrix} \cdot f(x, y, z) = f(ax + by + cz, dx + gy + hz, jx + ky + lz)$$

Theorem 3.13. *The unstable points for the action of G on X are polynomials with either a non-ordinary double point or a triple point.*

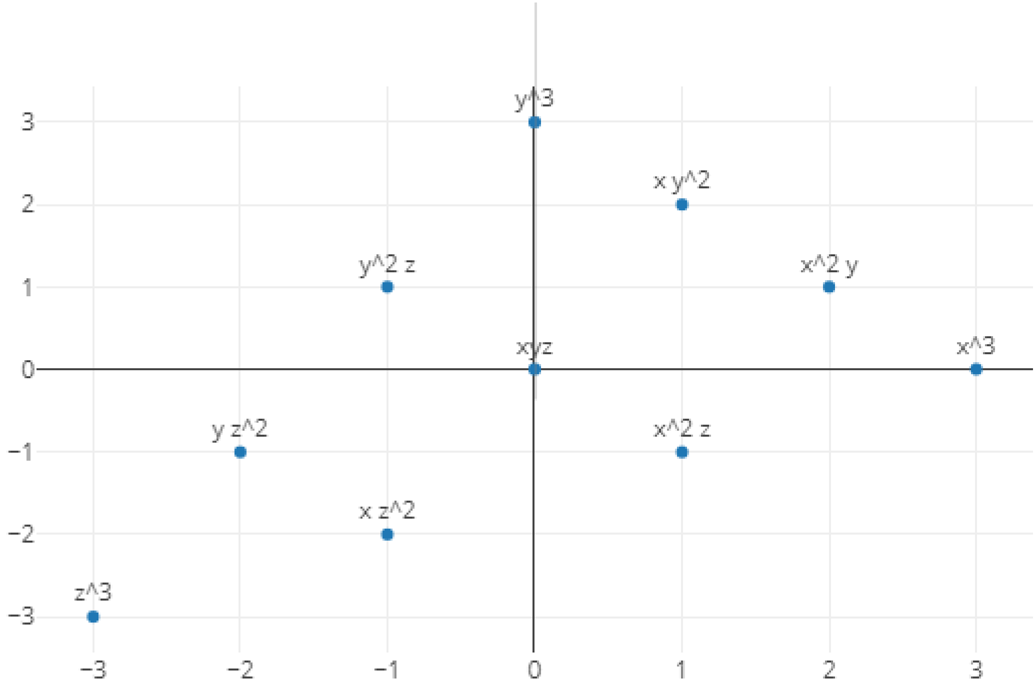
Proof. Consider the action of the maximal torus:

$$\begin{pmatrix} s & 0 & 0 \\ 0 & t & 0 \\ 0 & 0 & s^{-1}t^{-1} \end{pmatrix}$$

Again, the breakdown into one dimension representation provides us with data corresponding to each basis elements, this time 2-tuples of integers corresponding to the powers of s and t:

Using our program to compute the sets, we see that up to conjugation there are two maximal convex sets for which any linear combination of monomials within the sets (that is all monomials are taken from set 1, or all monomials are taken from set 2) will result in an unstable polynomial

- $\{x^3, x^2y, xy^2, y^3, y^2z\}$
- $\{x^3, x^2y, z^3, xz^2, x^2z\}$



Now since sending y to z gets us to every monomial in the second set from the first except for x^2y it suffices to check all possible combinations of monomials in the top set, and then check only the combination of all the monomials in the bottom set. A simple algebraic check of these combinations will demonstrate that each either has a triple point or a non-ordinary double point. For this direction it remains to show that this is conserved under the group action. This is clear however, as if the plane curve has a repeated linear factor as its degree two form at a given point, any affine transformation τ will preserve the linearity of the factor, and so there will be a non-ordinary double point and $\tau(p)$. A triple point is clearly unchanged under affine transformation.

Now consider a plane curve $P(x, y, z)$ with a triple point. Without loss of generality, we can assume that this point is at $[0 : 0 : 1]$, so we can simply make an affine transformation. Dehomogenizing with respect to z, the plane curve must look like $ax^3 + by^3 + cx^2y + dxy^2$, but this is already an unstable point.
Considering a plane curve with a non-ordinary double point, we again can assume the point is at $[0 : 0 : 1]$. Dehomogenizing with repsect to z, the plane curve must look like the triple root curve above plus a repeated root in two variables $(\alpha x + \beta y)^2$. Taking the affine transformation $\alpha x + \beta y \to y$ gives a curve

of the form $a'x^3 + b'y^3 + c'x^2y + d'xy^2 + y^2$ - which after homogenization is an unstable point. $\square$

3.5.3 SL_2^2 on Bidegree $(2, 2)$ Curves

For our last example, we will see a slightly more specific geometric criterion. Consider $G = SL_2^2$ acting on X: the projectivisation of bidegree $(2, 2)$ curves in $\mathbb{P}^1 \times \mathbb{P}^1$ in the following way:

$$(\begin{pmatrix} a & b \\ c & d \end{pmatrix}, \begin{pmatrix} g & h \\ j & k \end{pmatrix}) \cdot f((x, y), (\alpha, \beta)) = f((ax + by, cx + dy), (g\alpha + h\beta, j\alpha + k\beta))$$

Theorem 3.14. *The unstable points of the action of G on X are reducible curves $Z = F \cup g$ where F is a fiber and g is a curve such that the intersection of F and g is a single point.*

Proof. Consider the action of the maximal torus:

