Construction of Ordinary Irreducible Representations of Finite Groups

Allan Kenneth Steel

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

Pure Mathematics
University of Sydney

January 2012

Abstract

Efficient algorithms are presented for the construction of ordinary irreducible representations of a finite group. The algorithms are generic in the sense that they are applicable to any kind of group and they allow the construction in practice of many representations of very high degree and for very large groups which were not possible by previous methods. The constructed representations are always realized over a minimal-degree number field and the matrices defining the representations have very small entries in general.

A key algorithm is presented for automatic fixed-point condensation in characteristic zero which can be used to extract an irreducible representation as a constituent of a large-degree representation of the group G. Another key algorithm is presented for extending a generally reducible representation of a subgroup up to G; this involves solving a system of non-linear equations in characteristic zero via tools from Algebraic Geometry based on Gröbner bases. A new heuristic algorithm is also presented which reduces the entries of the matrices defining a representation and is very effective for high degree representations defined over a number field. Asymptotically-fast modular techniques for matrix operations over rings of characteristic zero are also exploited as much as possible.

All of the algorithms have been implemented by the author within the MAGMA Computer Algebra System and perform very effectively, as is shown by extensive tables describing constructed representations. A database has been constructed of more than 1000 absolutely irreducible ordinary representations of quasi-simple groups. The database includes representations for all entries of the Hiss/Malle classification to degree 250 and also all representations of every sporadic simple group to degree 10000 and its covers to degree 1000 at least. For the first time, minimal-degree faithful ordinary representations have been constructed for every sporadic simple group and its covers, excepting only the Monster and the double cover of the Baby Monster.

Contents

Acknowledgements

First and foremost, I thank my supervisor John Cannon for so much guidance and support over several years, and especially for providing such a wonderful environment in the Computation Algebra Group at the University of Sydney which has enabled me to develop all the algorithms of this thesis within the MAGMA Computer Algebra System.

Special thanks are also given to Claus Fieker and Bill Unger for their help and encouragement. I thank Gavin Brown, Nils Bruin, Jon Carlson, Brendan Creutz, Steve Donnelly, Markus Grassl, Derek Holt, Gabi Nebe, Bernd Souvignier, Don Taylor and Rob Wilson for helpful discussions (some going back many years) on various topics related to this thesis.

I thank Derek Holt also for providing definitions of some quasi-simple groups with standard generators which were not in the online ATLAS.

Declaration

To the best of my knowledge, this thesis contains no material previously published by any other person except where due acknowledgement has been made.

Index of Algorithms

Introduction

This thesis presents practical algorithms for the construction of irreducible ordinary representations of finite groups. Even though the theory of ordinary representations is well understood and elegant, several major practical challenges arise when attempting to construct them on a computer which do not arise when constructing modular representations. One main issue is that the representations may have to be realized over non-trivial number fields, and algorithms for basic operations with matrices over rings of characteristic zero are generally much more difficult than for matrices over finite fields, particularly when the entries of the matrices become very large. But even ignoring the issue of the time and memory needed to construct a representation, there is a very great challenge in controlling the size of the entries in the matrices defining the output representation, for the simple reason that any representation written over a field F can be conjugated by an invertible transformation to an equivalent representation, so there is a vast amount of freedom when F is a field of characteristic zero and the entries can be arbitrarily large.

The algorithms presented here overcome these major challenges in practice and enable the construction of many representations of very high degree and for very large groups which were not possible by previous methods. The representations are realized over a minimal-degree number field and the matrices defining the representations generally have small entries, even when the representation must be realized over a non-trivial number field.

Previous Work

Earlier work on the classification of representations of small degree was done by Jordan, Klein, Schur [Sch04, Sch11], Blichfeldt [Bli05, Bli07], Brauer [Bra67], Lindsey [Lin71], Huffman and Wales [HW76, HW78, Wal68, Wal69]. More recently, the primitive finite linear groups of prime degree have been classified by Dixon and Zalesskii [DZ98, DZ08].

There has been much work on constructing ordinary representations of particular classes of groups. For soluble groups, there is a basic induction/extension method, going back to Schur. Brückner [Brü98] described an algorithm based on this for computing all irreducible representations of a soluble group. Janusz [Jan66] described a method applicable to soluble groups and certain insoluble groups. Püschel [Püs02] presented an algorithm for decomposing monomial representations of soluble groups. Baum & Clausen [Bau91, BC94] presented algorithms for constructing irreducible representations of supersoluble groups. Methods for decomposing representations of nilpotent groups over infinite fields have been described by Rossmann [Ros10]. For classical linear groups of degree 2, methods to construct representations have been described by Piatetski-Shapiro [PS83] and Pergler [Per95] for $\mathrm{GL}_2(p)$, by Tanaka [Tan67] for $\mathrm{SL}_2(p)$, and by Böge [Bög93], Dixon and Gollan [DG93] and Plesken & Souvignier [PS98] for $\mathrm{PSL}_2(p)$. Szechtman [Sze99] has described methods for construction of Weil representations of unitary groups.

For a general finite group G, other methods have been proposed. One major approach is based on decomposing reducible representations via some analogy to Parker's 'Meataxe' algorithm [Par84] in characteristic zero, and has been presented in various forms by Plesken & Souvignier [PS96, Sou09], Parker [Par98], Holt [Hol98], Schulz [Sch02]; this will be discussed in detail below. Methods for extending a representation defined on a subgroup have been presented by Minkwitz [Min96], Plesken & Souvignier [PS98], Wilson [Wil99], Schulz [Sch02] and Dabbaghian-Abdoly [DA05]. Dixon [Dix93] presented a novel method which involves extracting a degree-n irreducible representation of G directly from a degree-n^2 representation of G. Dabbaghian and Dixon [DA03, DA05, DD10] described methods for a general group by reducing to the case that the group is perfect (which they could handle by some case analysis), and then using an extension method. Schulz [Sch02] described a method based on lifting modular representations. Theoretical methods have been given by Babai & Rónyai [BR90].

Methods for computing approximate complex representations have been given by Dixon [Dix70] and Babai & Friedl [BF91].

The Fundamental Goal and Strategy

The fundamental goal of the thesis is to develop efficient methods to solve the following problem: given an absolutely irreducible character χ of a finite group G, construct an ordinary representation $\rho : G \to \mathrm{GL}_n(F)$ which affords χ, where F is $\mathbb{Q}$ or a number field $\mathbb{Q}(\alpha)$ and such that:

1. The field F has minimal degree for χ (i.e., there is no number field of smaller degree over which a representation affording χ can be realized).
2. The entries of the matrices defining ρ are reasonably small.

While the minimal-degree condition on the field is of interest in itself and has applications, it has the practical advantage that for any operations done with the representation, the arithmetic of the elements of the matrices will in general be faster than otherwise, since the field degree is as small as possible. Having small entries in the matrices also means of course that subsequent operations with the representation will be faster and the space needed to store and work with such a representation will be less than otherwise. Many algorithms to construct representations use recursion (e.g., by first constructing a representation of a subgroup) and so the field degree and the size of the entries will grow with each new level of recursion unless it is controlled in some way.

If a desired representation can only be realized over a non-trivial number field, then constructing a suitable representation with small entries can be a huge challenge. Most of the existing methods referred to in the previous section do not attempt to write their results over a field of minimal degree and they do not control the size of the entries in the result. In particular, it is easy to list several examples with degree less than 100 where the existing methods fail to produce representations written over minimal fields with reasonably small entries (e.g., the representation 35a of Sz(8) over a degree-3 number field, or the representation 85a of J_3 over a quadratic field).

The thesis is structured around three major approaches for the construction of an ordinary representation $\rho : G \to \mathrm{GL}_n(F)$ which affords a given absolutely irreducible character χ of a finite group G, for a minimal field F:

1. The **splitting** approach: ρ is extracted as an irreducible constituent of some representation σ of G. Usually σ will be relatively easy to construct, typically arising from a permutation representation of G, the induction to G of a representation of some subgroup of H or the tensor product of existing representations of G.
2. The **extension** approach: ρ is extended from a representation $\rho_H : H \to \mathrm{GL}_n(F)$ which affords the restricted character $\chi \downarrow_H$ (for some proper subgroup H of G) so that $\rho \downarrow_H = \rho_H$.
3. The **hybrid** approach: this combines aspects of both the splitting and extension approaches in the one algorithm.

The following sections outline these approaches.

The Splitting Approach

The key operation in the splitting approach is the extraction of an irreducible constituent ρ affording χ from a representation σ of G whose degree is often much larger than that of χ. To do this efficiently we construct an absolutely irreducible representation ρ over a minimal field F by first constructing an irreducible rational representation $\rho_{\mathbb{Q}}$ and then extracting $\rho : G \to \mathrm{GL}_n(F)$ as a constituent of $\rho_{\mathbb{Q}}$, where F is derived from the endomorphism ring of $\rho_{\mathbb{Q}}$. The bulk of the effort in this approach is spent on constructing irreducible rational representations.

We thus focus first on constructing irreducible rational representations by the splitting approach. Now for splitting modular representations, there are very effective methods: the basic computational tool is Parker's 'Meataxe' algorithm [Par84], which was later improved by Holt & Rees [HR94]. In the attempt to extend the Meataxe algorithm to characteristic zero, there are major difficulties, particularly because the Schur index of an irreducible ordinary representation may be non-trivial; in such a case, the endomorphism ring of the representation is a noncommutative division ring. These difficulties have been well-known for some time and various techniques to overcome these were proposed by Plesken & Souvignier [PS96], Holt [Hol98] and Parker [Par98] in the mid 1990s.

The first major challenge is to determine whether a homogeneous rational representation is irreducible or not. Plesken & Souvignier [PS96] presented methods for solving this problem based on analyzing the structure of the endomorphism ring; they presented heuristics for non-trivial cases based on solving norm equations which can be applied in many but not all cases. Determining the structure of a homogeneous rational representation can now be achieved by an algorithm of Unger [Ung09] to compute the Schur index of a given absolutely irreducible character or by an algorithm by Nebe and the present author [NS09a] which computes a maximal order of a central simple algebra and recognizes the associated Schur index and multiplicity. For explicitly splitting reducible homogeneous rational representations, Souvignier [Sou09] recently suggested searching for singular elements in a reduced basis of a maximal order of the endomorphism ring, based on the algorithm in [NS09a]. We present a variant of this method, but also present alternative methods based on finding a rational point on a conic and using Fieker's algorithm [Fie09] for rewriting a representation over a field of minimal degree.

The second major challenge with a rational Meataxe is that as the degree grows, the entry growth in the matrices can make the computations very expensive, and the resulting representations may have very large entries and so be unusable. Plesken & Souvignier [PS96] and Parker [Par98] proposed that when computing with $\mathbb{Q}G$-modules, one should always work with saturated lattices ($\mathbb{Z}$-modules) with bases which are reduced by the LLL algorithm [LLL82]. We give a detailed description of efficient algorithms and approaches for performing the relevant computations with integer matrices. Combining this with the tools for homogeneous representations above, we present a complete 'rational Meataxe' to decompose a semisimple A-module, where A is a finite-dimensional algebra over $\mathbb{Q}$.

Hitherto, the rational Meataxe has mostly been applied directly to group representations when attempting to constructing an irreducible rational representation. This approach is greatly limited as the degree grows, since computing the endomorphism ring or the minimal polynomial of a group algebra element becomes very expensive as the degree approaches 1000. We present a new automatic algorithm to extract an irreducible rational representation from a larger-degree representation σ by using fixed point condensation over $\mathbb{Q}$. The major advantage of this approach is that the rational Meataxe algorithm need only be applied to a condensed module $\tilde{M}$, which is derived from σ and a suitable condensation subgroup K of G and whose dimension is typically much smaller than the degree of σ, so this avoids the above limitations of the rational Meataxe in high degree. The original examples of condensation go back to Parker and Thackray in 1979 [Tha81] and were used to construct modular representations, but condensation has apparently been hardly used hitherto to construct representations in characteristic zero. Nickerson [Nic06] gave an algorithm for decomposing permutation representations over a field of characteristic zero, which effectively uses a special case of fixed-point condensation where the condensation subgroup is always chosen to be a point stabilizer. The key component of our automatic algorithm is a search to find a suitable condensation subgroup K so that the dimension of the condensed module $\tilde{M}$ is minimized but also so that the relevant information to construct the irreducible constituent may be discovered. We also present an algorithm which automatically searches for a suitable 'virtual' rational representation σ to which the automatic condensation algorithm can be applied to extract the desired irreducible constituent. The search considers permutation, induced and tensor product representations.

Previous work which uses a characteristic zero Meataxe approach has been mostly focused on computing irreducible rational representations. One can move from an irreducible rational representation $\rho_{\mathbb{Q}}$ to an absolutely irreducible representation ρ over a suitable minimal field F in polynomial time by computing the action on an eigenspace over F of a suitable endomorphism of $\rho_{\mathbb{Q}}$, but it it is often very difficult to control the size of the entries in the result. We present a heuristic LLL-based algorithm which attempts to select a basis of the eigenspace over F so that the final representation has small entries. Many absolutely irreducible irrational representations with very small entries can be constructed by this algorithm. However, the success of the method depends very strongly on finding a reasonably sparse endomorphism of the rational representation $\rho_{\mathbb{Q}}$: as the degree of the representation increases (typically above 100), this algorithm becomes quite slow and often fails to find a representation with small entries. Algorithms presented later overcome these problems.

Based on the splitting approach, we also present an algorithm to construct irreducible F-representations, where F is any number field which is normal over $\mathbb{Q}$.

The key advantages of the condensation-based splitting approach are that it does not place any conditions on G or χ and allows the construction of irreducible rational representations of rather high degree (up to 1000) with small integral entries in reasonable time and it allows the construction of absolutely irreducible representations over non-trivial number fields, often with small entries. The major limitations of the approach are that it is not applicable in practice when G has no proper subgroups of moderate index, and it will often fail to construct an irrational representation with reasonably small entries.

The Extension Approach

Let χ be an absolutely irreducible character of a finite group G. Suppose that H is a proper subgroup of G and $\rho_H : H \longrightarrow \mathrm{GL}_n(F)$ affords the restricted character $\chi_H = \chi \downarrow_H$. Then one can attempt to extend ρ_H to a representation $\rho : G \longrightarrow \mathrm{GL}_n(F)$ affording χ, such that $\rho \downarrow_H = \rho_H$. It is easy to see that the set of all such extensions forms an orbit under the action of the centralizer of ρ_H in $\mathrm{GL}_n(F)$. If ρ_H is absolutely irreducible, then the centralizer is trivial, so the extension ρ is unique; we call this case 'irreducible extension'.

For an arbitrary finite group G, Minkwitz [Min96] gave an algorithm for irreducible extension which involves looping over H, so this algorithm is obviously only practical when H is relatively small. Plesken & Souvignier [PS98, 3.1] and Dabbaghian-Abdoly [DA05] described algorithms based on linear algebra which involve evaluating ρ_H at $O(n^2)$ elements of H and solving a linear system over F of rank n^2 where n is the degree of the character χ, so this approach becomes very expensive as n grows. Wilson [Wil99] suggested that in extension algorithms one could use an amalgam of H and a normalizer of some subgroup of H and Unger [Ung10] noted that this idea can be directly applied to the linear algebra-based irreducible extension algorithm of Dabbaghian-Abdoly so that the rank of the linear system to be solved can usually be reduced dramatically. We describe how this variant can be implemented efficiently.

The major limitation of the irreducible extension algorithm is that it is very often the case that there is no subgroup H of G such that $\chi \downarrow_H$ is absolutely irreducible, so the algorithm simply cannot be used. Instead, one can attempt to do 'general extension' from ρ_H to G, where ρ_H is not assumed to be absolutely irreducible. Schulz described a generalization of Minkwitz's irreducible extension algorithm, for the case that the multiplicity of each absolutely irreducible constituent of ρ_H is 1 [Sch02, 2.2]; since this involves looping over H, the algorithm is again limited to the case that H is rather small. An alternative approach is to set up a symbolic matrix X with entries in a suitable polynomial ring $F[x_1, \ldots, x_k]$, so that X represents the image of some $g \in G \setminus H$ in the proposed extension ρ of ρ_H; one can then attempt to gather polynomial relations on $x_1, \ldots, x_k$ corresponding to suitable relations in the group involving g and elements of H, and then solve the associated system. There has hitherto been no practical algorithm presented for general extension in characteristic zero based on this approach which can handle non-trivial cases. Wilson [Wil99] outlined the basic method and gave some simple examples, but with no general algorithm for characteristic zero (the focus for larger examples was on modular representations). Plesken & Souvignier [PS97] described a similar method with some basic improvements which is only suitable in practice for groups defined by short presentations and representations of small degree.

We present a practical heuristic algorithm for general extension which is effective for an arbitrary finite group G and absolutely irreducible character χ. Instead of using polynomial relations derived from a complete presentation of G (for which the polynomial system would be impossible to manage in non-trivial examples), we show how one can construct a suitable polynomial system from a small set of group relations based on elements of G of small order. The termination of the algorithm depends on a precise criterion which we develop by using concepts from Algebraic Geometry and Gröbner bases. We also describe several techniques by which the polynomial system can be reduced as the algorithm proceeds, so that group relations of relatively high length can often be handled. Practical heuristics are also described so that the final representation can generally be written over a minimal field.

One major advantage of the general extension algorithm is that it can easily handle the situation where G has no proper subgroups of moderate index, and does not require any specific conditions for G or χ, so the algorithm can be applied recursively. It also typically yields a result with very small entries, even when the result is written over an irrational field and the degree is large. Using this algorithm, we have been able for the first time to compute many ordinary representations of the very large sporadic groups which do not have maximal subgroups of moderate index.

The Hybrid Approach

Suppose that G is a finite group and $\rho_1 : G \to \mathrm{GL}_n(F)$ is a representation of G, where F is $\mathbb{Q}$ or a number field, and such that the image matrices of ρ_1 have large entries. A very challenging problem is to compute an equivalent representation ρ over F which has smaller entries than ρ_1. There is a well-known algorithm [PS96, Sou09, Sch02] to reduce the entries of a rational or integral representation, which works via LLL-reduction of a positive definite form fixed by the representation. The major limitation is that above degree 100, this method loses its effectiveness (and becomes very slow) and there does not seem to be any practical analogy for representations over number fields.

We present a new heuristic algorithm for reducing the entries of a given ordinary representation ρ_1, whose character is χ. The basic idea is to conjugate ρ_1 to a representation ρ which is an extension of ρ_H, where ρ_H affords $\chi \downarrow_H$ for some subgroup H of G. The algorithm can be considered in a sense to be the reverse of the general extension algorithm, combined with a heuristic LLL-based reduction. The algorithm is very effective for reducing a representation even when it has high degree and is defined over a non-trivial number field.

Finally, we present a hybrid algorithm to construct an absolutely irreducible representation of a given character χ which combines aspects of both the splitting and extension approaches. Using the condensation-based splitting approach, it first sets up information determining an absolutely irreducible representation ρ_1 which affords χ and is written over a minimal field F, though ρ_1 is not constructed explicitly (often it will have very large entries and would take a very long time to construct). Then the algorithm uses the above entry reduction algorithm and modular techniques to conjugate ρ_1 directly to a reduced representation ρ which is the extension of some representation $\rho_H : H \longrightarrow \mathrm{GL}_n(F)$ which affords $\chi \downarrow_H$ (for a subgroup H of G).

The great advantage of the hybrid algorithm is that it always produces representations over a minimal field and generally with very small entries, even over non-trivial number fields (of a similar or better quality to those returned by the general extension algorithm), while it is often much more efficient than the general extension algorithm when the polynomial system arising in that algorithm is very large or is difficult to solve over the minimal field F. Using this algorithm, we have been able to construct the degree-10944 irreducible rational representation of the O'Nan sporadic group for the first time.

The Implementation and Database of Representations

Prior computational programs to construct representations have been developed by Flodmark and Blokker [FB67], Brott and Neubüser [BN70], Gollan and Grabmeier [GG90], Brückner [Brü98], and Dabbaghian [DA03, Dab08].

All of the algorithms in this thesis have been implemented by the author within the MAGMA Computer Algebra System [BCP97, CP96] (several of the fundamental algorithms described in Chapters 1 and 2 have been implemented by the author within the C kernel of MAGMA). A first version of the rational Meataxe and algorithms for construction of irreducible rational representations via condensation were released in MAGMA 2.16 in November 2009 and it is planned that the other algorithms will be released within MAGMA in the future. Note that all timings are for a 2.8GHz Intel Xeon64 (with 128GB memory, though much less than that was used for most computations).

The final goal of this thesis is to apply the algorithms to build a database of ordinary representations of interest. There has been much previous work to construct such databases. The online ATLAS of finite group representations of Wilson et al. [WWT$^+$] contains very many permutation and modular matrix representations of almost simple groups. There are also ordinary representations for many of the groups, but there are many gaps at the time of writing. For several important groups, an irreducible rational representation is present in the database, but not a minimal-degree faithful absolutely irreducible representation, presumably because it has been hitherto very difficult to compute such representations with reasonably small entries (e.g., degree 56 for J_1 and degree 85 for J_3 are missing).

Of particular interest are representations of quasi-simple groups. Hiss & Malle have given a classification of all faithful irreducible representations of quasi-simple groups to degree 250 [HM01, HM02]. Nickerson [Nic06] constructed many ordinary representations from this classification, but there are many absolutely irreducible representations which he could not construct (see Appendix A of that thesis). Holt has also constructed a partial database of representations of quasi-simple groups within MAGMA matching this classification. Using our algorithms, we have constructed a complete database of the 669 absolutely irreducible ordinary representations in the main classification and we present a table describing these representations which matches the main table of Hiss & Malle. We have also constructed representations of $\mathrm{L}_2(q)$ and $2.\mathrm{L}_2(q)$ for $q < 100$. Every representation in our database is written over a field of minimal degree and generally has small entries.

The sporadic simple groups are of special interest. Wilson [Wil98a] noted that it was desirable to have ordinary representations of the sporadic simple groups and these have been missing hitherto for several of the groups. For some of these groups, a minimal-degree faithful representation has degree above 1000, and previous methods have been inadequate to construct these. But we have been able to construct such representations for the first time for all such groups, excluding only the Monster group. To summarize the chief results, we have succeeding in constructing the following faithful absolutely irreducible ordinary representations:

- The minimal-degree representation of every sporadic group and its covers except for the Monster group (degree 196883) and the double cover 2.B of the Baby Monster (degree 96256).
- All representations of every sporadic group to degree 10000 at least.
- All representations of every cover of every sporadic group to degree 1000 at least.
- All representations of every Mathieu group and its covers.

The database will be released within MAGMA in the near future. The webpage [Ste11] contains several of the representations (including all those representations of moderate degree which are mentioned in the examples of this thesis).

Outline of the Thesis

We now give a brief overview of the thesis.

In Part I, we present algorithms to construct irreducible ordinary representations.

- In Chapter 1, we present basic results from the theory of Group Representations and outline fundamental efficient algorithms for fast linear algebra over the rings of characteristic zero which we will encounter.

- In Chapter 2, we describe a 'rational Meataxe' which decomposes a semisimple A-module, where A is a finite-dimensional algebra over $\mathbb{Q}$. A special variant of the algorithm extracts only the desired constituents matching some given trace information.

- In Chapter 3, we describe the splitting approach for constructing irreducible representations. We show how condensation can be used automatically in characteristic zero to decompose permutation, induced or tensor representations efficiently. Based on this, we present a generic algorithm to construct irreducible rational representations via condensation. This immediately leads to algorithms to construct absolutely irreducible representations over minimal fields and irreducible representations over a general number field which is normal over $\mathbb{Q}$. An algorithm is also presented to rewrite a given absolutely irreducible representation over a minimal field.

- In Chapter 4, we consider irreducible extension, where an absolutely irreducible representation ρ_H of a subgroup H is extended to a representation of G. We show how to make a linear algebra-based algorithm efficient and develop important techniques to be used in the general extension algorithm.

- In Chapter 5, we present our general extension algorithm, where an arbitrary representation ρ_H of a subgroup H is extended to a representation of G. This algorithm is particularly effective when G does not have any maximal subgroups of reasonably small index and we describe in detail how we have been able to construct some very high-degree ordinary representations of the sporadic simple groups.

- In Chapter 6, we first present the new heuristic algorithm for reducing the entries of a given representation ρ of G. We next introduce the concept of a 'black-box' representation, which encodes a fixed representation over a number field which has potentially huge entries but to which one can apply modular techniques efficiently. Combining this with the reduction technique yields the hybrid algorithm for constructing an irreducible representation of any group G such that the representation has very small entries in general, even when the degree of the representation is large.
- In Chapter 7, we outline a basic strategy for computing a representation affording a given character χ, using all the algorithms presented in the thesis.

In Part II, we describe our database of ordinary representations which have been constructed by the algorithms of the thesis and are all realized over a minimal field. This is presented by a series of tables which lists information for each constructed representation.

- In Chapter 8, we first give a description of the format of the tables (principally on how to read the detailed information which describes the methods used).
- In Chapter 9, we give tables describing the many representations of quasi-simple groups which we have constructed. We first give a table up to degree 250, exactly matching the main table in the classification of Hiss & Malle [HM02]. We then give a table listing higher-degree representations of quasi-simple groups; this includes several of the minimal-degree faithful representations of the sporadic groups.
- In Chapter 10, we describe representations of $\mathrm{L}_2(q)$ and $2.\mathrm{L}_2(q)$ for $q < 100$.
- In Chapter 11, we describe representations of some other types of groups.

Part 1

Constructing Irreducible Representations

Chapter 1

Representation Theory and Basic Tools

1.1. Introduction

In this chapter, we present basic results in the theory of group representations and fundamental tools for linear algebra in characteristic zero which we will need.

Throughout the thesis, all groups are finite and all algebras and modules over a field F are finite-dimensional. Also, fields will in general be either the rational field $\mathbb{Q}$ or a number field $\mathbb{Q}(\alpha)$ (the only exceptions will be finite fields used in modular algorithms, which will be noted).

For the presentation of representation theory, we generally follow Isaacs [Isa06] and Huppert [Hup98], so we refer the reader to those standard references. We assume that the following basic concepts are familiar (see appropriate references in [Isa06]): algebras and modules [Chap 1], Schur's lemma [1.5], Maschke's Theorem [1.9], representations [2.1], characters [2.2], similarity [2.9], irreducible characters and the character table [p. 15–17]. We also use the following notation and conventions throughout the thesis:

1. $\mathcal{M}_n(R)$ denotes the ring of $n \times n$ matrices over the ring R and $\mathcal{M}_{m\times n}(R)$ denotes the R-module of $m \times n$ matrices over the ring R.
2. For a representation $\rho : G \longrightarrow \mathrm{GL}_n(F)$ of a group G, there is a corresponding FG-module M, where $v \cdot a := vR(a)$ for $v \in M, a \in FG$. Conversely, if M is an FG-module of dimension n with a fixed basis B for the underlying vector space F^n, then for $a \in FG$, we have a map $a_M : M \longrightarrow M$ given by $v \mapsto va$ and there is a corresponding representation $\rho : G \longrightarrow \mathrm{GL}_n(F)$ such that $\rho(g)$ for $g \in G$ is defined to be the matrix of $1.g \in FG$ with respect to B.
3. For a representation $\rho : G \longrightarrow \mathrm{GL}_n(F)$ and an extension field E of F, let $\rho^E : G \longrightarrow \mathrm{GL}_n(E)$ denote the extension of ρ (via extension of scalars from F to E). Similarly, for an A-module M, where A is an F-algebra, let M^E denote corresponding A_E-module, where A_E is the extension of A to E.
4. If χ is the character of some representation $\rho : G \longrightarrow \mathrm{GL}_n(F)$, then we say that ρ **affords** χ, and we say that a character χ can be **realized** over a field F if there exists some representation $\rho : G \longrightarrow \mathrm{GL}_n(F)$ which affords χ.
5. Suppose that χ is an E-character (a character whose values lie in a field E) and F is a subfield of E. Then $F(\chi)$ denotes the subfield of E generated by F and the character values of χ. Also, $\mathbb{Q}(\chi)$ is called the **character field** of χ. Note that $F(\chi)$ is always a finite degree Galois extension of F and the Galois group $\mathrm{Gal}(F(\chi)/F)$ is abelian [Isa06, p. 152].
6. Let A be an F-algebra and let M_1, M_2 be A-modules of dimensions d_1, d_2 respectively. Let $H = \mathrm{Hom}_A(M_1, M_2)$. Then relative to standard bases of M_1, M_2, elements of

H may be identified with elements of $\mathcal{M}_{d_1\times d_2}(F)$ and H may be identified with a subspace of the F-vector space $\mathcal{M}_{d_1\times d_2}(F)$. Similarly, $\mathrm{End}_A(M_1)$ can be identified with a subalgebra of the matrix algebra $\mathcal{M}_{d_1}(F)$. We can do the same with representations $\rho_1 : G \to \mathrm{GL}_{d_1}(F)$ and $\rho_2 : G \to \mathrm{GL}_{d_2}(F)$, identifying $\mathrm{Hom}_{FG}(\rho_1, \rho_2)$ with a subspace of the F-vector space $\mathcal{M}_{d_1\times d_2}(F)$, and identifying $\mathrm{End}_{FG}(\rho_1)$ with a subalgebra of the matrix algebra $\mathcal{M}_{d_1}(F)$.

1.2. Splitting Fields and the Schur Index

Let G be a finite group.

Definition 1.2.1. *Define* $\mathrm{Irr}(G)$ *to be the set of all absolutely irreducible* $\mathbb{C}$*-characters of* G *(the characters afforded by absolutely irreducible representations). A field* E *is called a* **splitting field** *for* G *if every irreducible* E*-representation of* G *is absolutely irreducible. Following* [Isa06, p. 149], *if* E *is a splitting field for* G *we let* $\mathrm{Irr}_E(G)$ *denote the set of characters of the (absolutely) irreducible* E*-representations of* G*.*

Suppose E is a splitting field for G and let F be a subfield of E. If $\chi, \psi \in \mathrm{Irr}_E(G)$, we say that χ and ψ are **Galois conjugate over** F if $F(\chi) = F(\psi)$ and there exists $\tau \in \mathrm{Gal}(F(\chi)/F)$ such that $\chi^\tau = \psi$. This clearly defines an equivalence relation on $\mathrm{Irr}_E(G)$, and the size of the class is $|F(\chi) : F|$ [Isa06, 9.17]. For a character χ of G, let $\mathrm{GalSum_{E/F}}(\chi)$ denote the sum of the orbit of χ under the Galois group $\mathrm{Gal}(E/F)$, where F is a subfield of E and it is assumed that $E(\chi) = E$. Also, we will let $\mathrm{GalSum_F}(\chi)$ denote $\mathrm{GalSum}_{F(\chi)/F}(\chi)$. Clearly the character values of $\mathrm{GalSum_{E/F}}(\chi)$ and $\mathrm{GalSum_F}(\chi)$ all lie in F.

Definition 1.2.2. [Isa06, 10.1] *Suppose* F *is a subfield of* E*, where* E *is a splitting field for* G *and* $\chi \in \mathrm{Irr}(G)$*. Choose an irreducible* E*-representation* ρ_E *which affords* χ *and an irreducible* F*-representation* ρ_F *such that* ρ_E *is a constituent of* $(\rho_F)^E$*. Then the multiplicity of* ρ_E *as a constituent of* $(\rho_F)^E$ *is called the* **Schur index** *of* χ *over* F *and is denoted by* $s_F(\chi)$*.*

Theorem 1.2.3. [Isa06, 10.2, 10.17] *Suppose* $\chi \in \mathrm{Irr}(G)$ *and* F *is a subfield of* $\mathbb{C}$*. Then:*

1. $s_{F(\chi)}(\chi) = s_F(\chi)$.
2. *Let* C *be the Galois conjugacy class of* χ *over* F*. Then* $s_F(\chi)(\sum C)$ *is the character of an irreducible* F*-representation of* G*.*
3. *Suppose* $F(\chi) = F$*. Then there exists an extension field* E *of* F *such that* χ *is afforded by an* E*-representation and* $|E : F| = s_F(\chi)$*.*

Remarks 1.2.4. Isaacs uses $m_F(\chi)$ (or m) while Huppert uses $s_F(\chi)$ (or s) for the Schur index. We use the latter because we wish to use m in general for the multiplicity of a representation (which may have a non-trivial Schur index) as a constituent of some other representation (not necessarily irreducible over some field).

Definition 1.2.5. *Let* $\chi \in \mathrm{Irr}(G)$ *and* $F \subset \mathbb{C}$ *be a field. Call an extension field* E *of* F *a* **minimal extension of** F **for** χ *if* χ *can be realized over* E *and* $\mathrm{Deg}_F(E)$ *is minimal under such a condition. Also, call any field* $F \subset \mathbb{C}$ *a* **minimal field for** χ *if* F *is a minimal extension of* $\mathbb{Q}$ *for* χ*; by the definition of the Schur index and Thm. 1.2.3, it is clear that a minimal field* F *for* χ *must be a degree-*s *extension field of* $\mathbb{Q}(\chi)$*, where* $s = s_{\mathbb{Q}}(\chi)$*.*

1.3. Irreducible F-representations

We will often need to work with representations which are irreducible over a field F but not necessarily absolutely irreducible. This suggests the following definition.

Definition 1.3.1. *Let G be a finite group and F a field. Define $\mathrm{Irr}_F(G)$ to be the set of characters of all irreducible F-representations of G.*

Remarks 1.3.2. Note that if E is a splitting field for G, then $\mathrm{Irr}_E(G)$ according to this definition coincides with the definition of $\mathrm{Irr}_E(G)$ in Def. 1.2.1. Isaacs [Isa06] uses the notation $\mathrm{Irr}_E(G)$ only for the case that E is a splitting field, but in this thesis F will be allowed to be any field. By [Isa06, 9.22], the characters in $\mathrm{Irr}_F(G)$ are non-zero, distinct and linearly independent over F, and given an arbitrary F-representation ρ, ρ can be decomposed into irreducible F-representations, so that the character of ρ equals the corresponding combination of the characters of the irreducible modules in the decomposition.