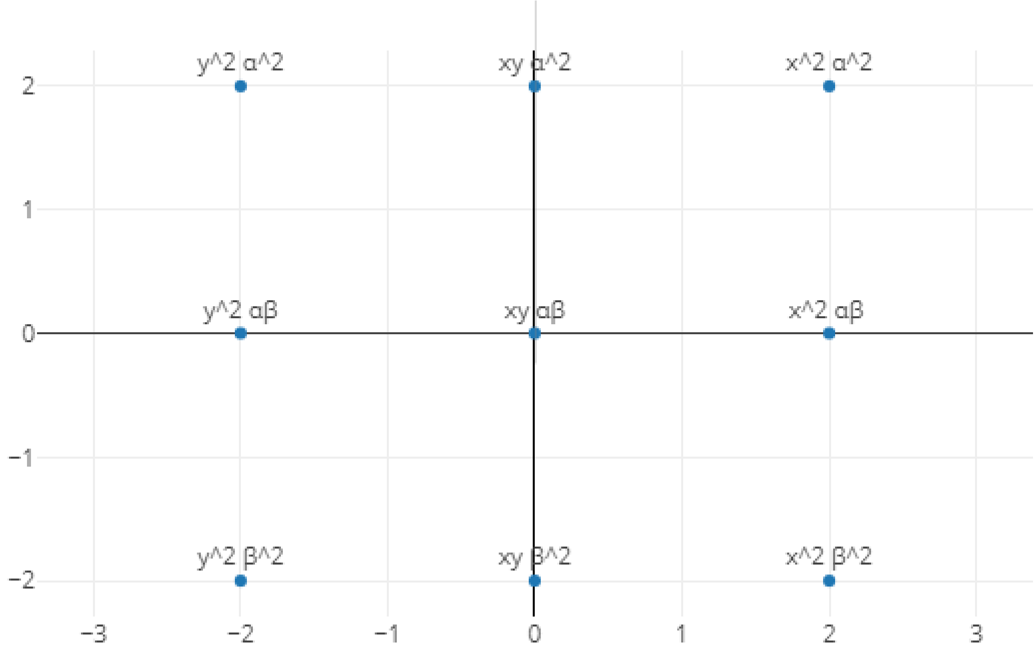
$$(\begin{pmatrix} s & 0 \\ 0 & s^{-1} \end{pmatrix}, \begin{pmatrix} t & 0 \\ 0 & t^{-1} \end{pmatrix})$$

Yet again, the breakdown into one dimensional representations gives 2-tuples corresponding to the powers of s and t. Our basis (and their corresponding 2-tuples) (with variables (x, y) and (α, β)) are:

Again, we use our program or simply plot in two dimensions to see that up to conjugation we have two maximal convex sets:

- $\{x^2\alpha^2.x^2\alpha\beta, xy\alpha^2.y^2\alpha^2\}$
- $\{x^2\alpha^2, xy\alpha^2, y^2\alpha\beta, y^2\alpha^2\}$

First, let $f = ax^2\alpha^2 + bx^2\alpha\beta + cxy\alpha^2 + dy^2\alpha^2$. Then f is divisible by α. That is, its zero locus contains the fiber $\alpha = 0$. Furthermore, $f = \alpha g$, where $g = ax^2\alpha + bx^2\beta + cxy\alpha + dy^2\alpha$. At $\alpha = 0$, $g = bx^2\beta$, and so $g = 0$ at precisely one point when $\alpha = 0$, $x = 0$. or $[0 : 1], [0 : 1]$.



Second, let $f = ax^2\alpha^2 + bxy\alpha^2 + cy^2\alpha\beta + dy^2\alpha^2$. Then $f = \alpha g$ where $g = ax^2\alpha + bxy\alpha + cy^2\beta + dy^2\alpha$. Again, setting $\alpha = 0$, $g = cy^2\beta$ and so its zero locus only has a single point of intersection, this time at $[1:0],[0:1]$.

Note that the property of having a zero fiber and one point of intersection is invariant under the action of G, since the action of the second factor only moves where the fiber is, and the action of the first factor only moves where the point of intersection is.

To begin the reverse direction, note that if we have the same curve, except swapping x with α and y with β, it is still unstable, since one can just swap the order of the two factors of SL_2^2. Thus, without loss of generality, we can assume that a curve f with a fiber has a fiber at $(a\alpha + b\beta)$, and then without further loss of generality, we can assume the fiber is at $\alpha = 0$, since we can take the curve under the affine transformation $a\alpha + b\beta \to \alpha$, and so $f = \alpha g$. Further, suppose the curve has only one point of intersection at $\alpha = 0$. This must be at $\beta = 1$, and so we must have a double point in g at $\beta = 1$. Thus, the quadratic form of g homogenised at β must look like $(ax + by)^2$, but then we simply take the affine transformation $ax + by \mapsto x$ to get a curve in the form of the first unstable point. $\square$

Chapter 4

GIT of 3-Tensors

At this point, we have rigorously developed the theory of the GIT quotient and unstable points. We have also found a method to solve the problem of finding unstable points, and tested it with some toy examples. These examples were fairly specific, and dealt with the geometry of simple curves or points in projective space. It would be useful to consider some examples for some more commonly used groups.

Matrix groups and their related cubal matrix groups (or 3-tensors, and indeed for higher dimensions also) are a fairly fundamental and simple object, and are thus deserving of some consideration. Consider the tensor

$$V_1 \otimes V_2 \otimes \cdots \otimes V_n$$

On this tensor, we have an action of $G = SL(V_1) \times SL(V_2) \times \cdots \times SL(V_n)$. In particular, if we assume that all our V_i are k-dimensional, we have an action of SL_k^n. This chapter will mostly explore the first more complicated n-tensor GIT problem, where $n = 3$. We will first address $n = 1$ and $n = 2$.

The GIT problems of the action of SL_n on 1-tensors, and SL_n^2 on the action of $M_{n\times n}$ are fairly trivial.