Theorem 1.3.3. [Isa06, 9.21] *Let F be a subfield of E, where E is a splitting field for G. Let ρ be an irreducible F-representation of G. Then*

1. *The irreducible constituents of ρ^E all occur with equal multiplicity s.*
2. *The characters $\chi_i \in \mathrm{Irr}_E(G)$ afforded by the irreducible constituents of ρ^E constitute a Galois conjugacy class over F and so the fields $F(\chi_i)$ are all equal.*

Theorem 1.3.4. *Let F be a field and let E be a splitting field for G containing F. Partition $\mathrm{Irr}_E(G)$ (the absolutely irreducible characters of G) into $\mathrm{Gal}(E/F)$-classes $\{\mathcal{C}_1, \ldots, \mathcal{C}_r\}$. For $i = 1, \ldots, r$, let s_i be the common Schur index over F of the characters in $\mathcal{C}_i$ and let χ_i be s_i times the sum of the characters in $\mathcal{C}_i$. Then $\mathrm{Irr}_F(G) = \{\chi_1, \ldots, \chi_r\}$. Also, the χ_i do not depend on the choice of E, so this procedure gives a simple algorithm for computing $\mathrm{Irr}_F(G)$ from the character table of G and the s_i values.*

Proof. By Thm. 1.2.3, each χ_i is the character of an irreducible F-representation of G, so is in $\mathrm{Irr}_F(G)$. Conversely, if $\chi \in \mathrm{Irr}_F(G)$, then there is an irreducible F-representation affording χ and this must equal s_i times the sum of the characters in $\mathcal{C}_i$ for some i, by Thm. 1.3.3 (1), (2) and Def. 1.2.2. The last statement follows from [Isa06, 9.13]. □

Most algorithms in this thesis assume that one can first compute the character table of G. We use W. Unger's algorithm [Ung06], which has been implemented by him in MAGMA (function `CharacterTable`) and is very efficient: it typically takes only a small number of seconds for most groups of order up to about 10^{10} when there is a moderate number of conjugacy classes. Further, the algorithm can frequently handle groups of much larger orders within reasonable time (e.g., the character table of Fi_{22}, of order $\sim 6.5 \times 10^{13}$, is computed in about 7 seconds). We will thus use this algorithm extensively for moderately-sized groups but we will also present a method later to compute representations without needing to compute the character table of G explicitly. Unger has also developed an algorithm to compute the Schur index $s_{\mathbb{Q}}(\chi)$ of χ for a given $\chi \in \mathrm{Irr}(G)$ [Ung09]. This algorithm has also been implemented by him in MAGMA (function `SchurIndex`) and usually takes less than a second for a given character.

Based on these two algorithms, we can easily compute $\mathrm{Irr}_F(G)$, using the simple method described in Thm. 1.3.4. In particular, we frequently compute $\mathrm{Irr}_{\mathbb{Q}}(G)$ by this method; the characters thus computed all have rational integers as entries. Given a rational character

χ of G, we could compute the unique decomposition of χ w.r.t. $\mathrm{Irr}_{\mathbb{Q}}(G) = \{\chi_1, \dots, \chi_k\}$ by taking the k inner products of χ with each χ_i, but it is generally faster to compute and store the matrix $C \in \mathcal{M}_{k\times n}$ whose rows are the χ_i (where n is the number of classes of G) and then decompose any χ simply by solving the linear system $v \times C = w$ for $v \in \mathbb{Z}^k$, where $w \in \mathbb{Z}^n$ is the vector corresponding to χ. Standard modular techniques can also be used to compute the unique integral vector C.

1.4. Division Algebras and Central Simple Algebras

Definition 1.4.1. *Let A be an algebra of finite dimension over a field F. The algebra A is said to be* **central simple** *over F if A is simple (i.e., 0 and A are the only two-sided ideals of A) and the centre of A is F.*

Theorem 1.4.2. [Hup98, 38.6] *Let A be a central simple algebra over the field F. Then A is isomorphic to $\mathcal{M}_n(D)$ for some division algebra D, with the centre $Z(D)$ of D equal to F.*

Theorem 1.4.3. [Hup98, 38.8, 38.12] *Let D be a division algebra, central over a field F. Then:*

1. $\mathrm{Dim}_F(D) = s^2$ *for some integer s.*
2. *Suppose E is a subfield of $A = \mathcal{M}_m(D)$ and E contains F. Then E is a maximal commutative subalgebra of A if and only if* $\mathrm{Dim}_F(E) = ms$.
3. *Let E be a maximal commutative subfield of D. Then* $\mathrm{Dim}_F(E) = s$. *Such an E always exists.*

Definition 1.4.4. *Let F be a field. Given a monic polynomial*

$$f = x^d + \sum_{i=0}^{d-1} c_i x^i \in F[x],$$

the **companion matrix** C_f *of f is defined to be the following matrix in $\mathcal{M}_d(F)$:*

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & 1 \\ -c_0 & -c_1 & -c_2 & \cdots & -c_{d-1} \end{pmatrix}.$$

Remarks 1.4.5. The essential fact about C_f is that it is the rational form of itself, so its minimal polynomial and characteristic polynomial over F both equal f and its trace equals the trace of f ($-c_{d-1}$, the sum of the roots of f over an algebraic closure).

Proposition 1.4.6. *Let D be a division algebra, central over a field F, with* $\mathrm{Dim}_F(D) = s^2$ *and suppose $m \geq 1$. Then $\mathcal{M}_m(D)$ contains a maximal subfield S containing F and for all such S,* $\mathrm{Deg}_F(S) = ms$.

Proof. By the third point of Thm. 1.4.3, D contains a maximal subfield S_D with $\mathrm{Deg}_F(S_D) = s$. Let f be any irreducible polynomial of degree m over S_D. Then the companion matrix of f is in $\mathcal{M}_m(S_D)$ and thus also in $\mathcal{M}_m(D)$ and it must generate a subfield S of $\mathcal{M}_m(D)$

of degree ms over F. By the second point of Thm. 1.4.3, S must be a maximal subfield of $\mathcal{M}_m(D)$. The last statement also follows by the second point of Thm. 1.4.3. □

1.5. Decomposing over an Extension Field

The following basic results consider what happens to an irreducible representation when moving to an extension field.

Theorem 1.5.1. [CR87, Thm. 74.5] *Suppose $\chi \in \mathrm{Irr}(G)$. Let $C = \mathbb{Q}(\chi)$ and let $s = s_{\mathbb{Q}}(\chi)$. Let $\psi = s \cdot \mathrm{GalSum}_{\mathbb{Q}}(\chi)$, which is the character of an irreducible $\mathbb{Q}$-representation of G, by Thm. 1.2.3. Suppose ρ affords ψ and let $E = \mathrm{End}_{\mathbb{Q}G}(\rho)$. Then E is a division algebra, the centre of E is isomorphic to C, and $\mathrm{Dim}_C(E) = s^2$.*

Proposition 1.5.2. [LP10, 1.5.4] *Let M be a semisimple A-module (a direct sum of simple A-modules), where A is a finite-dimensional algebra over a field F. Suppose*

$$M \cong \oplus_{i=1}^{k} \oplus_{j=1}^{m_i} S_i$$

where the S_i are pairwise non-isomorphic simple modules. Let $E = \mathrm{End}_A(M)$. Then:

- *$E \cong \oplus_{i=1}^{k} \mathcal{M}_{m_i}(D_i)$, where $D_i = \mathrm{End}_A(S_i)$ is a division algebra.*
- *$Z(E) \cong \oplus_{i=1}^{k} F_i$, where the F_i are fields.*

Lemma 1.5.3. *Let F be a field, A an F-algebra and M an A-module of dimension n. Suppose e is an invertible element of $\mathrm{End}_A(M)$. Let f be the minimal polynomial of e over F and let $d = \mathrm{Deg}(f)$. Let E be the field extension $F(\alpha)$ of F, where the minimal polynomial of α over F is f and let S_α be the α-eigenspace of e over E (i.e., the kernel of $e - \alpha$ in $\mathcal{M}_n(E)$). Then S_α is a submodule of M^E of dimension $\frac{n}{d}$.*

Proof. Since e is invertible, its minimal polynomial $f \in F[x]$ is irreducible, so the characteristic polynomial $c_e \in F[x]$ of e must be a perfect power of f_A. Since $e \in \mathcal{M}_n(F)$ and $\mathrm{Deg}_F(f) = d$, we have $c_e = (f_e)^q$, where $q = \frac{n}{d}$. Factoring these polynomials in $E[x]$, $(x - \alpha)$ must occur with multiplicity 1 in f_e and multiplicity n in c_e. So the α-eigenspace of e over E has dimension q and since it is the kernel of an endomorphism of M^E, it is a submodule of M^E, and has dimension $q = \frac{n}{d}$. □

Lemma 1.5.4. *Suppose that $\chi \in \mathrm{Irr}(G)$ and $\rho : G \to \mathrm{GL}_n(F)$ is a representation of G for some number field $F = \mathbb{Q}(\alpha)$ which has a subfield isomorphic to $\mathbb{Q}(\chi)$. Suppose also that E is some splitting field for G which contains $\mathbb{Q}(\chi)$ and F and is such that χ_ρ, the character of ρ, is conjugate to χ, lifted to E. Let $g_1, \ldots, g_k$ be elements of G such that $\{\chi(g_1), \ldots, \chi(g_k)\}$ generate $\mathbb{Q}(\chi)$ over $\mathbb{Q}$. Define the field monomorphism $\phi : \mathbb{Q}(\chi) \to F$ via $\phi(\chi(g_i)) = \chi_\rho(g_i)$ for $1 \le i \le k$. Then under this embedding of $\mathbb{Q}(\chi)$ into F, the characters χ and χ_F are equal, so ρ affords χ.*

Proof. ϕ is well-defined because $\mathbb{Q}(\chi)$ is normal and the characters are conjugate under automorphisms of $\mathbb{Q}(\chi)$. By construction, ϕ identifies the character values of χ with those of χ_ρ for the generators of $\mathbb{Q}(\chi)$ and thus for all character values. □

Corollary 1.5.5. *Suppose $\chi \in \mathrm{Irr}(G)$. Let $s = s_{\mathbb{Q}}(\chi)$ and $\chi_{\mathbb{Q}} = s \cdot \mathrm{GalSum}_{\mathbb{Q}}(\chi) \in \mathrm{Irr}_{\mathbb{Q}}(G)$, and suppose that $\rho_{\mathbb{Q}} : G \to \mathrm{GL}_l(\mathbb{Q})$ affords $m \cdot \chi_{\mathbb{Q}}$ for some $m \ge 1$. Let $E = \mathrm{End}_{\mathbb{Q}G}(\rho_{\mathbb{Q}})$ and let C be the centre of E. Then:*

1. *There exists a maximal subfield F_1 of E which contains C and is such that* $\mathrm{Deg}_C(F_1) = ms$.
2. *Let e be a generator of F_1 over C, let f be the minimal polynomial of e over $\mathbb{Q}$ and let F be the number field $\mathbb{Q}(\alpha)$, where α has minimal polynomial f. Let ρ be the representation of G corresponding to the submodule of $(M_{\mathbb{Q}})^F$ generated by $e - \alpha$, where $M_{\mathbb{Q}}$ is the $\mathbb{Q}G$-module corresponding to $\rho_{\mathbb{Q}}$. Then ρ is absolutely irreducible and under a suitable embedding of $\mathbb{Q}(\chi)$ into F, the character of ρ equals χ. Furthermore, F is a minimal field for χ if $m = 1$.*

Proof. By Thm. 1.5.1 and Prop. 1.5.2, we have that C is isomorphic to $\mathbb{Q}(\chi)$ and $E \cong \mathcal{M}_m(D)$, where D is a division algebra with centre isomorphic to C and $\mathrm{Dim}_C(E) = s^2$. By Prop. 1.4.6, E must then contain a maximal subfield F containing C, with $\mathrm{Deg}_C(F) = ms$, which proves the first point. For the second point, first write $c = \mathrm{Deg}_{\mathbb{Q}}(C)$. Now the degree of $\rho_{\mathbb{Q}}$ equals $msc\chi(1)$ and the degree of f equals msc, so by Lem. 1.5.3, the degree of ρ equals $\chi(1)$. As ρ is also a constituent of $\rho_{\mathbb{Q}}$, whose character is just a sum of conjugates of χ, ρ must be absolutely irreducible and as F contains C which is isomorphic to the normal field $\mathbb{Q}(\chi)$, the character of ρ can be considered equal to χ under a suitable isomorphism from $\mathbb{Q}(\chi)$ to C (giving an embedding of $\mathbb{Q}(\chi)$ into F), as in Lem. 1.5.4. The last statement follows from the remark at the end of Def. 1.2.5. □

1.6. Rewriting over a Subfield

Definition 1.6.1. *Suppose F is a field and $E = F(\alpha)$ is a simple extension field of F, with the monic minimal polynomial of α over F equal to $f \in F[x]$, of degree d. Define the map $\mathcal{B}_{E/F} : E \longrightarrow \mathcal{M}_d(F)$ by*

$$\sum_{i=0}^{d-1} c_i \alpha^i \mapsto \sum_{i=0}^{d-1} c_i (C_f)^i \quad (c_1, \ldots, c_{d-1} \in F),$$

where C_f is the companion matrix of f (see Def. 1.4.4). It is easy to see that $\mathcal{B}_{E/F}$ is an F-algebra monomorphism. We can also naturally extend $\mathcal{B}_{E/F}$ to an F-algebra monomorphism

$$\mathcal{B}_{E/F} : \mathcal{M}_n(E) \longrightarrow \mathcal{M}_{nd}(F).$$

Proposition 1.6.2. *Suppose $\rho_E : G \longrightarrow \mathrm{GL}_n(E)$ is a representation affording χ and suppose F is a subfield of E, where $\mathrm{Deg}_F(E) = d$. Define a new representation $\rho_F : G \longrightarrow \mathrm{GL}_{nd}(F)$ by*

$$g \mapsto \mathcal{B}_{E/F}(\rho_E(g)),$$

which we call the **restriction of scalars** *of ρ_E from E to F. Then:*

1. *ρ_F is a representation of G and the character of ρ_F equals the trace w.r.t. F of χ (obtained by applying $\mathrm{Tr}_{E/F}$ to each value of χ).*
2. *Suppose also that E is a minimal extension of F such that ρ_E affords χ. Then ρ_F is irreducible.*

Proof. 1. It is trivial to check that ρ_F is a valid representation and the statement on the character follows from the fact that for $x \in E$, $\mathrm{Tr}(\mathcal{B}_{E/F}(x)) = \mathrm{Tr}_{E/F}(x)$.

2. Let $\psi \in \mathrm{Irr}_F(G)$ be the F-irreducible character containing χ and let $\rho_\psi : G \to \mathrm{GL}_n(F)$ be any F-representation which affords ψ. Let E_1 be a maximal subfield of $\mathrm{End}_{FG}(\rho_\psi)$. Then some constituent of $(\rho_\psi)^{E_1}$ affords χ and $\mathrm{Deg}(\psi) = |E_1 : F| \cdot \mathrm{Deg}(\chi)$. Now let χ_F be the character of ρ_F. Then $\mathrm{Deg}(\chi_F) = |E : F| \cdot \mathrm{Deg}(\chi)$ and since E is minimal, we must have $|E : F| \leq |E_1 : F|$, so $\mathrm{Deg}(\chi_F) \leq \mathrm{Deg}(\psi)$ and we must thus have equality, so $\chi_F = \psi$ and ρ_F is irreducible.

□

1.7. Algorithms for Integral Matrices

In this section we describe fundamental operations and associated algorithms for integral matrices which are of critical importance for constructing ordinary representations.

1.7.1. Hermite Form.

Definition 1.7.1. *A matrix $T \in \mathcal{M}_n(\mathbb{Z})$ is called* **unimodular** *if T is invertible over $\mathbb{Z}$; i.e., if its determinant is ± 1.*

Definition 1.7.2. *Suppose $A \in \mathcal{M}_{m\times n}(\mathbb{Z})$. The (row) Hermite form of A is the unique matrix $H = TA$ for unimodular $T \in \mathcal{M}_m(\mathbb{Z})$ such that:*

- *Rows $[1, \dots, r]$ of H are non-zero and rows $[r+1, \dots, m]$ are zero, where r is the rank of A.*
- *If c_i is the column of the first non-zero entry of row i (for $1 \leq i \leq r$), then $c_1 < c_2 < \dots < c_r$, and for $1 \leq i \leq r$: $d_i = H[i, c_i]$ is positive, $H[k, c_i] < d_i$ for $1 \leq k < i$ and $H[k, c_i] = 0$ for $i < k \leq r$.*

A good effective classical (non-modular) algorithm for computing the Hermite form was described by Kannan & Bachem [KB79] (with improved bounds given in [CC82]). The basic algorithm simply takes m steps and at the end of k-th step, the first k rows of A are replaced with the Hermite form of the first k rows of A. The k-th step involves expanding the Hermite form of the first k rows to include the k-th row (using euclidean operations and basic row operations).

A modular technique was suggested by Micciancio & Warinschi in [MW01] to compute the Hermite form of a $n \times n$ integral matrix of full rank n, under the assumption that the index g of the lattice generated by the first $n-1$ columns of A in $\mathbb{Z}^{n-1}$ is very small. This is the case at least for matrices with random entries bounded by some bit length. We have implemented an extension of this algorithm which works on an arbitrary $m \times n$ integral matrix A with any rank. We will let HERMITEFORM(A) denote the algorithm which returns the Hermite form of A.

1.7.2. Smith Form.

Definition 1.7.3. *Suppose $S \in \mathcal{M}_{m\times n}(\mathbb{Z})$ and has rank r. The matrix S is said to be in* **Smith (normal) form** *if $e_i = S_{[i,i]}$ is positive for $1 \leq i \leq r$, S is zero elsewhere, and $e_i | e_{i+1}$ for $1 \leq i < r$.*

Theorem 1.7.4. [Smi61], [Coh93, 2.4.12] *Suppose $A \in \mathcal{M}_{m\times n}(\mathbb{Z})$ and A has rank r. Then there exists a unique matrix $S \in \mathcal{M}_{m\times n}(\mathbb{Z})$ which is in Smith normal form such that $S = PAQ$ for unimodular matrices $P \in \mathcal{M}_m(\mathbb{Z}), Q \in \mathcal{M}_n(\mathbb{Z})$. The matrix S is called the* **Smith (normal) form** *of A. Note that P and Q are not unique in general.*

Definition 1.7.5. *Define the* **elementary divisors** *of A to be the non-zero positive integers $[e_1, \ldots, e_r]$ on the diagonal of the Smith form of A (so $e_i | e_{i+1}$ for $1 \leq i < r$). (Note that we call a matrix A 'diagonal' if it has non-zero entries only on its diagonal i.e., $A[i, j] = 0$ for $i \neq j$; the matrix need* **not** *be square.)*

For computing the Smith form S of a matrix $A \in \mathcal{M}_{m \times n}(\mathbb{Z})$, our MAGMA implementation uses the following strategy:

1. If A is dense, then first a multiple D of the largest elementary divisor of A is computed using the method outlined in [ABM99]; if D is smooth, then the modular algorithm of F. Lübeck [Lüb02] is then used to compute S. Otherwise, the algorithm repeatedly calls the dense Hermite form algorithm above and transposes, until a diagonal form is obtained; the divisibility condition on the diagonal is easily obtained by successively computing GCDs and LCMs of adjacent diagonal entries.
2. If A is sparse, then first sparse elimination is performed via Markowitz pivoting [DER84, Sec. 9.2] to obtain a smaller dense matrix A_1 with density at least 50% (this is similar to using the techniques described in [HHR93]), and then the above methods are applied to the dense matrix A_1.

We will let ELEMENTARYDIVISORS(A) denote the algorithm which computes the Smith form of A and returns the elementary divisors of A.

1.7.3. Saturation.

Definition 1.7.6. *Let $L \subseteq \mathbb{Z}^n$ be a lattice of rank r. Define the* **saturation** *of L to be $(L \otimes \mathbb{Q}) \cap \mathbb{Z}^n$, where $L \otimes \mathbb{Q}$ is the subspace of the vector space $\mathbb{Q}^n$ generated by L. L is also said to be* **saturated** *if its saturation equals itself. (Note: some authors also use the terms 'purified lattice'/'purified' instead of 'saturation'/'saturated'.)*

Lemma 1.7.7. *If $L, L' \subseteq \mathbb{Z}^n$ are lattices which have the same $\mathbb{Q}$-span and L' is saturated, then L' equals the saturation of L.*

Proof. The saturation of L is $(L \otimes \mathbb{Q}) \cap \mathbb{Z}^n = (L' \otimes \mathbb{Q}) \cap \mathbb{Z} = L'$. □

Lemma 1.7.8. *If $L \subseteq \mathbb{Z}^n$ is a lattice of rank r and B is a basis matrix of L with trivial elementary divisors, then L is saturated.*

Proof. Let $S = PBQ$ be the Smith form of B, where P and Q are unimodular and $S = [I_r|0]$. Suppose v is in the saturation of L. Since P is unimodular, PB is also a basis matrix for L and we can write $v = uPB$ for $u \in \mathbb{Q}^r$. Since $v \in \mathbb{Z}^n$ and Q is unimodular, $vQ \in \mathbb{Z}^n$ also, so $vQ = uPBQ = uS \in \mathbb{Z}^n$ and u must be integral since $S = [I_r|0]$. Thus $v = uPB \in L$. □

Proposition 1.7.9. *Suppose $L \subseteq \mathbb{Z}^n$ is a lattice of rank r. Let $B \in \mathcal{M}_{r \times n}(\mathbb{Z})$ be a basis matrix of L. The saturation L' of L can be computed by either of these methods:*

1. *Let $S = PBQ$ be the Smith form of B, where P and Q are unimodular and let $[e_1, \ldots, e_r]$ be the elementary divisors of B (the non-zero diagonal entries of S). Then let $[v_1, \ldots, v_r]$ be the rows of PB and set $w_i = \frac{1}{e_i} v_i \in \mathbb{Z}^n$ for $1 \leq i \leq r$. Set $L' \subseteq \mathbb{Z}^n$ to the lattice spanned by $[w_1, \ldots, w_r]$.*

2. *Set $H_1 \in \mathcal{M}_{n\times r}(\mathbb{Z})$ to the* **column** *Hermite form of B (i.e., the transpose of the usual row Hermite form of the transpose of B, so $H_1 = BT$ for unimodular T). Let H_2 equal the first r columns of H_1 (the rest are zero). Let $U = H_2^{-1} \in \mathcal{M}_r(\mathbb{Q})$ and $W = UB$, which is integral. Set $L' \subseteq \mathbb{Z}^n$ to the lattice spanned by the rows of W.*

Proof. 1. Since S has only e_i as a non-zero entry in the i-th row and multiplication by Q^{-1} on the right only does column operations, the same holds for $PB = SQ^{-1}$, so v_i must be divisible by e_i for $1 \le i \le r$. The matrix whose rows are the w_i must have trivial elementary divisors by construction, so L' is saturated and has the same $\mathbb{Q}$-span as L.

2. We have $H_1 = BT_1$ for some unimodular $T_1 \in \mathcal{M}_n(\mathbb{Z})$ and since B has rank r, we must have $H_1 = [H_2|Z]$, where H_2 is non-singular and Z is the $r \times (n-r)$ zero matrix. Then $WT_1 = UBT_1 = UH_1 = [UH_2|Z] = [I_r|Z]$ (since $UH_2 = I_r$), so W is integral and the Smith form of W equals $[I_r|Z]$ so L' is saturated and has the same $\mathbb{Q}$-span as L.

□

The first method to compute the saturation of a lattice is well known, but our MAGMA implementation uses the second method, since we have already implemented fast modular algorithms to compute both the Hermite form and inverse. The time for the whole algorithm is in general very much dominated by the initial column Hermite form computation, as will be seen in examples. For the matrices arising in the 'integral spin' algorithm presented later (to compute the submodule of a module generated by some integral vectors), it is often the case that $n >> r$ (e.g., $r \sim 500$ and $n \sim 10000$).

To avoid switching back and forth between lattices and their basis matrices, we will let SATURATION(B) denote the algorithm which takes a basis matrix $B \in \mathcal{M}_{r\times n}(\mathbb{Z})$ for a rank-r lattice L and returns a basis matrix for the saturation of L.

Lemma 1.7.10. *Suppose $A \in \mathcal{M}_n(\mathbb{Z})$ is non-singular (i.e., has rank n). Then the lowest common denominator of $A^{-1} \in \mathcal{M}_n(\mathbb{Q})$ is e_n, the largest elementary divisor of A.*

Proof. Let S be the Smith form of A, so $S = PAQ$ with P, Q with unimodular. Then S is a diagonal matrix with non-zero diagonal entries $[e_1, \dots, e_n]$, so over $\mathbb{Q}$ we have $A^{-1} = QS^{-1}P$ and the lowest common denominator of S^{-1} is clearly e_n and multiplication by the unimodular P and Q does not change this. □

Proposition 1.7.11. *Suppose that $B \in \mathcal{M}_{r\times n}(\mathbb{Z})$ is a basis matrix for a rank-r sublattice L of $\mathbb{Z}^n$. Let V be the subspace of $\mathbb{Q}^n$ generated by L (so B is also a $\mathbb{Q}$-basis of V). Suppose that $A \in \mathcal{M}_n(\mathbb{Z})$ and V is invariant under right multiplication by A. Let e_r be the largest elementary divisor of B. Then there is a unique matrix $X \in \mathcal{M}_r(\mathbb{Q})$ satisfying $XB = BA$ and the lowest common denominator of X is a divisor of e_r (in particular, X is integral if $e_r = 1$, i.e., if L is saturated).*

Proof. Let $S = PBQ$ be the Smith form of B, where P and Q are unimodular and S is a diagonal matrix with non-zero diagonal entries $[e_1, \dots, e_r]$. Since P is unimodular, PB is also a basis matrix for L and we can write $BA = UPB$ for unique $U \in \mathcal{M}_r(\mathbb{Q})$ (since the rowspace of BA is a subspace of V). Since BA is integral and Q is unimodular, $BAQ = UPBQ = US$ is also integral. Thus Ue_r is also integral, so the lowest common denominator of U must be a divisor of e_r. Setting $X = UP$, the lowest common denominator of X must also be a divisor of e_r since P is unimodular, and $XB = UPB = BA$. □

1.7.4. Minimal and Characteristic Polynomial. To compute the minimal or characteristic polynomial of a matrix $A \in \mathcal{M}_n(\mathbb{Z})$, our implementation uses algorithms similar to those described in [CLG97] and [DPW05]. The basic idea is to choose an initial non-zero vector $v \in \mathbb{Z}^n$ and compute the smallest d such that the vectors $v, vA, vA^2, \ldots, vA^d$ are linearly dependent; the corresponding relation gives a polynomial f such that $v \cdot f(A) = 0$, so f is a divisor of the minimal polynomial of A, and the submodule of $\mathbb{Z}^n$ generated by the above vectors is called the *Krylov subspace* generated by v. In practice, the algorithm first finds the relation modulo a suitable prime p, and then p-adically lifts this to the integral relation (using a technique similar to that described in [Dix82]). If the degree of f equals n, then the minimal and characteristic polynomials of A are equal and f equals them (this is a common situation). Otherwise, the algorithm computes another Krylov subspace generated by a new vector v_2 not in the current submodule and combines the results, iterating as needed until rank n is reached (working in the quotient space and multiplying the resulting polynomials for the characteristic polynomial, or computing the LCM of the resulting polynomials for the minimal polynomial; see the above references for details).

There is one very simple but useful extension to this algorithm which we will use later. Suppose that we have computed $v, vA, vA^2, \ldots, vA^d$ and the corresponding f as above, so that $v \cdot f(A) = 0$. Suppose also that g is an irreducible factor of f such that the multiplicity m of g in f equals the multiplicity of g in the characteristic polynomial of A. Then we can compute the nullspace of $g^m(A)$ efficiently as follows:

1. Set $q = f/g^m \in \mathbb{Z}[x]$ and write $e = \mathrm{Deg}(g^m)$.
2. Set $w_i := v \cdot (x^i q)(A)$ for $0 \leq i < e$.
3. Set $B := [w_0, \ldots, w_{e-1}]$.

It is easy to see that B is a $\mathbb{Q}$-basis for the nullspace of $g^m(A)$ since $w_i \cdot g^m(A) = v \cdot x^i f = 0$ for $0 \leq i < e$, and the w_i are linearly independent since the degree of $x^i q$ is less than d for $0 \leq i < e$. Each w_i can be computed as a linear combination of the already known vA^i vectors, so further multiplication by A is avoided and the number of arithmetic operations is $O(e(n-e)n)$. One can then compute the saturation of the lattice spanned by the rows of B to obtain the nullspace over $\mathbb{Z}$. This method is particularly useful when the degree of g^m is rather high, since it avoids the computation of $g^m(A)$ (which takes $O(e \cdot \mathsf{MM}(n))$ arithmetic operations, where $\mathsf{MM}(n)$ denotes the complexity of the matrix multiplication algorithm).

1.8. Lattice Basis Reduction Tools

1.8.1. LLL reduction. The Lenstra-Lenstra-Lovász (LLL) algorithm [LLL82] takes a basis B of a lattice L and returns a *LLL-reduced* basis B' of L. In practice, the entries of B' are often much smaller than the entries of B (see the reference for the precise definition of 'LLL-reduced'). The algorithm is very useful in many areas of computational algebra. We cannot over-emphasize the fact that it contributes enormously to the effectiveness of our algorithms. For a detailed exposition and analysis of the algorithm, we refer the reader to the recent book [NVe09]. We just note here some basic properties of the algorithm.

Theorem 1.8.1. [Coh93, 2.6.2] *Let $v_1, \ldots, v_k$ be a LLL-reduced basis of a lattice L. Then for any non-zero $w \in L$, we have $|v_1| \leq 2^{(n-1)/2}|w|$.*

A parameter δ is used in the algorithm and by default it is usually set to 3/4 (including in the MAGMA implementation). But it may be set to any value in the range $1/2 < \delta < 1$ and then the base 2 in the bound of the above theorem can be replaced with $1/(\delta - 1/4)$. Taking the value of δ to be just under 1 (say 0.999), the algorithm can run slower in general, but the output will often have better quality in general; the base of the above bound becomes close to 4/3.

We note also that there is a simple extension of the original algorithm, called MLLL ('modified LLL') [Poh93, Alg 3.8] which takes a set S of vectors in $\mathbb{Z}^n$ which are not necessarily independent; the output is a LLL-reduced basis of the lattice spanned by S. For simplicity, we will let 'LLL' refer to the extended algorithm (just as the MAGMA implementation does).

We use the implementation of the algorithm in MAGMA by D. Stehle [NS09b, Ste09]. The algorithm is very effective for the kinds of lattices which we encounter even if the rank is over 1000 (particularly if the matrix is first reduced to Hermite form; see Sec. 3.4 below for more discussion).

1.8.2. Seysen Reduction. Let L be a lattice of rank n with basis $B = (b_1, \cdots, b_n)$. The dual lattice L^* of L is defined by the basis vectors $(b_1^*, \cdots, b_n^*)$, where $(b_i, b_i^*) = 1$, $(b_i, b_j^*) = 0$, for $1 \leq i, j \leq n$, $j \neq i$. Seysen introduced a lattice basis reduction algorithm which computes simultaneous reduction of a lattice basis and its corresponding dual basis [Sey93]. LaMacchia analyzed the algorithm and described a practical heuristic version of the algorithm [LaM91] (the original motivation was for cryptographic problems). The author has implemented LaMacchia's version of the algorithm in MAGMA.

The usefulness of the algorithm in the context of ordinary representations is that when computing the reduced action of a reducible integral representation $\rho : G \to \mathrm{GL}_n(\mathbb{Z})$ on a saturated invariant sublattice S of $\mathbb{Z}^n$, then if a basis B of S is reduced by Seysen's algorithm, this tends to reduce the size of the entries in the matrices defining the corresponding representation. As the degree increases, the algorithm's cost increases and often its effectiveness decreases (i.e., it often does not reduce much more than LLL), but it is certainly worth applying in up to moderate dimensions to reduce the entries, and Ex. 3.7.3 below presents an example where Seysen reduction is worth using in a higher dimension.

1.9. Computing Homomorphisms and Endomorphisms

Let A be a finite-dimensional algebra over a field F and suppose that M_1 and M_2 are A-modules. We outline efficient algorithms to compute $\mathrm{Hom}_A(M_1, M_2)$ and $\mathrm{End}_A(M_1)$ for each kind of field which we will encounter.

1.9.1. Homomorphisms over a Finite field. Suppose that F is a finite field. Our implementation uses two methods to compute $\mathrm{Hom}_A(M_1, M_2)$:

1. If M_1 is semisimple, then first the composition factors of M_1 are computed using the modular Meataxe and then a basis of the Hom-module is constructed from the homomorphisms from C into M_2, for each irreducible constituent C of M_1 (using the algorithm given in [HR94]).
2. In the general case, we use an algorithm of C. Leedham-Green and the present author developed in 1994 (unpublished), which is very similar to the algorithm given in [LS03], except that the vectors chosen to generate submodules of M_1 are chosen from the transformation matrix corresponding to the generalized Jordan form of a random algebra element instead of using peakwords.

The modules which arise in this thesis are practically always semisimple, so the first method can usually be used, which is faster in general. Computing $\mathrm{End}_A(M_1)$ is simply done by computing $\mathrm{Hom}_A(M_1, M_1)$. Also, it is easy to adapt the first method above to an efficient algorithm to compute the centre of the endomorphism ring of M_1.