Example 4.1. In the case of 1-tensors, SL_n acts on an n-dimensional vector space by linear transformation. That is, for $M \in SL_n$, $v \in V$, M acts on v by $M \cdot v$. In this instance, the point $v = (1, 0, \cdots, 0)$ is destabilised by the matrix λ whose diagonal entries are t and t^{-1} followed by the remainder being 1. However, this means all points are destabilised, since any element in the orbit of v, say gv is destabilised by $g\lambda g^{-1}$. However, we can get any vector $w = (w_1, \cdots, w_n)$ from

v by acting with the matrix:

$$\begin{pmatrix} v_1 & 0 & 0 & 0 & \cdots & 0 \\ v_2 & v_1^{-1} & 0 & 0 & \cdots & 0 \\ v_3 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ v_n & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

and thus every point is unstable.

Example 4.2. For the case of $SL(V) \times SL(W)$ acting on $V \otimes W$, the solution is also fairly trivial. For $a \in V$, $b \in W$, $M \in SL(V)$, $N \in SL(W)$, $M \times N$ acts on $a \otimes b$ by $Ma \otimes Nb$, and this action is linear, so for an element $\sum a_i \otimes b_i \in V \otimes W$, $M \times N$ acts as $\sum Ma_i \otimes Nb_i$. If the dimensions of V and W are different, then every point is unstable. However, in the case the dimensions are the same, we see that the determinant is an invariant polynomial, and so all the nonsingular matrices are semistable. Indeed, the Hilbert-Mumford criterion confirms that the unstable points are the singular matrices, which suggests that the ring of invariant functions is generated by the determinant.

However, the problem gets dramatically more interesting in the case of 3-tensors, as we shall see in this chapter. Furthermore, this GIT problem has some significant ramifications for elliptic curves, since, as Bhargava and Ho [3] demonstrate, the SL_3^3-orbits of $3 \times 3 \times 3$ matrices can be used to classify elliptic curves.

Thus, the remainder of this thesis will be dedicated to studying the GIT quotient that is the action of SL_n^3 on a 3-tensor of n dimensional vector spaces over an algebraically closed field k. Since we need SL_n^3 to be linearly reductive, we must assume the characteristic of k is zero.

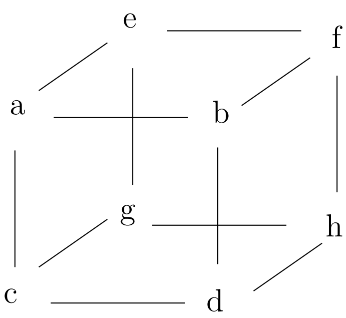
Remark 4.3. Due to the tensor-hom adjunction, we can instead view this 3-tensor of n-dimensional vector spaces as $n \times n \times n$ cubal matrices over k.

4.1 $2 \times 2 \times 2$ and $3 \times 3 \times 3$ Cubal Matrices

We will first describe the action in the simple case of $n = 2$, after which we will consider the simplest interesting case for unstable point computation: $n = 3$. Finally, using computed examples for $n = 4$ and $n = 5$, we come up with a generalised conjecture for all n.

4.1.1 $2 \times 2 \times 2$ Matrices

Let A, B, C be n-dimensional vector spaces. We can define the action of SL_n^3 on $A \otimes B \otimes C$ in the following way. Let $(M, N, K) \in SL_n^3$ and let $a \in A$, $b \in B$, $c \in C$. Then, the action of (M, N, K) on $a \otimes b \otimes c$ is $Ma \otimes Nb \otimes Kc$. Now, elements in $A \otimes B \otimes C$ are sums $\sum a_i \otimes b_i \otimes c_i$, but the action of SL_n^3 is linear, so the action on such an arbitary element is $\sum Ma_i \otimes Nb_i \otimes Kc_i$. We can describe this action in greater specificity in individual dimensions. We first provide some background from Bhargava [2] on the $2 \times 2 \times 2$ cubal matrices.
We take C_2 to be $k^2 \otimes k^2 \otimes k^2$. An example of an element in C_2 is:



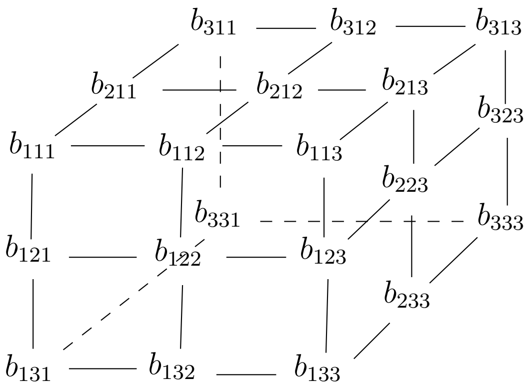
The cube can be partitioned 3 different ways into 2×2 matrices:

$$M_1 = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \qquad N_1 = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$
$$M_2 = \begin{bmatrix} a & c \\ e & g \end{bmatrix}, \qquad N_2 = \begin{bmatrix} b & d \\ f & h \end{bmatrix}$$
$$M_3 = \begin{bmatrix} a & e \\ b & f \end{bmatrix}, \qquad N_3 = \begin{bmatrix} c & g \\ d & h \end{bmatrix}$$

We restate our action of SL_2^3 in the following way: for a group element $\Gamma = (\Gamma_1, \Gamma_2, \Gamma_3)$ where $\Gamma_i = \begin{bmatrix} r & s \\ t & u \end{bmatrix}$ the ith component of the tuple, the remaining components the identity matrix, we replace (M_i, N_i) by $(rM_i + sN_i, tM_i + uN_i)$. One can check that the actions of the components Γ_i commute, and so the action is well-defined.

4.1.2 $3 \times 3 \times 3$ Matrices

In the same way we can consider the action of SL_3^3 on $3\times3\times3$ cubes ($A \otimes B \otimes C$, all 3 dimensional vector spaces over k) which we label as follows:



Again, like the $2 \times 2 \times 2$ case, the action is defined with respect to partitions on the cube. The partitions are:

1. $M_1 = (b_{1jk})$, $N_1 = (b_{2jk})$, $P_1 = (b_{3jk})$
2. $M_2 = (b_{i1k})$, $N_1 = (b_{i2k})$, $P_2 = (b_{i3k})$
3. $M_3 = (b_{ij1})$, $N_3 = (b_{ij2})$, $P_3 = (b_{ij3})$.