1.9.2. Homomorphisms over the Rational Field. Suppose now that F equals $\mathbb{Q}$. We have implemented a modular algorithm HOM to compute $\mathrm{Hom}_A(M_1, M_2)$. The algorithm uses the standard 'small primes with Chinese Remaindering' modular scheme (see [vzGG03, Fig. 5.2]), as follows.

1. For each successive prime p_i, the algorithm computes an echelonized form of the basis of the corresponding Hom-module over $\mathbb{F}_{p_i}$.
2. The modular basis matrices are then combined by the Chinese Remainder Theorem [vzGG03, 5.4] to obtain the basis matrix modulo $P = \prod_{i=1}^{k} p_i$ after the k-th step. The algorithm then attempts rational reconstruction of each entry of the basis modulo P to obtain the echelonized basis over $\mathbb{Q}$. Rational construction ([vzGG03, 5.10], [Mon04]) takes an integer residue x with $0 \leq x < P$ and determines whether there is a rational $\frac{n}{d} \in \mathbb{Q}$ with $(d, P) = 1$, $x \equiv n \cdot d^{-1} \pmod{P}$, $|n| \leq B_N$ and $0 < d \leq B_D$, where B_N, B_D are positive integer bounds with $2B_N B_D \leq P$; the solution is unique if there is one.
3. If the rational reconstruction of each entry succeeds, then the algorithm simply checks that the associated rational matrices actually form a basis of homomorphisms for the original input modules (this simply involves checking that $a_{1,j}h_i = h_i a_{2,j}$ for $1 \leq i \leq r$ and $1 \leq j \leq k$, where r is the dimension of the Hom-module and k is the number of generators of A and the $a_{1,j}$ and $a_{2,j}$ are the matrices of the action of M on A_1 and A_2 respectively). If the check passes, then the algorithm is finished; otherwise it continues with more primes.
4. A so-called 'bad prime' p is such that the Hom-module of the modulo-p reduction of the input does not equal the modulo-p reduction of the Hom-module of the rational input modules. For such a p, the pivot structure of the echelonized basis matrix modulo p will not match the pivot structure of the correct rational echelonized basis and this can easily be detected by comparing the new modular pivot structure with that of the current pivot structure (coming from the previous primes). The set of bad primes must be finite, since they either divide an input denominator or a denominator of an entry in the echelonized rational basis. So it is easy to detect and reject any bad primes and sufficiently many good primes will always be found. Note also that if r is the rank of

the correct rational Hom-module, then a good prime will always give a Hom-module of rank r, so the resulting rational basis will have the correct rank.

5. In our implementation within MAGMA, the entries of the matrices over $\mathbb{F}_p$ are represented by exact-integer double-precision floating point numbers. The algorithm chooses each prime p to be just below $2^{23.5}$, so $64p^2 < 2^{53}$ (the maximum integer which can be represented exactly) so that 64 products of integers between 0 and $p-1$ can be added before reducing the sum modulo p. Several critical matrix operations such as echelon form, inverse, determinant and rank are mapped to fast multiplication routines which use the ATLAS (Automatically Tuned Linear Algebra Software) library of Whaley [WP05, Wha] and also Strassen's asymptotically-fast matrix multiplication algorithm [Str69] when the dimension is above 1024. Strassen's algorithm is not just of theoretical interest, since later in the thesis those operations are applied to matrices with dimensions in the thousands, and this algorithm gives a very significant practical improvement.
6. For rational reconstruction, our implementation uses an asymptotically-fast version of the algorithm, which is similar to the 'Half-GCD' algorithm of [AHU75, 8.9]. Rational reconstruction is often applied with the numerator bound B_N and denominator bound B_D both taken to be $\lfloor \frac{\sqrt{P}}{2} \rfloor$, but it is better in practice to make the bounds tighter, which means that if the whole basis reconstructs successfully, then the probability that it is correct is much higher, so that in practice the verification in point 3 above will virtually always only be tried when the current result is already correct.

Note also that if A is a $\mathbb{Z}$-algebra, then one can compute $\mathrm{Hom}_A(M_1, M_2)$ for A-modules M_1, M_2 by applying the above modular algorithm over $\mathbb{Q}$ and then saturating the result by the methods of the previous section.

Plesken & Souvignier also presented algorithms [PS96] for computing homomorphisms and endomorphisms over $\mathbb{Q}$ by the averaging operator technique (see also [Sch02, 2.2]), but we have found that the modular algorithm is generally faster and preferable, particularly since it is better to compute the full endomorphism ring so that it can be saturated and LLL-reduced so that small endomorphisms can be used, and subsequent operations will have matrices with smaller entries.

1.9.3. Homomorphisms over a Number Field. Suppose $F = \mathbb{Q}(\alpha)$, where the minimal polynomial of α is $f \in \mathbb{Q}[x]$, of degree d. We have also implemented a fast modular algorithm to compute $\mathrm{Hom}_A(M_1, M_2)$, where A is an F-algebra. This algorithm is very similar to the above modular algorithm for rational modules, except for the following extensions:

1. Each prime p is chosen so that f has d distinct roots $\beta_1, \dots, \beta_d$ in $\mathbb{F}_p$ and then for each root β_i, we reduce the input entries modulo p and map α to β_i, compute the echelonized basis modulo p and combine the d results by interpolation ([vzGG03, 5.2]) to obtain each entry in $\mathbb{F}_p[x]/\langle f \rangle$.
2. The algorithm proceeds as above, using Chinese remainder on the successive primes and rational reconstruction on the entries in $(\mathbb{Z}/(P\mathbb{Z}))[x]\langle f \rangle$ of the basis matrix: the only difference is that there are d times as many modular entries to which we apply Chinese remaindering and rational reconstruction. The termination check involving

the matrix products is the same (and a modular algorithm can be used in the matrix multiplications).

1.9.4. Endomorphisms over $\mathbb{Q}$ or a Number Field. We also have the following similar modular algorithms for computing endomorphisms over $\mathbb{Q}$ or a number field:

1. EndomorphismRing(M): computes $\mathrm{End}_A(M)$ by computing $\mathrm{Hom}_A(M, M)$ (the inner algorithms can be simplified of course because of the repeated module).
2. CentreOfEndomorphismRing(M): computes the centre of $\mathrm{End}_A(M)$ by making the inner modular algorithm compute the centre of the endomorphism ring over the inputs reduced modulo p (and this centre can be computed in the semisimple case very efficiently via the Meataxe). This algorithm often requires less primes than when computing the full endomorphism ring (when the dimension of the centre is smaller) and is useful for decomposing modules into homogeneous components.

1.10. Entry Reduction of a Rational Representation

Suppose G is a finite group and $\rho : G \longrightarrow \mathrm{GL}_n(F)$ a representation of G, where F is $\mathbb{Q}$ or a number field. We use the terminology '**entry reduction** of ρ' to denote some computation which yields an equivalent representation ρ' which typically has smaller entries than ρ. We first outline well-known methods to reduce the entries of a rational or integral representation.

Given a rational representation $\rho : G \longrightarrow \mathrm{GL}_n(\mathbb{Q})$, ρ can always be conjugated to an integral representation [KP02]. There is a simple practical method to do this, as follows. Let $M = \rho(G)$ (i.e., the matrix group defined by the image of ρ). Since M is finite, the denominators of all entries of elements of M are bounded and thus the M-invariant set $L = \{v \cdot g | v \in \mathbb{Z}^n, g \in M\}$, is a sublattice of $\mathbb{Z}^n$ of finite index. Then conjugating ρ by a basis matrix of L gives an integral representation which is equivalent to ρ.

Now for a given integral representation $\rho : G \longrightarrow \mathrm{GL}_n(\mathbb{Z})$, let $M := \rho(G)$ again and compute a positive definite form F which is invariant under M, using, for example, the iterative algorithm in [PS96] (the original statement of the algorithm in [PS96] used a fixed generating set of the matrix group, but this is improved in [Sou09] by applying the product replacement algorithm [CLGM$^+$95] after each iteration step to speed up the convergence). After applying LLL-reduction to the Gram matrix F (and optionally also Seysen reduction) to obtain a reduced Gram matrix F' and transformation matrix T such that $F' = T \cdot F \cdot T^{tr}$, simply set $\rho' := \rho^T$. The basic idea is that since the new representation ρ' fixes the form F', so if F' has smaller entries than F, then ρ' will in general have smaller entries than ρ.

If the degree n is up to about 20, then this approach tends to conjugate any rational or integral representation, no matter how large its entries, to an integral one with extremely small entries (single digit and often sparse). But as n grows, the quality of the output diminishes. For $n > 100$, the algorithm often has very little effect on the size of the entries. The basic reason is the increasing weakness of LLL as the dimension increases for computing a minimally-reduced basis: see the bound in Thm. 1.8.1 on the ratio between the shortest vector of a LLL-reduced basis and a shortest vector of lattice. So for small n, this algorithm is very effective at producing an equivalent representation with very small entries but for larger n the algorithm is not very useful.

Another limitation is that there is no obvious way to extend the above algorithm to a method to reduce the entries of a representation $\rho : G \to \mathrm{GL}_n(F)$ defined over an irrational number field F. Given such a ρ, we can always compute the restriction of scalars of ρ to $\mathbb{Q}$, and then reduce that rational representation using the above algorithm, but it is often very difficult to extract an irreducible constituent of this over F again with small entries (see more discussion on this issue on p. 75).

We will introduce a new algorithm for reducing the entries of a representation in Chapter 6, which works very effectively for representations with degrees in the hundreds or even thousands and which are defined over number fields. The new algorithm still relies upon LLL-reduction, but the dimension of the relevant lattice is typically much smaller than the degree n.

CHAPTER 2

A Rational Meataxe

2.1. Introduction

Let M be an A-module, where A is a finite-dimensional algebra over a field F. If F is a finite field, then Parker's **Meataxe** algorithm [Par84] is a very effective algorithm for determining whether M is simple, and for finding a proper submodule of M when it is not simple. Holt & Rees later described an improved version of the algorithm [HR94]. The basic approach is to generate a random element $a \in A$ and then to consider the submodule of M generated by a non-zero element of some generalized eigenspace of a. If the submodule is not proper, then a criterion is applied to attempt to determine whether M simple. When we try to extend the same algorithm to a **rational Meataxe** (where $F = \mathbb{Q}$), there are several major difficulties. These have been well-known for some time and various techniques to overcome these have been proposed by Holt [Hol98], Plesken & Souvignier [PS96] and Parker [Par98] and others. Besides the practical issue of growth of the matrix entries (which can make computations of even moderate degree infeasible), there are least two major algorithmic problems:

1. The traditional Meataxe criterion to prove the simplicity of M may fail (in particular, if M is a $\mathbb{Q}G$-module and has a constituent with a non-trivial Schur index, then the criterion will fail).
2. Even if it is known that M has a proper submodule, it may be very hard to find one.

In this chapter we describe a rational Meataxe; using our implementation of this, the first problem is now easily solvable in practice, and the second problem can now be solved in most situations which arise in practice. The algorithm will only apply to semisimple A-modules, so in this case, a module will be simple if and only if it is indecomposable, and our algorithm will return a direct sum decomposition of its input. The two types of semisimple module to which we will later apply the rational Meataxe algorithm are as follows.

1. M may be a $\mathbb{Q}G$-module, in which case information from the character table of G may also be used.
2. M is a condensed A-module so A is a condensed algebra (see next chapter for details), in which case information involving the trace of the action of A can also be used.

An A-module M is called **homogeneous** if it is isomorphic to the direct sum of one or more copies of the same simple A-module S; i.e., if $M \cong \oplus_{i=1}^{m} S$ for some $m \geq 1$. In practice, it is straightforward to split a module into homogeneous components, but it can be much harder to decompose each homogeneous component; this requires analysis of its endomorphism ring. Algorithmic techniques using this approach were first described by

Plesken & Souvignier [PS96], but more recent improvements have been proposed, based on using a maximal order of the endomorphism ring [NS09a, Sou09]. We outline alternative methods based on tools from Arithmetic Geometry and Cohomology to split homogeneous modules for which the centre of the endomorphism ring has large dimension.

Note that in the usual usage of the traditional (modular) Meataxe to find a composition series of M, if a proper submodule S of M is found, then one typically recurses on S and the quotient module M/S. We avoid this approach in characteristic zero, since it is harder to control the growth of coefficients in the quotient module and recursively constructed submodules (and the basis for their embedding into the original module M); rather, it is better to compute a direct sum decomposition of M without a recursive splitting if possible. Also, if the algebra A has generators with entries in $\mathbb{Z}$ alone, then the algorithm always returns submodules such that the reduced action is also integral.

2.2. Decomposing into Homogeneous Components

The following simple algorithm first decomposes a semisimple module M over $\mathbb{Q}$ into homogeneous components.

```
Algorithm HomogeneousComponents($M$)
Input:
    • An $A$-module $M$ where $A$ is a subalgebra of $\mathcal{M}_n(\mathbb{Q})$.
Output:
    • Submodules $[S_1, \dots, S_k]$ of $M$ such that $M = \oplus_{i=1}^{m} S_i$, and the $S_i$ are homogeneous.
Steps:
  1. Set $Z :=$ CentreOfEndomorphismRing($M$).
  2. Set $d := \mathrm{Dim}_{\mathbb{Q}}(Z)$. If $d = 1$ then return $[M]$.
  3. Set $B :=$ LLL(Saturation(Basis($Z$))).
  4. For $b$ in $B$ do:
     {
        Set $f$ to the minimal polynomial of $b$.
        If $f$ is irreducible and $\mathrm{Deg}(f) = d$ then return $[M]$.
        Factorize $f$ as $\prod_{i=1}^{k} g_i^{e_i}$ with the $g_i$ irreducible.
        If $k > 1$ then:
        {
           Set $S_i$ to the submodule of $M$ generated by $(g_i^{e_i})(b)$ for $1 \le i \le k$.
           Set $L_i :=$ HomogeneousComponents($S_i$) for $1 \le i \le k$.
           Return the concatenation of $L_1, \dots, L_k$.
        }
     }
  5. Return $[M]$.
```

Lemma 2.2.1. *Algorithm* HomogeneousComponents *is correct.*

Proof. By Prop. 1.5.2, the centre Z of $\mathrm{End}_A(M)$ is isomorphic to a direct sum of m fields, where m is the number of homogeneous components of M. If $m = 1$, then all the minimal polynomials will be irreducible (if degree d is encountered, then that immediately proves that Z is a field), and so the single homogeneous component M will be correctly returned. If $m > 1$ then an element of the basis B must split Z (by [CIW97, Cor. 13]), and then the recursive call ensures a complete splitting into homogeneous submodules by induction. □

In the implementation, we use the modular algorithm to compute the centre Z of the endomorphism ring (see p. 25). Note that this can often be computed more more quickly than the full endomorphism ring, so it is well worth using the modular algorithm for the initial decomposition via the centre Z (one can also use the regular representation of Z on itself to reduce the dimension). We use the LLL-reduced basis B of Z so that it is generally faster to compute the minimal polynomials and also so that the bases of the submodules in the decomposition will tend to have smaller entries.

2.3. Splitting Homogeneous Modules

2.3.1. Introduction. This section presents algorithms to split homogeneous A-modules over $\mathbb{Q}$. A very useful approach is to use a maximal order of the endomorphism ring.

Lemma 2.3.1. *Suppose M is a homogeneous A-module, where A is a subalgebra of $\mathcal{M}_n(\mathbb{Q})$, so $M \cong \oplus_{i=1}^{m} S$ for a simple A-module S and $m \geq 1$. Let $E = \mathrm{End}_A(M)$. Then $E \cong \mathcal{M}_m(D)$, where D is a division algebra with $F = Z(D)$ a field and $\mathrm{Dim}_F(D) = s^2$ for some integer $s \geq 1$.*

Proof. This follows directly from Thm. 1.4.2 and Thm. 1.4.3 (1). □

Remarks 2.3.2. The integer s in the last Lemma is called the **Schur index** of the central simple algebra E. (It is easy to see that if M is an $\mathbb{Q}G$-module, then s equals the Schur index of the character of an absolutely irreducible constituent of M.)

Definition 2.3.3. *Let A be a subalgebra of $\mathcal{M}_n(\mathbb{Q})$. An* **order** *of A is a finitely-generated subring O of A such that $\mathbb{Z}$ is in the centre of O and $O \otimes \mathbb{Q} = A$ (so O generates A over $\mathbb{Q}$). A* **maximal order** *of A is an order O such that no other order of A properly contains O.*

Remarks 2.3.4. Let A be a subalgebra of $\mathcal{M}_n(\mathbb{Q})$. The saturation S of $A \cap \mathcal{M}_n(\mathbb{Z})$ (see Def. 1.7.6) is an order of A but is not always maximal. A maximal order O of A will contain S but may also contain elements of $\mathcal{M}_n(\mathbb{Q})$ which are not in $\mathcal{M}_n(\mathbb{Z})$ (see Ex. 2.3.7 below), but every element of O is always integral (has monic minimal polynomial in $\mathbb{Z}[x]$) [Rei03, 8.6]. Note also that if A is isomorphic to a number field F, then a maximal order of A is isomorphic to a maximal order of F.

G. Nebe and the current author developed an algorithm (implemented in Magma) to compute a maximal order of a central simple algebra and recognize the associated Schur index and multiplicity. Since we use the algorithm heavily in subsequent algorithms, we state its specification formally here. See [NS09a] for a detailed description of the algorithm.

Algorithm MAXIMALORDER(E)
INPUT:

- A central simple matrix algebra $E \subseteq \mathcal{M}_n(\mathbb{Q})$.

OUTPUT:

- A $\mathbb{Z}$-basis $B = [b_1, \ldots b_k]$ (with $b_i \in \mathcal{M}_n(\mathbb{Q})$) of a maximal order O of E.
- The Schur index s of E.
- The multiplicity m.

Given the output of the algorithm, we always have $E \cong \mathcal{M}_m(D)$, where D is a division algebra, $F = Z(D)$ is a field and $\mathrm{Dim}_F(D) = s^2$.

Remarks 2.3.5. If $E = \mathrm{End}_A(M)$ where M is a homogeneous A-module (which is always the case in our applications), then the returned m gives the multiplicity such that $M \cong \oplus_{i=1}^m S$ for simple S, so we recognize that M is simple if and only if $m = 1$.

Let E be a $\mathbb{Q}$-algebra. Call a non-zero element $a \in E$ a **split element** if the minimal polynomial f of a has at least two distinct irreducible factors. In such a case, if g is a factor of f with $g \neq 1, f$, then $b = g(a)$ must be singular (a zero divisor). If E is the endomorphism ring of some A-module M, then the kernel of b gives a proper non-zero submodule of M. The main technique to split a homogeneous module M is to find a split element in the endomorphism ring of M.

2.3.2. Splitting via Maximal Order Basis Search. Suppose that M is a homogeneous A-module which is not simple, where A is a $\mathbb{Q}$-algebra. Plesken & Souvignier [PS96, 6(i)] suggested that one could split M by searching for split elements in a LLL-reduced basis of the saturation $E \subseteq \mathcal{M}_n(\mathbb{Z})$ of the endomorphism ring of M. While this works very often for cases where the dimension of E is small, it often fails when the dimension is larger. Souvignier later proposed [Sou09] to search for split elements in a maximal order O of E. Since the elements of a maximal order O are integral and a reduced basis of O goes 'deeper' into the structure of E, there is generally a much better chance of finding split elements via O than via E. Souvignier described an algorithm to split M by using a LLL-reduced basis of O w.r.t. the trace product form, but we have found that this does not work very well in higher dimensions. After much experimentation, we have found that the best method is first to compute a LLL-reduced basis B of O (using the standard coordinates, so not with a trace-based form) and then try the following in order:

1. See if any element of B is a split element;
2. See if a sum or product of basis elements of B is a split element;

Using these ideas steps alone, we tend to find a split element fairly quickly for any algebra E where the dimension z of the centre is at most 10, so this works very quickly in practice in nearly all situations which we encounter in this thesis. If this fails, then we successively perturb the basis B search for a split element in each new basis. The full algorithm to do all this is as follows.

```
Algorithm MaximalOrderBasisSearch($B, T$)
Input:
  • A basis $B = [b_1, \dots, b_k]$ of a $\mathbb{Z}$-algebra with $b_i \in \mathcal{M}_n(\mathbb{Q})$ (typically, the algebra is a
    maximal order).
  • A parameter $T$ (number of tries for search loop); may be $\infty$.
Output:
  • A split element of the $\mathbb{Z}$-algebra generated by $B$, or 'Fail' if one cannot be found.
Steps:
  1. For $i = 1, \dots k$ do: if $b_i$ is a split element then return $b_i$.
  2. For $i, j = 1, \dots k$ do:
     {
        If $b_i \cdot b_j$ is a split element then return $b_i \cdot b_j$.
        If $b_i + b_j$ is a split element then return $b_i + b_j$.
     }
  3. For $c := 1$ to $T$ do:
     {
        Set $[i_1, \dots, i_k]$ and $[j_1, \dots, j_k]$ to random integers (not necessarily distinct)
           in the range $[1 \dots k]$.
        Set $L := [b_{i_1} \cdot b_{j_1}, \dots, b_{i_k} \cdot b_{j_k}, b_1, \dots, b_k]$.
        Set $[r_1, \dots, r_k] := \mathrm{LLL}(L)$.
        For $i := 1$ to $k$ do: if $r_i$ is a split element then return $r_i$.
     }
  4. Return 'Fail'
```

Remarks 2.3.6. The call to LLL in Step 3 will use the MLLL algorithm (see Subsec.1.8.1) because of the dependencies. The initial k vectors will present a different basis for a suborder which the LLL will act on, and adding the basis for O after that ensures that the reduced basis is another basis of O. In general, the new basis can be quite different because of the initial vectors coming from the products. So the heuristic idea is that this perturbed basis hopefully has quite a different structure and so there is a chance that split elements will 'pop out' of the new basis.

Example 2.3.7. Here is a very small example where the use of a maximal order provides a splitting of a homogeneous module. We let M be the dimension-4 A-module, where A is a $\mathbb{Q}$-algebra with action on M given by these 2 generators:

$$a_1 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & -13 & 0 & 0 \\ 13 & 13 & 0 & 0 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 13 & 0 & 0 & 0 \\ 0 & 13 & 0 & 0 \end{pmatrix}.$$

M is a condensed module arising from the construction of a degree-14 irreducible rational representation of $\mathrm{L}_2(13)$. Now the endomorphism ring E of M has dimension 4 and a

LLL-reduced basis of the saturation of E is:

$$e_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, e_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{pmatrix},$$

$$e_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 \\ 13 & 0 & 0 & 0 \\ -13 & -13 & 0 & 0 \end{pmatrix}, e_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 13 & 0 & 0 \\ 13 & 0 & 0 & 0 \end{pmatrix}$$

The minimal polynomials of e_2, e_3 and e_4 are x^2+x+1, x^2-13 and x^2-13, respectively. Since these are irreducible over $\mathbb{Q}$, the reduced basis elements do not split M. But if we compute a maximal order O of E, then a LLL-reduced basis of O is:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{pmatrix},$$

$$\begin{pmatrix} \frac{1}{3} & -\frac{1}{3} & \frac{2}{39} & \frac{7}{39} \\ \frac{1}{3} & \frac{2}{3} & \frac{5}{39} & -\frac{2}{39} \\ \frac{2}{3} & \frac{7}{3} & \frac{1}{3} & -\frac{1}{3} \\ \frac{5}{3} & -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} \end{pmatrix}, \begin{pmatrix} -\frac{1}{3} & \frac{1}{3} & -\frac{5}{39} & \frac{2}{39} \\ -\frac{1}{3} & -\frac{2}{3} & \frac{7}{39} & \frac{5}{39} \\ -\frac{5}{3} & \frac{2}{3} & -\frac{1}{3} & \frac{1}{3} \\ \frac{7}{3} & \frac{5}{3} & -\frac{1}{3} & -\frac{2}{3} \end{pmatrix}$$

The last 2 matrices both have minimal polynomial $x(x-1)$, so are split elements. LLL-reduced basis matrices of the kernels over $\mathbb{Z}$ for the first element are:

$$\begin{pmatrix} -1 & 3 & -1 & 0 \\ 4 & 1 & 0 & -1 \end{pmatrix}, \begin{pmatrix} -1 & 3 & 0 & -1 \\ 4 & 1 & 1 & 1 \end{pmatrix}$$

thus yielding a decomposition of the original module into simple components.

2.3.3. Splitting Via Solving a Conic. Suppose $M \cong S \oplus S$ for a simple A-module S, where A is a $\mathbb{Q}$-algebra, and suppose that $E = \mathrm{End}_A(M)$ has Schur index 1, so $s = 1$ and $m = 2$ in the notation of Lem. 2.3.1. Then E is isomorphic to $\mathcal{M}_2(F)$, where F is a number field of degree d, and E is a quaternion algebra. When d is very large, the above heuristic search using the maximal order may take a very long time, so we present another approach here to split M.

Plesken & Souvignier [PS96, 6(i)] presented a method for splitting M in this situation by solving a norm equation in a quadratic extension of F. We present an alternative method here which involves finding a rational point on a conic over F. This method is equivalent to the above method in the worst case, but is often much more efficient in practice because we can apply several heuristics.

Let F be a field. A **conic** C over F is a plane algebraic curve which can be defined by a bivariate polynomial $f \in F[x,y]$ of degree 2. The rational points of C are the set of pairs $(x_0, y_0) \in F^2$ such that $f(x_0, y_0) = 0$. MAGMA has a highly optimized algorithm, developed by S. Donnelly, to determine whether a conic C has a rational point over F and compute one if so, where F is $\mathbb{Q}$ or a number field. For the $\mathbb{Q}$ case, the algorithm is due to D. Simon [Sim05], and for the number field case, the algorithm is due to S. Donnelly (unpublished), based on Lagrange's method plus other techniques. In the worst case, the algorithm may involve solving a norm equation in a quadratic extension F_2 of F and the difficulty of this computation is affected by the size of the norms of the coefficients defining the conic. So the algorithm first reduces the conic to an equivalent one where the norms of the coefficients are reduced to have absolute value of the order of $\sqrt{D}$ where D is the discriminant of F. If D is smooth (which happens in general for the kinds of fields which we use, since they are subfields of cyclotomic fields or small-degree extensions of such) then the reduction of a and b also often leads immediately to a solution, without the need to solve a norm equation in the quadratic extension F_2.

We now present a concrete algorithm to find a singular element of the above endomorphism ring E by finding a point on a related conic.

Algorithm SPLITALGEBRABYCONIC(A)

INPUT:

- An algebra $A \subset \mathcal{M}_{4d}(\mathbb{Q})$ which is known to be isomorphic to $\mathcal{M}_2(F)$ where F is a number field of degree d. (An explicit isomorphism is not necessarily known.)

OUTPUT:

- A singular element of A.

STEPS:

1. Let Z be the centre of A and let $F = \mathbb{Q}(\alpha)$ be the number field to which Z is isomorphic. Let z be the element of Z corresponding to α under some isomorphism between Z and F (so the minimal polynomial of z over $\mathbb{Q}$ has degree d).
2. Let A_F be A considered as an F-algebra. Choose e_1, e_2, e_3 from a basis of a maximal order of A so that $B = [1, e_1, e_2, e_3]$ form an F-basis of A_F; i.e., so that
$$B_{\mathbb{Q}} = [1, z, \ldots, z^{d-1}, \quad e_1, ze_1 \ldots, z^{d-1}e_1, \quad e_2, ze_1 \ldots, z^{d-1}e_2, \quad e_3, ze_3 \ldots, z^{d-1}e_3]$$
is a $\mathbb{Q}$-basis of A.
3. Let T_0 be the kernel of the linear trace map $\mathrm{Tr} : A_F \longrightarrow F$ (so T_0 has dimension 3).
4. Choose non-zero $i \in T_0$ and then choose any non-zero j which is not a scalar multiple of i from the dimension-2 subspace $\{j : j \in T_0, ij + ji = 0\}$ of T_0.
5. Set $a := i^2, b := j^2$ (so $a, b \in F$ since i, j have trace 0) and $k := ij = -ji$ so A_F is explicitly recognized as a quaternion algebra A_Q with basis $[1, i, j, k]$.
6. Let C be the conic $f(x, y) = 0$, where $f(x, y) = x^2 + (b/a)y^2 + b \in F[x, y]$. Let $(x_0, y_0) \in F^2$ be a rational point on C.
7. Set $s := x_0 i + y_0 j + k \in A_Q$. ($s$ has norm 0 in A_Q so is a zero divisor.)
8. Let a be the element of A corresponding to s by writing an element of A_Q in terms of the basis B from Step 2, and then expanding in terms of the basis $B_{\mathbb{Q}}$.
9. Return a.

Proposition 2.3.8. SPLITALGEBRABYCONIC *is correct.*

Proof. The correctness of the construction of i, j, a, b with the stated properties is easy to see: A_F is clearly recognized as a dimension-4 F-algebra via z and the basis B and trace map must have kernel of dimension 3, so it is elementary to find i, j satisfying the relevant conditions. If $(x_0, y_0) \in F^2$ is a rational point on the conic C, then $s = x_0 i + y_0 j + k$ is non-zero and

$$s^2 = (x_0 i + y_0 j + k)^2 = a x_0^2 + b y_0^2 + ab = 0$$

(the cross products are all zero, since $ij = -ji$, $ik = -ki$, $jk = -kj$). So a solution to the conic clearly yields a non-zero singular element $s \in A_F$ of trace 0. Since A is known to be isomorphic to $\mathcal{M}_2(F)$, the minimal polynomial of any element has degree at most 2 and A also contains singular elements of trace 0 (i.e., having minimal polynomial of the form $x^2 + c$ for $c \in F$), so there must be always a solution to the conic. The final element a obviously corresponds to s by the isomorphisms underlying the chosen bases, so a is singular. □

There are a few optimizations to our algorithm based on the conic solution algorithm, which are often very effective:

1. If $a = -c^2$ for $c \in F$, then $(0, c)$ is easily seen to be a solution. Similarly, if $b = -c^2$ for $c \in F$, then $(c, 0)$ is a solution. So we can first check whether $-a$ or $-b$ are squares in F.
2. For each subfield S of F (starting with $\mathbb{Q}$ and then proceeding by increasing degree), we test whether a and b lie in S; if so, then we attempt to solve the conic C over S (instead of F) and if there is a solution then we can just immediately lift it to F. Solving the conic over the subfield is dramatically easier in general, so this simple test is well worth trying.
3. If $a = c_a \alpha^i, b = c_b \alpha^j$ for $c_a, c_b \in \mathbb{Q}$, and α^i, α^j are squares in F (which obviously will be the case if i and j are even, but may be true too if either are odd), then we may replace a with $c_a \sqrt{\alpha^i}$ and b with $c_b \sqrt{\alpha^j}$ and scale the final result appropriately. This case arises very often for the applications we have here, since the basis of the maximal order is often sparse and a, b often have this form.
4. We can choose each of e_1, e_2, e_3 in Step 2 to be from the basis of the maximal order O for several different choices. Since the basis is often sparse, the corresponding a and b are often small for some choice or satisfy the conditions for at least one of the above optimizations.

Combining all these optimizations yields a method which is very often much better than just solving a norm equation in a quadratic extension field.

Example 2.3.9. Let G equal the third small group of order 240, according to the classification of [BEO01] (created by `SmallGroup(240, 3)` in MAGMA). Then G has an irreducible rational representation ρ_{32} of degree 32 which is difficult to compute. The representation occurs with multiplicity 2 in either the induction to G of a degree-8 representation of an index 8 subgroup of G, or the tensor product of two degree-8 irreducible representations of G. This degree-64 representation $\rho_{64} \cong \rho_{32} \oplus \rho_{32}$ is very easy to construct in a second or so (by methods described later) and is sparse, but is difficult to split. We use the above algorithm to do this. Let E be the endomorphism ring of ρ_{64}. The centre of E is isomorphic to the degree-16 number field $F = \mathbb{Q}(\alpha)$, where the minimal polynomial of α is

$$x^{16} + 29x^{12} + 246x^8 + 524x^4 + 1.$$

We apply SplitAlgebraByConic to E; the corresponding conic C is defined by $f(x, y) = x^2 + c_1y^2 + c_2 \in F[x, y]$ where:

$$c_1 = \frac{1}{1653}(15\alpha^{13} + 475\alpha^9 + 4222\alpha^5 + 7915\alpha),$$

$$c_2 = \frac{1}{1653}(-77227687\alpha^{13} - 1252610055\alpha^9 - 3222152922\alpha^5 - 6150095\alpha).$$

A rational point on C is found in 126s by Donnelly's algorithm and is:

$$(1/1653(8905\alpha^{15} + 21368\alpha^{14} + 28989\alpha^{13} + 30316\alpha^{12} + 164996\alpha^{11} + 373787\alpha^{10} + 430901\alpha^9 + \\ 401660\alpha^8 + 451598\alpha^7 + 998039\alpha^6 + 980348\alpha^5 + 1116410\alpha^4 + 862\alpha^3 + 1905\alpha^2 + \\ 1871\alpha + 2131), 1/1653(-9315\alpha^{15} - 36446\alpha^{14} - 32131\alpha^{13} - 157776\alpha^{11} - 592287\alpha^{10} - \\ 443208\alpha^7 - 1427751\alpha^6 - 1445442\alpha^5 - 846\alpha^3 - 2725\alpha^2 - 2759\alpha))$$

We can then instantly compute the corresponding endomorphism in E which has rank 32 (and only $0, \pm\frac{1}{2} \pm 1, \pm 2$ as entries) and from this the desired irreducible rational representation ρ_{32} of G of degree 32 (which is integral and has absolute maximum entry 3).

We have performed similar splittings for most of the hard cases (where there is a very large centre) which occur when constructing all irreducible rational representations of any group having up to order 500. The results have been stored in a database. At the time of writing, there are only a small number of holes (where the centre dimension is above 20).