That is, M_1, N_1 and P_1 represent the front-to-back slicing, M_2, N_2 and P_2 represent top-to-bottom slicing, and M_3, N_3 and P_3 represent left-to-right slicing. We can describe the action in a similar way to the previous case. Let $\Gamma = (\Gamma_1, \Gamma_2, \Gamma_3)$ and let $\Gamma_i = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & j \end{bmatrix}$ be the ith component of Γ with the other two components the identity. Then Γ acts as

$$(M_i, N_i, P_i) \mapsto (aM_i + bN_i + cP_i, dM_i + eN_i + fP_i, gM_i + hN_i + jP_i)$$

A maximal torus for SL_3^3 is:

$$\left(\begin{bmatrix} s_1 & 0 & 0 \\ 0 & t_1 & 0 \\ 0 & 0 & s_1^{-1}t_1^{-1} \end{bmatrix}, \begin{bmatrix} s_2 & 0 & 0 \\ 0 & t_2 & 0 \\ 0 & 0 & s_2^{-1}t_2^{-1} \end{bmatrix}, \begin{bmatrix} s_3 & 0 & 0 \\ 0 & t_3 & 0 \\ 0 & 0 & s_3^{-1}t_3^{-1} \end{bmatrix} \right)$$

Thus, we have the group action has a decomposition into 27 one dimensional representations determined by a 27 "matrix" eigenbasis: the matrices B_{ijk}. We define the matrices B_{ijk} as the matrices where $b_{ijk} = 1$ and all else 0. Let $f : \{1,2,3\} \to \{1,0,-1\}$ be a function defined as $f(1) = 1$, $f(2) = 0$, $f(3) = -1$ and let $g : \{1,2,3\} \to \{1,0,-1\}$ be a function defined as $g(1) = 0$, $g(2) = 1$, $g(3) = -1$ Then our representations are:

$$B_{ijk} \to s_1^{f(i)} t_1^{g(i)} s_2^{f(j)} t_2^{g(j)} s_3^{f(k)} t_3^{g(k)} B_{ijk}$$

This provides us with the following 27 tuples:

$$\begin{aligned}
\{(1,0,1,0,1,0),(1,0,1,0,0,1),(1,0,1,0,-1,-1),(1,0,0,1,1,0),\\
(1,0,0,1,0,1),(1,0,0,1,-1,-1),(1,0,-1,-1,1,0),(1,0,-1,-1,0,1),\\
(1,0,-1,-1,-1,-1),(0,1,1,0,1,0),(0,1,1,0,0,1),(0,1,1,0,-1,-1),\\
(0,1,0,1,1,0),(0,1,0,1,0,1),(0,1,0,1,-1,-1),(0,1,-1,-1,1,0),\\
(0,1,-1,-1,0,1),(0,1,-1,-1,-1,-1),(-1,-1,1,0,1,0),(-1,-1,1,0,0,1),\\
(-1,-1,1,0,-1,-1),(-1,-1,0,1,1,0),(-1,-1,0,1,0,1),(-1,-1,0,1,-1,-1)\\
(-1,-1,-1,-1,1,0),(-1,-1,-1,-1,0,1),(-1,-1,-1,-1,-1,-1)\}
\end{aligned}$$

The program then provides us with the three types of unstable points. For further simplicity, we will write cubal matrices as separating the forward to back slices of the cube by lines, rather than drawing it in three dimensions. We will also represent (potentially) nonzero elements as asterisks. The three types are as follows:

$$A : \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \Bigg| \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B : \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \Bigg| \begin{bmatrix} * & 0 & 0 \\ * & 0 & 0 \\ * & 0 & 0 \end{bmatrix} \Bigg| \begin{bmatrix} * & 0 & 0 \\ * & 0 & 0 \\ * & 0 & 0 \end{bmatrix}$$

$$C : \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * \\ * & 0 & 0 \\ * & 0 & 0 \end{bmatrix} \Bigg| \begin{bmatrix} * & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Type A (up to coordinate change) is cut out by the one parameter subgroup parametrised by $(1,1,-2)$, $(0,0,0)$, $(0,0,0)$. Type B by $(2,-1,-1)$, $(0,0,0)$, $(2,-1,-1)$ and Type C by $(3,0,-3)$, $(2,-1,-1)$, $(2,-1,-1)$. That is, type A is cut out by the one parameter subgroup

$$t \mapsto \begin{bmatrix} t & 0 & 0 \\ 0 & t & 0 \\ 0 & 0 & t^{-2} \end{bmatrix} \Big| I \Big| I$$

for example.

If we recall the tensor-hom adjunction, can conclude the following linear algebraic condition for the instability of points:

A $3 \times 3 \times 3$ matrix M describes the several maps under the tensor-hom adjunction. Consider $\phi_{BC} : A^* \to B \otimes C$ and $\phi_A : B^* \otimes C^* \to A$ (the order of A B and C mutable)

Proposition 4.4. *The matrix M is unstable under the action of SL_3^3 as described above if and only if:*

1. *There exists a dimension* 1 *subspace $S \in A^*$ such that $\phi_{BC}(S) = 0$*
2. *There exists a codimension m subspace S in B^*, a codimension $k - m$ subspace T in C^*, $m \leq k \leq 1$ such that $\phi(S \otimes T)$ is at most $(3 - k - 1)$-dimensional.*
3. *There exists a codimension m, subspace S in B^*, $m \geq 1$, a codimension $k - m$ subspace T in C^*, $m \leq k \leq 2$ such that for $\overline{S} = S \otimes C^*$, $\overline{T} = B^* \otimes T$, $\phi(\overline{S} \cap \overline{T})$ is at most $(3 - k)$-dimensional and $\phi(\overline{S} \cup \overline{T})$ is at most $(n - 1)$-dimensional.*

Proof. We first check that each matrix type (A, B or C) lines up with one of the conditions. Let A have basis a_1, a_2 and a_3, B have basis b_1, b_2 and b_3 and let C have basis c_1, c_2 and c_3. Type A lines up with condition 1, since we can choose S to be the functions that send basis elements a_1 and a_2 to zero.

Type B lines up with condition 2, since we choose S to be the whole space, so $m = 0$ and we choose T to be the functions that sends the element c_1 to 0, so $k = 1$.

Type C lines up with condition 3, since we choose S to be the functions that send b_1 to zero, and T to be the functions that send c_1 to zero.

It remains to show that from the reverse direction we can get (up to changing which slicing is which, and the ordering of our basis vectors) the matrices from our three conditions.

For the first condition, let λ generate S. Pick a basis a_1, a_2, a_3 for A such that $\lambda(a_1) = 0$.

For the second condition, we either have $m = 0$, $k = 1$, or $m = 1$, $k = 1$. Either way, one of S or T is all of B or C.* Suppose, without loss of generality, that T is the codimension 1 space. Then, we simply choose a basis c_1, c_2, c_3 of C such that $T \cdot c_1 = 0$. Further, choose a basis for A such that $\phi(B^* \otimes T)$ spans a_1.

For the third condition, we have that $m = 1$ and $k = 2$. That is, S and T are both codimension 1. Choose bases b_1, b_2, b_3 of B and c_1, c_2, c_3 of C such that $S \cdot b_1 = 0$, $T \cdot c_1 = 0$ and choose a basis for A such that a_1 generates $\phi(\overline{S} \cap \overline{T})$ and a_1 and a_2 generate $\phi(\overline{S} \cup \overline{T})$. □

4.2 Higher Dimensions and a Conjecture for all Integers

For general n, we write $n \times n \times n$ matrices B as $A \otimes B \otimes C$, n-dimensional k vector spaces. We can partition these matrices up into 3 sets of n slices $(A_{11}, A_{12}, \cdots, A_{1n})$, $(A_{21}, \cdots, A_{2n})$, $(A_{31}, \cdots, A_{3n})$ in the same way by taking subspaces - fixing basis elements of the Vs, and we again get the same natural action. For (A, B, C) $A = (a_{ij})$, $B = (b_{ij})$, $C = (c_{ij})$, $A_{1j} = \sum_{i=1}^n a_{ij} A_{1i}$, $A_{2j} = \sum_{i=1}^n b_{ij} A_{2i}$, $A_{3j} = \sum_{i=1}^n c_{ij} A_{3i}$. A maximal torus for SL_n^3 are matrices whose diagonals are $(s_1, s_2, \cdots, s_{n-1}, s_1^{-1} \cdots s_{n-1}^{-1})$, $(t_1, t_2, \cdots, t_{n-1}, t_1^{-1} \cdots t_{n-1}^{-1})$, and $(v_1, v_2, \cdots, v_{n-1}, v_1^{-1} \cdots v_{n-1}^{-1})$ respectively. Thus, we have a one dimensional representation decomposition made up of n^3 elements whereby the "matrices" B_{ijk} are the matrices where $b_{ijk} = 1$ and all else 0. Much like the $3 \times 3 \times 3$ case†, our representations split into orderered triples of $n - 1$-tuples corresponding to an element's position in each of the slices - the first tuple corresponding to its position front to back in the cube, the second top to bottom, and the third left to right. No matter which choice of partition, being in the ith slice makes the tuple for that partition either a 1 at the ith component of the tuple and zero elswhere for $i \leq n - 1$, or, if $i = n$ -1 everywhere.

*The reason I've written the condition this way is to demonstrate how it generalises in higher dimensions

†You could again use functions for this but that would be fairly unsightly to write out

Doing this for $n = 4$, and running the 64 points through the program gives us 5 types of unstable points:

$$A = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \Bigg| \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}$$

$$C = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * & * \\ 0 & * & 0 & 0 \\ 0 & * & 0 & 0 \\ 0 & * & 0 & 0 \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * & * \\ 0 & * & 0 & 0 \\ 0 & * & 0 & 0 \\ 0 & * & 0 & 0 \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * & * \\ 0 & * & 0 & 0 \\ 0 & * & 0 & 0 \\ 0 & * & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * & * \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \end{bmatrix} \Bigg| \begin{bmatrix} * & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$E = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * & * \\ * & * & 0 & 0 \\ * & * & 0 & 0 \\ * & * & 0 & 0 \end{bmatrix} \Bigg| \begin{bmatrix} * & * & * & * \\ * & * & 0 & 0 \\ * & * & 0 & 0 \\ * & * & 0 & 0 \end{bmatrix} \Bigg| \begin{bmatrix} * & * & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Here, the types are cut out by the following tuples:

- A – $(1, 1, 1, -1)$, $(0, 0, 0, 0)$, $(0, 0, 0, 0)$
- B – $(2, -2, -2, 2)$, $(3, -1, -1, -1)$, $(0, 0, 0, 0)$
- C – $(3, -1, -1, -1)$, $(3, -1, -1, -1)$, $(-1, 3, -1, -1)$
- D – $(3, 3, -1, -5)$, $(3, -1, -1, -1)$, $(3, -1, -1, -1)$
- E – $(4, 0, 0, -4)$, $(3, -1, -1, -1)$, $(2, 2, -2, -2)$

We can see a pattern starting to emerge when we compare the 3 and 4-dimensional cases - there is always a type that has all but one matrix full, then types where some number of rows and columns are full, and are repeated along all not entirely full slices, and finally types where all but one of the non full slices are the same, with the one different one being the "intersection" of the rows and columns of the non-full slices. This will be confirmed in the $n = 5$ case.