2.3.4. Splitting via Fieker's Minimal Field Algorithm. In [Fie09], C. Fieker presents an algorithm which, given:

- an ordinary representation $\rho_0 : G \to \mathrm{GL}_n(F_0)$ affording an absolutely irreducible character χ for a number field F_0,
- another number field F,

returns an equivalent representation $\rho : G \to \mathrm{GL}_n(F')$ affording χ such that F' is a minimal extension of F for χ. The algorithm involves splitting a cocycle in the Brauer group of the character field and has been implemented by Fieker in Magma (function `WriteGModuleOverExtensionOf`) and uses the package for cohomology computations implemented by D.F. Holt. The algorithm can also be generalized to simple A-modules, where A is a semisimple algebra over a number field [Fie11].

One practical limitation is that the algorithm makes no attempt to control the quality of the coefficients in the output, and can be very slow when the degree of the representation is not very high. We thus avoid calling it if at all possible. But the algorithm can be more effective than other methods when the degree of the field F_0 is large, so we use it sometimes. When we do use it, we also try to improve the resulting entries by techniques explained later in the thesis. The algorithm can be applied to split homogeneous rational modules as follows.

Algorithm SplitHomogeneousByMinimalField(M)
Input:

- An homogeneous $\mathbb{Q}G$-module M.

Output:

- Simple submodules $[S_1, \ldots, S_m]$ of M such that $M = \oplus_{i=1}^{m} S_i$, and the S_i are all isomorphic.

Steps:

1. Set $E :=$ EndomorphismRing(M).
   Set $B, s, m :=$ MaximalOrder(E).
   If $m = 1$ then return $[M]$.
2. Let z be the dimension of the centre of E.
   Let e be an element of E which generates a subfield of E of degree msz over $\mathbb{Q}$ and let $F = \mathbb{Q}(\alpha)$ be the number field isomorphic to this subfield (under the isomorphism $\alpha \mapsto e$).
   Let S_F be the submodule of M^F generated by the α-eigenspace of e over F and let ρ_F be the representation corresponding to S_F (which is absolutely irreducible).
3. Set $\rho_{F'}$ to a representation equivalent to ρ_F, but written over a minimal extension field F' of $\mathbb{Q}$.
   Let $\rho_{\mathbb{Q}}$ be the restriction of scalars of $\rho_{F'}$ from F' to $\mathbb{Q}$ (as in Prop. 1.6.2).
   Let S be the $\mathbb{Q}G$-module corresponding to $\rho_{\mathbb{Q}}$.
4. Now S is an irreducible constituent of M. Compute $H = \mathrm{Hom}_{\mathbb{Q}G}(S, M)$ and compute submodules $S_1, \ldots, S_m$ of M which give a direct sum decomposition of M from images of suitable elements taken from a basis of H.

Proposition 2.3.10. *Algorithm* SplitHomogeneousByMinimalField *is correct.*

Proof. Since M is homogeneous, the character of M equals $m\chi$ for some $\chi \in \mathrm{Irr}_{\mathbb{Q}}(G)$. In Step 1, m is determined and if $m = 1$, then M is simple so the returned value is correct. In Step 2, by Cor. 1.5.5 there exists a maximal subfield of E (isomorphic to $F = \mathbb{Q}(\alpha)$) having degree ms over the centre of E, or degree msz over $\mathbb{Q}$, and the representation ρ_F derived from the α-eigenspace of e is absolutely irreducible and is a constituent of the representation corresponding to M. Thus Fieker's algorithm may be applied in Step 3 to obtain an equivalent representation $\rho_{F'}$ over a minimal field F'. By Prop. 1.6.2, the restriction to scalars representation $\rho_{\mathbb{Q}}$ is irreducible over $\mathbb{Q}$ and its character must equal χ, so the corresponding $\mathbb{Q}G$-module is an irreducible constituent of M. In Step 4, a suitable subset of a basis of the Hom-module H must always yield a full decomposition of M, since M is homogeneous and S is an irreducible constituent. □

Remarks 2.3.11. Since the output of Fieker's algorithm usually does not have small entries, we have usually applied the entry reduction algorithm for rational representations (p. 25) to the output whenever we have used this method. One can also use the algorithm SplitByEigenspace below (p. 74) to compute the submodule S_F in Step 2.

2.3.5. The Complete Split-Homogeneous Algorithm. We can now combine all of the above algorithms to obtain the following algorithm to split a homogeneous module into simple components.

Algorithm SplitHomogeneous(M)

Input:

- An homogeneous A-module M where A is a subalgebra of $\mathcal{M}_n(\mathbb{Q})$.

Output:

- Simple submodules $[S_1, \ldots, S_m]$ of M such that $M = \oplus_{i=1}^m S_i$, and the S_i are all isomorphic.

Steps:

1. Set $E :=$ EndomorphismRing(M).
2. For each e_i in LLL-basis of Saturation(E) do: if e_i is a split element then let S_1, S_2 be the submodules of M generated by the relevant kernels and return the concatenation of SplitHomogeneous(S_1) and SplitHomogeneous(S_2).
3. Set $[b_1, \ldots, b_k], s, m :=$ MaximalOrder(E).
   If $m = 1$ then return $[M]$.
4. Set $T := 2k$. [Default value; can be any other value.]
   Set $e :=$ MaximalOrderBasisSearch($[b_1, \ldots, b_k], T$).
5. If $e =$ `'Fail'` and $s = 1$ and $m = 2$ then set $e :=$ SplitAlgebraByConic(E).
6. If $e =$ `'Fail'` and M is an FG-module then return SplitHomogeneousByMinimalField(M).
7. If $e =$ `'Fail'` then set $e :=$ MaximalOrderBasisSearch($[b_1, \ldots, b_k], \infty$).
8. Let S_1, S_2 be the submodules of M generated by the relevant kernels of e and return the concatenation of SplitHomogeneous(S_1) and SplitHomogeneous(S_2).

Proposition 2.3.12. *Algorithm* SplitHomogeneous *is correct.*

Proof. The correctness essentially follows from the correctness of the previous algorithms (Prop. 2.3.8 and Prop. 2.3.10). □

2.4. The Rational Meataxe

We can now present the main rational Meataxe algorithm to decompose a semisimple A-module M into a direct sum of simple components, where A is a subalgebra of $\mathcal{M}_n(\mathbb{Q})$.

The algorithm includes a very important option which will be used in the next chapter: the caller can request that only one particular simple component S of M is desired. This is specified by giving special information about the trace of the action of A on S, and is denoted by an argument `TraceInfo` $= \langle T_S, m_S \rangle$. In this case, A has k generators and we assume that the first generator is the identity element of A. Then the i-th component of $T_S \in \mathbb{Z}^k$ gives the trace of the i-th generator of A acting on S (and $T[1]$ thus gives the dimension of S), while m_S is the multiplicity of S as a constituent of M. The algorithm assumes that if the vector of traces for a submodule W of M equals $m_S \cdot T_S$, then W is isomorphic to m_S copies of S. Note that it is not compulsory in this option for the algorithm to return the desired constituent S alone, but if it can find a constituent matching the above trace information, then it returns it alone.

After giving the formal algorithm, we note several points on how to make it efficient.

```
Algorithm RationalMeataxe($M$[, TraceInfo])
Input:
  • A semisimple $A$-module $M$ where $A$ is a subalgebra of $\mathcal{M}_n(\mathbb{Q})$.
  • [Optional:] TraceInfo $\langle T_S, m_S \rangle$, where $T_S \in \mathbb{Z}^k$ and $m_S \in \mathbb{Z}^{>0}$, giving trace information for a desired simple constituent of $M$ (see above for details).
Output:
  • If TraceInfo is given, and if a simple submodule $S_T$ of $M$ corresponding to the above information is found, then $[S_T]$ is returned.
  • Otherwise, simple submodules $[S_1, \ldots, S_m]$ of $M$ are returned, such that $M = \oplus_{i=1}^{m} S_i$.
Steps:
  1. [Optional: Use Characters]
     If $M$ is a $\mathbb{Q}G$-module, and such that the character table of $G$ is known or easy to compute (say, if $|G| \leq 10^{10}$), then do the following (otherwise skip to the next step).
     Compute $\mathrm{Irr}_{\mathbb{Q}}(G)$ (via the algorithm in Thm. 1.3.4).
     Let $\chi$ be the character of $M$.
     If $\chi = \chi_i$ for some $\chi_i \in \mathrm{Irr}_{\mathbb{Q}}(G)$, then return $[M]$.
     If $\chi = m\chi_i$ for some $\chi_i \in \mathrm{Irr}_{\mathbb{Q}}(G)$ and $m > 1$, then set $L := [M]$ and go to Step 4.
  2. [TraceInfo option: try to find component]
     Let $a$ be an element of $A$ obtained by linear combinations with small random coefficients (and possibly a few multiplications). Set $f$ to the minimal polynomial of $a$.
     Factorize $f$ as $\prod_{i=1}^{s} g_i^{e_i}$ for irreducible $g_i \in \mathbb{Q}[x]$.
     Set $D := m_S \cdot T_S[1]$ [desired homogeneous dimension].
     Sort $[\langle g_1, e_1 \rangle, \ldots, \langle g_k, e_k \rangle]$, so that a pair $\langle g_i, e_i \rangle$ with $d_i = e_i \cdot \mathrm{Deg}(g_i)$ dividing $D$ comes first, and otherwise a pair with smaller $d_i$ comes first.
     If for some $i$ with $1 \leq i \leq s$, the nullspace of $(g_i^{e_i})(a)$ is invariant under $A$
        and the trace vector $[t_1, \ldots, t_k]$ of the generators of $A$ acting on the
        corresponding submodule $W_i$ equals $m_S \cdot T_S$ then:
     {
        [Found desired constituent. Return it immediately if simple.]
        Set $L := [W_i]$.
        If $m_S = 1$, then return $L$; otherwise go to step 4.
     }
     [Trace-based search failed. Fall through to full splitting.]
```

```
    If the nullspace of $(g_i^{e_i})(a)$ is invariant under $A$ for all $1 \leq i \leq s$ then:
    {
        Let $V_i$ be the submodule of $M$ generated by the $i$-th nullspace.
        Set $L := [V_1, \ldots, V_s]$.
    }
    Else:
        Set $L := [M]$.
3.  [Split into Homogeneous Components]
    Set $L$ to the concatenation of [HomogeneousComponents($S$) : $S \in L$].
4.  [Split Homogeneous Components]
    Set $L$ to the concatenation of [SplitHomogeneous($S$) : $S \in L$] and return $L$.
```

Theorem 2.4.1. *Algorithm* RationalMeataxe *is correct.*

Proof. If Step 1 is applied, then if $\chi = \chi_i$ for some i, then obviously M is simple so the step is correct in returning $[M]$ immediately, while if $\chi = m\chi_i$, $m > 1$, then M is homogeneous, so it is valid to jump to Step 4.

Step 2 is only used in the `TraceInfo` case. If $U_1, \ldots, U_s$ are the generalized eigenspaces corresponding to the maximal powers of the irreducible factors of the minimal polynomial of a, then the U_i obviously give a direct sum decomposition of the underlying vector space F^n. Now the algorithm only needs to return a submodule which matches the trace information given by T_S; if such a submodule W_i is found (matching one of the eigenspaces) then either the multiplicity m_S is one 1 and W_i may be returned immediately or W_i is a homogeneous module isomorphic to $m_S \cdot S$ for simple S and Step 4 only needs to decompose W_i. If the trace test fails, then L is clearly always set to some decomposition of M.

In Step 3, by the correctness of HomogeneousComponents above, clearly each submodule of L is split into homogeneous components and in Step 4, by the correctness of SplitHomogeneous above, the homogeneous components of each component are split into simple submodules. Thus in the general case, a direct sum decomposition of M into simple components is returned, while in the `TraceInfo` case, the components returned will be simple and one of the components will match the trace information. □

Remarks 2.4.2. We note the following points on the implementation:

1. The use of the character in Step 1 when applicable is very effective in practice, since it predicts exactly the decomposition of M. For example, M may be proven to be irreducible very quickly, and this saves a lot of time when the dimension of M is large. Note however that in the rest of the thesis we will mostly apply the Meataxe to modules which are not $\mathbb{Q}G$-modules (the main exception is in the setting up the condensation of tensor modules below). One can also use the character information in the subsequent steps (e.g., to determine that a component is irreducible after an initial splitting).
2. The `TraceInfo` option will be used in the next chapter to extract a simple constituent of a condensed module. In this situation, M often has very many simple components, so that is why we first obtain a homogeneous splitting by an algebra element (like the 'traditional' Meataxe), instead of computing the endomorphism ring or its centre, both of which may be very large so much more expensive to compute. If the desired

component is not found, then the step will be a waste of time, but we have found that it practically always works in finding a direct sum decomposition so is well worth doing in practice. The nullspace of each $h_i = (g_i^{e_i})(a)$ in Step 2, where h_i is a divisor of the minimal polynomial of A (with maximal multiplicity), can be computed by the method discussed in Subsec. 1.7.4 so this can be done very efficiently, even when the degree of g is large. Since our modules are assumed to be semisimple, this method always works well in practice.

3. The `TraceInfo` option will be used heavily in the next chapter when M is a condensed module, and it will often be the case that M may have a large dimension, but we only wish to compute a single small constituent of M, and the trace information will allow us to identify this constituent uniquely. In this case, the computation of the minimal polynomial f of a and then the nullspace of the evaluation of a single small-degree factor of f at a is very fast and so the whole algorithm takes very little time.

   For simplicity of exposition, we have presented the algorithm so that in this option, only a single constituent is desired. But in the implementation, the algorithm allows the trace information for several constituents to be given, so that corresponding simple submodules are extracted. This avoids multiple calls of the Meataxe when multiple constituents are desired from one condensed module when condensation is used (see the next chapter).

The next chapter will give several examples of the use of the rational Meataxe, particularly in the case where the option with the trace information is used.

2.5. A Simplicity Test

We give here a practical algorithm to decide whether a semisimple A-module, for a subalgebra A of $\mathcal{M}_n(\mathbb{Q})$ is simple. This algorithm is a simplification of the more general Meataxe algorithm above and is not needed separately in subsequent algorithms, but is included here for completeness and to summarize all the techniques which can be used to prove simplicity.

Algorithm IsSimple(M)
Input: A semisimple A-module M, where A is a subalgebra of $\mathcal{M}_n(\mathbb{Q})$.
Output: A boolean flag indicating whether M is simple.
Steps:

1. [*Optional: Character Test*] If M is a $\mathbb{Q}G$-module, and such that the character table of G is known or easy to compute (say, if $|G| \leq 10^{10}$), then compute $\mathrm{Irr}_{\mathbb{Q}}(G)$ (via the algorithm in Thm. 1.3.4) and the character χ of M and then return whether $\chi = \chi_i$ for some $\chi_i \in \mathrm{Irr}_{\mathbb{Q}}(G)$.
2. [*Optional: Modular Test*] Test whether $M \bmod p$ is irreducible for some prime p; if so, return *true*. (*Occasionally works, but useless when there is a non-trivial Schur index.*)
3. [*Optional: Try Meataxe-type Split for Highly Decomposable Case*] Choose element $a \in A$ from a small random linear combination of the generators of A and if a generalized eigenspace of a generates a proper submodule of M, then return *false*.

4. [*Endomorphism Ring Centre Test*] Set $Z :=$ SATURATION(Centre($\mathrm{End}_A(M)$)). For each e in a LLL-reduced $\mathbb{Z}$-basis of Z do: if e is a split element, then return *false*.
5. [*Endomorphism Ring Test*] M is now homogeneous. Compute $E = \mathrm{End}_A(M)$. If $E = Z$, return *true*. Otherwise, for each e in a LLL-reduced basis of the saturation of E: if e is a split element, then return *false*.
6. [*Maximal Order Test*] Set $B, s, m :=$ MAXIMALORDER(E) and then return whether $m = 1$.

Theorem 2.5.1. *Algorithm* ISSIMPLE *is correct.*

Proof. Step 1 is correct because $\mathrm{Irr}_{\mathbb{Q}}(G)$ is exactly the set of characters of irreducible $\mathbb{Q}$-representations of G (Def. 1.3.1). Step 2 is correct, since if M is not simple, then it must be not simple mod p too. Steps 3 to 5 are clearly correct if they return *true* (a proper submodule is found). If Step 5 is reached, M must be homogeneous and E is a central simple algebra. The algorithm MAXIMALORDER determines the multiplicity m and thus M is simple if and only if $m = 1$. □

CHAPTER 3

Constructing Irreducible Representations Via Condensation

3.1. Introduction

In this chapter we describe efficient algorithms for the splitting approach for computing irreducible representations. The basic idea is to extract these as constituents of a potentially large-degree representation, using condensation in an automatic way. There has been extensive use of condensation in constructing modular representations of finite groups, but there has apparently hitherto been hardly any use of condensation in characteristic zero. We develop a key automatic algorithm which uses an algorithm for finding non-negative solutions of an integral linear system to choose a suitable condensed module so that the desired irreducible representations can be constructed efficiently.