For $n = 5$, we run the 125 points through the program and it produces 8 types:

$$A = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \end{bmatrix} \Big| \begin{bmatrix} * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \end{bmatrix} \Big| \begin{bmatrix} * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

$$C = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \end{bmatrix} \Big| \begin{bmatrix} * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

$$D = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

$$E = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & 0 & 0 & * \\ * & * & 0 & 0 & * \\ * & * & 0 & 0 & * \\ * & * & 0 & 0 & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & 0 & 0 & * \\ * & * & 0 & 0 & * \\ * & * & 0 & 0 & * \\ * & * & 0 & 0 & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & 0 & 0 & * \\ * & * & 0 & 0 & * \\ * & * & 0 & 0 & * \\ * & * & 0 & 0 & * \end{bmatrix} \Big| \begin{bmatrix} * & * & 0 & 0 & * \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$F = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \end{bmatrix} \Big| \begin{bmatrix} * & 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \end{bmatrix} \Big| \begin{bmatrix} * & 0 & 0 & 0 & * \\ * & 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$H = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \Big| \begin{bmatrix} * & * & * & * & * \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \end{bmatrix} \Big| \begin{bmatrix} * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The types are cut out by the following tuples:

- $A - (1,1,1,1,1)$, $(0,0,0,0,0)$, $(0,0,0,0,0)$
- $B - (3,-2,-2,-2,3)$, $(0,0,0,0,0)$, $(3,-2,-2,-2,3)$
- $C - (2,2,-3,-3,2)$, $(0,0,0,0,0)$, $(4,-1,-1,-1,-1)$
- $D - (3,-2,-2,-2,3)$, $(4,-1,-1,-1,-1)$, $(4,-1,-1,-1,-1)$

- E – $(5,0,0,0,-5)$, $(4,-1,-1,-1,-1)$, $(2,2,-3,-3,2)$
- F – $(4,4,-1,-1,-6)$, $(4,-1,-1,-1,-1)$, $(3,-2,-2,-2,3)$
- G – $(5,0,0,0,-5)$, $(3,3,-2,-2,-2)$, $(3,-2,-2,-2,3)$
- H – $(3,3,3,-2,-7)$, $(4,-1,-1,-1,-1)$, $(4,-1,-1,-1,-1)$

Thus, we again see the same types emerging, and a pattern that is being kept to as dimension increases. Indeed, we conjecture that:

Conjecture A. *For n dimensional vector spaces A, B, C, consider the matrix M. From the tensor-hom adjunction, we have (equivalently to M) a map $\phi : A^* \otimes B^* \to C$[‡]. A matrix M in $A \otimes B \otimes C$ $n \times n \times n$ dimensional is unstable with respect to the action of SL_n^3 if and only if either:*

1. *There exists a codimension 1 subspace $S \in A^*$ such that $\phi(S \otimes A^*) = 0$*
2. *There exists a codimension m subspace S in B^*, a codimension $k - m$ subspace T in C^*, $m \leq k \leq n-2$ such that $\phi(S \otimes T)$ is at most $(n-k-1)$-dimensional.*
3. *There exists a codimension m subspace S in B^*, $m \geq 1$, a codimension $k - m$ subspace T in C^*, $m \leq k \leq n-1$ such that for $\overline{S} = S \otimes C^*$, $\overline{T} = B^* \otimes T$, $\phi(\overline{S} \cap \overline{T})$ is at most $(n-k)$-dimensional and Span$\{\phi(\overline{S} \cup \overline{T})\}$ is at most $(n-1)$-dimensional.*

Since we have such a regular pattern, the if (or forward) direction is very simple:

Proof of forward direction. First note that without loss of generality we can assume that we have maximal dimension. That is, in the second type $\phi(S \otimes T)$ is $(n-k-1)$-dimensional, for example. This is because decreasing the number of nonzero entries in the matrices will have no effect on the positivity of powers of t remaining in the non-zero entries.

We prove that the types enumerated above are unstable.

Type 1 can be destabilised by $(1, 1, \cdots, 1, 1-n), (0, \cdots, 0), (0, \cdots, 0)$ by choosing a basis $(a_1, \cdots, a_n)$ for A such that S sends a_1, $a_2, \cdots, a_{n-1}$ to zero. Type 2 can be destabilised by [§]

[‡] note i j k may vary to account for swapping of matrix (in SL_n^3) order

[§] in the first tuple the first grouping lasts for $n-k-1$ many entries, in the second tuple the first grouping lasts for m many entries and in the third the first grouping lasts for $k-m$ many entries

1. $(k+1, \cdots k+1, k+1-n, \cdots, k+1-n)$
2. $(n-m, \cdots, n-m, -m, \cdots, -m)$
3. $(n+m-k, \cdots, n+m-k, m-k, \cdots, m-k)$

by choosing a basis $(b_1, \cdots, b_n)$ for B such that S sends b_1, $b_2, \cdots b_m$ to zero and a basis $(c_1, \cdots, c_n)$ for C such that T sends c_1, $c_2 \cdots c_{k-m}$ to zero and a basis for $(a_1, \cdots, a_n)$ for A such that $\phi(S \otimes T)$ spans the subspace generated by a_1, $a_2, \cdots a_{n-k-1}$. Type 3 can be destabilised by¶

1. $(k+1, \cdots, k+1, k+1-n, \cdots, k+1-n, k+1-2n)$
2. $(n-m, \cdots, n-m, -m, \cdots, -m)$
3. $(n+m-k, \cdots, n+m-k, m-k, \cdots, m-k)$

by choosing a basis $(b_1, \cdots, b_n)$ for B such that S sends b_1, $b_2, \cdots b_m$ to zero and a basis $(c_1, \cdots, c_n)$ for C such that T sends c_1, $c_2 \cdots c_{k-m}$ to zero and a basis for $(a_1, \cdots, a_n)$ for A such that $\phi(\overline{S} \cap \overline{T})$ spans the subspace generated by a_1, $a_2, \cdots a_{n-k}$ and $\phi((\overline{S} \cup \overline{T})$ spans the subspace generated by a_1, $a_2, \cdots a_{n-1}$. □

¶in the first tuple the first group lasts for $n-k$ many entries, the second group lasting for $k-1$ many entries. The other groupings last as long as in the previous footnote.

Appendix A

Complete Program

```
import csv
import time
import re

start_time = time.time()

data_points = []
with open('data_points.txt') as csv_file:
    csv_reader = csv.reader(csv_file, delimiter=',')
    for row in csv_reader:
        new_point = ()
        for x in row:
            num_x = int(x)
            new_point = new_point + (num_x,)
        data_points = data_points + [new_point]

default_points =
↪  [(3,0),(2,1),(1,2),(0,3),(-1,1),(-2,-1),(-3,-3),(-1,-2),(0,0),(1,-1)]

if data_points == []:
    data_points = default_points

min_x = 10000000
for y in data_points:
    if y[0] < min_x:
```

```
        min_x = y[0]

entry = min_x - 1
print(entry)

point_poly = Polyhedron(vertices = data_points)
center = point_poly.center()
dimension = len(data_points[0])
number_tuple = (entry,-1)
zero_tuple = (0,)*(dimension - 1)
zero = (0,)*dimension
default_half = number_tuple + zero_tuple

regex_string = '\(' + '([-]?[0-9]+), '*(dimension - 1) +
↪  '([-]?[0-9]+)\)'
tuple_regex = re.compile(regex_string)

testing = Polyhedron(ieqs = [default_half])
points_in_testing = []
for x in data_points:
    if testing.contains(x):
        points_in_testing +=[x]

print(points_in_testing)

check_list = [default_half]

i = 0
maximal_halves = []
while i < len(check_list):
    new_planes = []
    poly_points = []
    current_half = check_list[i]
    poly_to_check = Polyhedron(ieqs = [current_half])
    for y in data_points:
        if poly_to_check.contains(y):
            poly_points +=[y]
```

```
    for y in data_points:
        if not poly_to_check.contains(y):
            new_poly = Polyhedron(vertices = (poly_points + [y]))
            new_halves = new_poly.Hrepresentation()
            j = 0
            while j < len(new_halves):
                h = new_halves[j]
                if h.is_inequality():
                    check_half = Polyhedron(ieqs = [h])
                    if not check_half.contains(zero):
                        new_planes = new_planes + [h]
                else:
                    new_ineq = Polyhedron(ieqs =
                     ↪  [tuple(h.vector())]).Hrepresentation()
                    new_halves = new_halves + new_ineq
                    new_ineq = Polyhedron(ieqs = [tuple(-1 *
                     ↪  (h.vector()))]).Hrepresentation()
                    new_halves = new_halves + new_ineq
                j += 1
    if new_planes == []:
        maximal_halves = maximal_halves + [current_half]
    for h in new_planes:
        if h not in check_list:
            check_list = check_list + [h]
    i += 1

number_list = []
bad_ineqs = []
j = 0
for h in maximal_halves:
    ieq_regex = tuple_regex.search(str(h))
    ieq_tuple = ieq_regex.groups()
    ieq_list = list(ieq_tuple)

    k = len(ieq_list)
    m = k/3
    ieq_list_1 = []
```

```
ieq_list_2 = []
ieq_list_3 = []

i = 0

while i < m:
    ieq_list_1 += [ieq_list[i]]
    i+=1
while i < (2*m):
    ieq_list_2 += [ieq_list[i]]
    i+=1
while i < (3*m):
    ieq_list_3 += [ieq_list[i]]
    i+=1

i = 0
while i < len(ieq_list_1):
    ieq_list_1[i] = int(ieq_list_1[i])
    i +=1
sorted_list_1 = sorted(ieq_list_1)
print(sorted_list_1)
i = 0
while i < len(ieq_list_2):
    ieq_list_2[i] = int(ieq_list_2[i])
    i +=1

sorted_list_2 = sorted(ieq_list_2)

i = 0
while i < len(ieq_list_3):
    ieq_list_3[i] = int(ieq_list_3[i])
    i +=1
sorted_list_3 = sorted(ieq_list_3)

ieq_list = [sorted_list_1, sorted_list_2, sorted_list_3]
same_elts = False
sorted_list = sorted(ieq_list)
```

```
    for l in number_list:
        if l == sorted_list:
            same_elts = True
            bad_ineqs +=[h]
            break
    if same_elts == False:
        number_list += [sorted_list]

    j+=1

maximal_halves = [ele for ele in maximal_halves if ele not in
↪   bad_ineqs]

message = " contains points "
print(len(maximal_halves))
for h in maximal_halves:
    j = Polyhedron(ieqs =[h])
    points_in = []
    for x in data_points:
        if j.contains(x):
            points_in +=[x]
    str_h = str(h)
    str_pts = str(points_in)
    print(str_h + message + str_pts)

print("--- %s seconds ---" % (time.time() - start_time))
```

Appendix B

Naive Algorithm

```
import csv
import time
import re

start_time = time.time()

data_points = []
with open('data_points.txt') as csv_file:
    csv_reader = csv.reader(csv_file, delimiter=',')
    for row in csv_reader:
        new_point = ()
        for x in row:
            num_x = int(x)
            new_point = new_point + (num_x,)
        data_points = data_points + [new_point]

default_points = [(3,0),(2,1),(1,2),(0,3),(-1,1),(-2,-1),(-3,-3),(-1,-2),(0,0),

if data_points == []:
    data_points = default_points

vertex_sets = powerset(data_points)
sets_without_zero = []

zero = (0,)*dimension
```

```
for v in vertex_sets:
    check_poly = Polyhedron(vertices = v)
    if check_poly.contains(zero) == False:
        sets_without_zero +=[v]

maximal_sets = []

for v in sets_without_zero:
    contained = False
    for w in sets_without_zero:
        if v != w:
            if set(v).issubset(w):
                contained = True
                break
    if contained == False:
        maximal_sets += [v]

print("--- %s seconds ---" % (time.time() - start_time))
```

Appendix C

Specific Algorithm

```
import itertools
import math
import matplotlib.pyplot as plt
import time
import copy

degree_string = input("What degree: ")
#graph_string = input("Do you want to see the graphs? Answer 'y'
↪  or 'n': ")
graph_string = 'n'

#Sets up our list of homogeneous 3 tuples
degree = int(degree_string)

range_list = list(range(degree + 1))

before_list = list(itertools.product(range_list, range_list,
↪  range_list))

tuple_list = [x for x in before_list if x[0] + x[1] + x[2] ==
↪  degree]

working_list = [list(x) for x in tuple_list]
```

```
#Take our list of 3-tuples in R^3 and take the plane containing
↪ them and transform that into a copy of R^2 centered at the
↪ middle of the triangle
a = (2 * math.sqrt(3) * degree)/3

def plane_convert(list_in):
    list_in[0] = -a/(2 * degree) * list_in[0] + a/(2 * degree) *
     ↪ list_in[2]
    list_in[1] = list_in[1] - degree/3.0
    return list_in

def plane_inverse(list_in):
    list_in[0] = list_in[2] - ((2 * degree)/a) * list_in[0]
    list_in[1] = list_in[1] + degree/3.0
    return list_in

convert_list = [plane_convert(x) for x in working_list]

#Set up this triangle for graphing if wanted by getting a list of
↪ x and y coordinates.
x_list = []
y_list = []
for x in convert_list:
    x_list = x_list + [x[0]]
    y_list = y_list + [x[1]]

#Since we're only checking the first 60 degrees, the bottom right
↪ quadrant will never show up so computationally we can delete
↪ it and not have to worry. This is also why we get our list of
↪ graphed points before this step.
for x in convert_list:
    if x[0] > 0 and x[1] < 0:
        convert_list.remove(x)

x_mul = 0
```

```
#We can calculate the minimum angular distance between any two
↪ points in our triangle by looking at the bottom right and
↪ next closest outer point.
corner_point3 = plane_convert([degree, 0, 0])
near_point3 = plane_convert([degree-1 , 1, 0])
corner_point = [corner_point3[0], corner_point3[1] ]
near_point = [near_point3[0], near_point3[1]]

#long_side is the long side of the triangle, near_side is the
↪ side to the near point and between_side is the side between
↪ the two points
long_side = math.sqrt(corner_point[0]**2 + corner_point[1]**2)
near_side = math.sqrt(near_point[0]**2 + near_point[1]**2)

x_diff = corner_point[0] - near_point[0]
y_diff = corner_point[1] - near_point[1]

between_side = math.sqrt(x_diff**2 + y_diff**2)

big_value = 1/(2* long_side * near_side) * (long_side**2 +
↪ near_side**2 - between_side**2)
minimum_angle = math.acos(big_value)

extreme_x_right = (plane_convert([degree, 0, 0]))[0]
extreme_x_left = (plane_convert([0, 0, degree]))[0]

point_list = []
max_set_list = []
angle = 0

#Now, while the angle is less than 60, we check the points in the
↪ "left" halfspace generated by the line through the origin at
↪ that angle. If these points are contained in a set of points
↪ for a previous halfspace, we discard them. If not, they are
↪ added to a general list of points, the length of which tells
↪ us the number of different unstable polynomials we have.
while angle <= math.pi/3:
```

```
for x in convert_list:
    if x[1] - x_mul*x[0] > 0:
        point_list = point_list + [x]

if max_set_list == []:
    max_set_list = max_set_list + [point_list]

for x in max_set_list:
    contains = all(elem in x for elem in point_list)
    if contains == True:
        break
    rev_contain = all(elem in point_list for elem in x)
    if rev_contain == True:
        max_set_list.remove(x)

if contains == False:
    max_set_list = max_set_list + [copy.deepcopy(point_list)]


x_listp = []
y_listp = []
for x in point_list:
    x_listp = x_listp + [x[0]]
    y_listp = y_listp + [x[1]]

#Optional graphing - it's a little bit awkward at the moment
↪ since you have to close each window as it arrives.
if graph_string == "y":
    plt.plot([extreme_x_left, extreme_x_right],
    ↪ [x_mul*extreme_x_left, x_mul*extreme_x_right], '-g')
    plt.plot(x_list,y_list, 'ro')
    plt.plot(x_listp, y_listp, 'bo')
    plt.axis('equal')
    plt.grid()
    plt.show()

angle = angle + minimum_angle
```

```
    x_mul = math.tan(angle)
    point_list = []

for x in max_set_list:
    for y in x:
        y = plane_inverse(y)

print(max_set_list)
```

Bibliography

[1] D. Anderson. Linear algebraic groups: A crash course, Jan 2011.

[2] M. Bhargava. Higher composition laws i: A new view on gauss composition, and quadratic generalizations. *Annals of Mathematics*, 159(1):217–250, 2004.

[3] M. Bhargava and W. Ho. Coregular spaces and genus one curves. *Cambridge Journal of Mathematics*, 4(1):1–119, 2016.

[4] I. S. Cohen. On the structure and ideal theory of complete local rings. *Transactions of the American Mathematical Society*, 59(1):54–106, Jan 1946.

[5] W. Fulton. *Algebraic curves: an introduction to algebraic geometry*. Addison-Wesley, 1989.

[6] V. Hoskins. Moduli problems and geometric invariant theory.

[7] J. E. Humphreys. *Linear algebraic groups*. Springer, 2004.

[8] S. Mukai. *An introduction to invariants and moduli*. Cambridge University Press, 2015.

[9] D. Mumford, J. Fogarty, and F. C. Kirwan. *Geometric invariant theory*. Springer, 1994.

[10] P.-L. Yung. A proof of simplicity of sln.