3.2. Non-negative Solutions to Integral Linear Systems

In this section we describe an algorithm to solve the following important problem. Suppose that we are given vectors $[v_1, \ldots, v_k]$, w, all in $\mathbb{Z}^n$ and such that the first coordinate of each vector is strictly positive. Let V be the $k \times n$ matrix whose rows are $[v_1, \ldots, v_k]$. We wish to find all solutions in s to the linear system given by:

$$s \cdot V = w,$$

such that the entries of s are **all non-negative**.

The motivation for solving this problem is clear when we consider the characters of rational representations of finite groups. A rational character has integral entries and the first value is always positive (the degree of the character). If the v_i are the irreducible rational characters of a group G and w is an arbitrary rational character of G, then w can be written uniquely as a non-negative linear combination of the v_i. In this case, $n \geq k$ and the above matrix V has rank k, so the solution over $\mathbb{Z}$ is unique (and has non-negative coordinates), so the problem can easily be solved by standard linear algebra. But there are two more general situations which we will encounter:

1. We may only have partial characters; i.e., the v_i and w may have character values only for a proper subset of the full list of conjugacy classes, in which case we may have $n < k$ and then the rank of the corresponding matrix V will be less than k, so there may not be a unique solution for s and it may be hard to find a non-negative solution.
2. We will also need to solve this problem for vectors of traces of elements of a condensed algebra (in Sec. 3.7 below); again, the rank of the corresponding V matrix may be less than k, so it may be difficult to find non-negative solutions.

This problem is clearly related to the well-known Knapsack (or subset-sum) problem. There are well-known methods to solve this restricted problem, such as those based on the

LLL algorithm (see [SE91] for example). We develop our own simple heuristic algorithm here since it seems to work very effectively for the kinds of inputs we apply it to, and we can take advantage of the special condition that the first coordinate of every vector must be strictly positive. The basic idea is to determine bounds $B_1, \dots, B_k$ for each coordinate of a solution vector s, and then do a standard recursive search, using the bounds for each coordinate. Since the first coordinate of every vector in the input is positive, each B_i can be initialized to a non-negative value. A naive search obviously has exponential complexity in k when the bounds are uniform. But we first we use some heuristics to reduce the bounds, and usually this reduction works well enough that the recursive search is very easy.

The first basic subalgorithm SEARCH does a simple recursive search based on the given bounds on each coordinate.

Subalgorithm SEARCH($[v_1, \dots, v_k], w, [B_1, \dots, B_k]$, `MaxSolutions`)

INPUT:

- Vectors $[v_1, \dots, v_k]$ and w, all in $\mathbb{Z}^n$ and such that the first coordinate of the v_i and w are positive.
- Bounds $[B_1, \dots, B_k]$ for each coordinate of the solutions.
- A positive integer `MaxSolutions` (may be ∞), bounding the number of solutions returned.

OUTPUT:

- All solution vectors $[s_1, \dots, s_r] \in (\mathbb{Z}^{\geq 0})^k$ such that $s_i \cdot V = w$, where V is the matrix whose rows are $[v_1, \dots, v_k]$ and the j-th coordinate of each s_i is at most B_j. If `MaxSolutions` $< \infty$, then r is limited to at most `MaxSolutions`.

STEPS:

```
1. If $w = 0$ then return $\{t\}$ where $t = (0, \dots, 0) \in \mathbb{Z}^k$.
   If $k = 0$ then return $\{\}$.
2. Set $m$ to the minimum of $B_k$ and $\lfloor \frac{w[1]}{v_k[1]} \rfloor$ and set $S := \{\}$.
   For $i := 0$ to $m$ do:
   {
      Set $T :=$ SEARCH($[v_1, \dots, v_{k-1}], w - iv_k, [B_1, \dots, B_{k-1}]$, MaxSolutions).
      For $t$ in $T$ do:
      {
         Write $t = (c_1, \dots c_{k-1})$.
         Insert $(c_1, \dots c_{k-1}, i)$ into $S$.
         If $\#S =$ MaxSolutions then return $S$.
      }
   }
   Return $S$.
```

Lemma 3.2.1. *Subalgorithm* SEARCH *is correct.*

Proof. This algorithm is easily seen to be correct by induction. For the base case, if $w = 0$, then the zero vector is the unique solution; otherwise, if $k = 0$, then there can be no solution. In the general case, m is clearly set to an upper bound on the number of times that v_k can contribute to a sum equal to w, since the first coordinates are all positive (if the bound B_k is smaller, then it is used instead). Then SEARCH simply recurses with one less v_i vector, finds the relevant solutions, and extends each solution with the coefficient corresponding to v_k. □

We now give an improved algorithm which first attempts to reduce the bounds on the coordinates, and then calls the above subalgorithm. Let V be the matrix whose rows are $[v_1, \ldots, v_k]$, so that we wish to find the set of all solution vectors of the form $s \in (\mathbb{Z}^{\geq 0})^k$ with $sV = w$. Clearly, if T is an $n \times n$ invertible matrix over $\mathbb{Q}$, with VT and wT having integral entries, then $sV = w$ if and only if $sVT = wT$ for any $s \in (\mathbb{Z}^{\geq 0})^k$. So we can replace the original problem involving (V, w) with (VT, wT) for any such T. The advanced algorithm reduces the bounds on the coordinates by doing column operations on the associated matrix to generate equivalent systems for which there are coordinates with every coefficient positive, thus giving extra bounds. First there is a subalgorithm UPDATEBOUNDS which simply makes the bounds smaller if possible, based on matrices defining an equivalent system. Then the main algorithm NONNEGATIVESOLUTIONS calls UPDATEBOUNDS on various matrices until no more bound reduction is possible, and then calls SEARCH with the final bounds.

```
Subalgorithm UPDATEBOUNDS($[B_1, \ldots, B_k], C, A$)
INPUT: Current bounds $[B_1, \ldots, B_k]$, a positive column vector $C \in (\mathbb{Z}^{>0})^{(k+1)\times 1}$, and a
matrix $A \in \mathbb{Z}^{(k+1)\times c}$.
OUTPUT: Updated bounds $[B_1, \ldots, B_k]$ based on $C$ and $A$.
STEPS:
 1. For $j := 1$ to $c$ do:
    {
       [Update bounds by adding suitable multiple of $C$ to column $j$ of $A$.]
       Let $u$ be the $j$-th column vector of $A$.
       Let $q \in \mathbb{Z}^{>0}$ be minimal such that $u' = u + qC$ has no negative entry.
       Set $a := u'[k+1]$.
       For $i := 1$ to $k$ do:
          If $u'[i] \neq 0$ then set $B_i$ to the minimum of $B_i$ and $\lfloor \frac{a}{u'[i]} \rfloor$.
    }
 2. Return $[B_1, \ldots, B_k]$.
```

Algorithm NonNegativeSolutions($[v_1, \dots, v_k], w,$ `MaxSolutions`)

Input:

- Vectors $[v_1, \dots, v_k]$ and w, all in $\mathbb{Z}^n$ and such that the first coordinate of the v_i and w are positive.
- A positive integer `MaxSolutions` (may be ∞), bounding the number of solutions returned.

Output:

- All solution vectors $[s_1, \dots, s_r] \in (\mathbb{Z}^{\geq 0})^k$ such that $s_i \cdot V = w$, where V is the matrix whose rows are $[v_1, \dots, v_k]$.

Steps:

```
1. Let $C$ be the positive column vector in $(\mathbb{Z}^{>0})^{(k+1)\times 1}$ with $C[i] = v_i[1]$ for $1 \leq i \leq k$
   and $C[k+1] = w[1]$.
2. Let $A \in \mathbb{Z}^{(k+1)\times n}$ be the matrix whose $i$-th row is $v_i$ for $1 \leq i \leq k$ and whose $(k+1)$-th
   row is $w$.
   Set $[B_1, \dots, B_k] :=$ UpdateBounds($[\infty, \dots, \infty], C, A$).
3. Set index label $I := [1, 2, \dots, k]$.
   Loop forever:
   {
       Set $H :=$ HermiteForm(Saturation(Transpose($A^{tr}$))$^{tr}$.
       Remove all zero columns from $H$.
       Set $[B_1, \dots, B_k] :=$ UpdateBounds($[B_1, \dots, B_k], C, H$).
       Set $L :=$ LLL$(H^{tr})^{tr}$.
       Set $[B_1, \dots, B_k] :=$ UpdateBounds($[B_1, \dots, B_k], C, L$).
       If $B_i \neq 0$ for all $i$, then break out of the loop [no more reduction possible].
       For each $i$ with $B_i = 0$ do:
           Delete row $i$ of $A$ and $C$, delete $B_i$ and index $I[i]$.
   }
4. Let $[v'_1, \dots, v'_r, w']$ be the rows of $A$.
5. Set $S' :=$ Search($[v'_1, \dots, v'_r], w', [B_1, \dots, B_r]$, MaxSolutions).
6. Expand each vector $s'$ of $S'$ according to $I$ (expand $s' \in \mathbb{Z}^{n'}$ to $s \in \mathbb{Z}^n$ by mapping
   column $j$ in $s'$ to column $I[j]$ in $s$), set $S$ to the result, and return $S$.
```

Proposition 3.2.2. *Algorithm* NonNegativeSolutions *is correct.*

Proof. Let V be the matrix whose rows are $[v_1, \dots, v_k]$. As noted above, for $s \in (\mathbb{Z}^{\geq 0})^k$, we have $sV = w$ if and only if $sVT = wT$ for any $n \times n$ invertible matrix T over $\mathbb{Q}$. The initial steps of NonNegativeSolutions simply try to reduce the problem by multiplying by such invertible T to the current system (clearly the column Hermite form, saturation and LLL operations apply invertible column operations only). Each new (column equivalent) matrix is passed to subalgorithm UpdateBounds. This also effectively multiplies its input by an invertible matrix on the right (by doing column operations only): since the column vector C contains positive entries only, there must exist a q each time such that qC can be added to the j-th column vector of A to make it non-negative. Then for this column, each bound B_i is correctly updated, based on the quotients of the relevant coordinates. Whenever a bound becomes 0, then obviously the corresponding row can be removed in the loop in Step 3 of NONNEGATIVESOLUTIONS. Finally, SEARCH is applied to an equivalent system with the updated bounds, so after fixing the coordinates for the deleted rows, the output must be same as if SEARCH had been applied to the original input. □

Example 3.2.3. We give an example of a typical use of NONNEGATIVESOLUTIONS, which comes from recognizing a partial character in a soluble group of order 500, which has 12 distinct irreducible rational characters; we only use the character values on 9 classes here.

Let A be the following 17×9 integral matrix:

$$\left(\begin{array}{r|rrrrrrrr}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 \\
4 & 4 & 4 & 4 & 4 & 4 & 4 & -1 & -1 \\
4 & -4 & 4 & -4 & 4 & 4 & -4 & -1 & 1 \\
4 & 4 & -4 & -4 & 4 & 4 & 4 & -1 & -1 \\
4 & -4 & -4 & 4 & 4 & 4 & -4 & -1 & 1 \\
20 & 20 & 20 & 20 & 20 & -5 & -5 & 0 & 0 \\
20 & -20 & 20 & -20 & 20 & -5 & 5 & 0 & 0 \\
20 & 20 & -20 & -20 & 20 & -5 & -5 & 0 & 0 \\
20 & -20 & -20 & 20 & 20 & -5 & 5 & 0 & 0 \\
100 & 100 & 100 & 100 & -25 & 0 & 0 & 0 & 0 \\
100 & -100 & 100 & -100 & -25 & 0 & 0 & 0 & 0 \\
100 & 100 & -100 & -100 & -25 & 0 & 0 & 0 & 0 \\
100 & -100 & -100 & 100 & -25 & 0 & 0 & 0 & 0 \\
\hline
410 & -8 & -206 & -192 & -90 & 10 & -8 & 0 & 2
\end{array}\right)$$

Let $[v_1, \ldots, v_{16}]$ be the first 16 rows of X and w the last row of X. We call NONNEGATIVESOLUTIONS on the v_i and w, with ∞ for each initial bound. The first call to UPDATEBOUNDS on the original input gives these initial bounds:

$$[102, 102, 160, 109, 25, 25, 40, 27, 5, 5, 8, 5, 1, 1, 2, 1].$$

The next call to UPDATEBOUNDS on the saturated column-Hermite form reduces the bounds to:

$$[101, 102, 160, 108, 25, 10, 40, 10, 5, 5, 8, 5, 1, 1, 2, 1].$$

The next call to UPDATEBOUNDS on the column-LLL-reduced matrix reduces the bounds to:

$$[1, 1, 1, 1, 0, 10, 0, 10, 0, 0, 0, 0, 1, 1, 2, 1].$$

After rows 5, 7, 9, 10, 11, 12 are removed (for which the bound is now 0), the reduced bounds become:

$$[1, 1, 1, 1, 10, 10, 1, 1, 2, 1].$$

One more round of the loop (using the saturated column-Hermite form and the column-LLL-reduced matrix) reduces the bounds to:

$$[1, 1, 1, 1, 2, 2, 1, 1, 2, 1].$$

The reduced combined matrix whose rows are then passed to SEARCH is:

$$\left(\begin{array}{r|rrrrrrr}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
4 & 0 & -4 & 0 & 4 & 1 & 0 & 0 \\
100 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
100 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
100 & -100 & 0 & 100 & 0 & 0 & 1 & 0 \\
100 & 0 & -100 & 0 & 100 & 0 & 0 & 1 \\
\hline
410 & -199 & -107 & 200 & 108 & 2 & 2 & 2
\end{array}\right)$$

This subalgorithm instantly finds that there is a unique non-negative solution vector for this system. After inserting the zeros corresponding to the removed rows, we obtain the final solution:

$$\left(\begin{array}{cccccccccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 1 & 2 & 1 \end{array}\right)$$

The whole computation took less than 0.02 seconds. The SEARCH subalgorithm was entered 30 times, at all levels of recursion, so did very little work.

3.3. Computing Characters

We now describe the first important application of the algorithm of the previous section. Suppose that $\rho : G \longrightarrow \mathrm{GL}_n(F)$ is an ordinary representation and we wish to compute its character χ. There is an obvious simple algorithm which evaluates ρ at each member of a set of class representatives of G, but this of course can take a very long time in high degree since there may be many matrix multiplications needed. We show how this naive algorithm can be greatly improved.

The first obvious improvement is that we can first compute the traces of the generators of G (for which we already have the images under ρ) and then also products of the generators and random products of reasonably short length. This often covers many of the classes of G. After several trials with random products yielding nothing new, we can then revert to evaluating ρ at the missing classes (using words in strong generators to evaluate the words efficiently). We can also evaluate the character χ quicker by checking orders of elements, as follows. The default method to evaluate $\chi(g)$ for $g \in G$ involves evaluating the *class map* for G at g: this computes the relevant conjugacy class which g lies in. This can be expensive for larger groups, particularly if g lies in one of the more 'obscure' classes of the elements of higher order in G. We have implemented a simple trick which helps enormously: for fixed χ, we compute the orders $\{o_1, \ldots, o_r\}$ and corresponding character values $\{v_1, \ldots, v_r\}$ such that for any element $g \in G$ of order o_i, the character value $\chi(g) = v_i$ (i.e., the character values must be constant for elements of the specific order). Then for any $g \in G$, if g has an order o_i, then $\chi(g)$ can be computed instantly as v_i. For most characters, this covers most of the classes of G (or at least most of those with high order) and so speeds up the character evaluation greatly.

Now if $F = \mathbb{Q}$, then we can use the non-negative solutions algorithm from the previous section to speed up the algorithm greatly in most situations. We first compute $\mathrm{Irr}_{\mathbb{Q}}(G)$ and set $[\chi_1, \ldots, \chi_r] := \mathrm{Irr}_{\mathbb{Q}}(G)$. Suppose then that at any point we have computed the values of χ for class indices $j_1, \ldots, j_l$ $(1 \leq j_c \leq k)$. Let $w = (a_1, \ldots, a_l) \in \mathbb{Z}^l$ be the corresponding vector of these known values and let $v_i = (\chi_i[j_1], \ldots, \chi_i[j_l])$ be the vector of corresponding values selected from χ_i, for $1 \leq i \leq k$. Then we call NonNegativeSolutions on the v_i and w and if there is a unique solution $(s_1, \ldots, s_r)$, then we know that the character χ must equal $\sum_{i=1}^{r} s_i \chi_i$ so we can stop immediately. The call to NonNegativeSolutions can pass the value 2 for the argument `MaxSolutions`, so that if there is not a unique solution, then the search will stop very quickly, and we then continue to gather more values of the character via the methods in the previous paragraph. Each time a new character value is found, we can check whether the associated linear system now has a unique solution, but if the degree n is reasonably small (so that evaluating ρ is cheap), then we can of course wait till the number of values builds up a bit before calling NonNegativeSolutions again. This method works extremely well in practice in high degree, since it cuts down the number of matrix multiplications dramatically. It is often the case that the degree n and the traces of the images of the generators of G alone are enough to determine the character uniquely. For example, in Ex. 3.7.3 below, the character of a degree-782 rational representation of Fi_{23} is verified to be the irreducible character of degree 782 by using the traces of the images of the generators alone.

One other obvious optimization in the case that $F = \mathbb{Q}$ is the following. Since the character values of ρ are integers, they must be bounded in absolute value by the degree n [Hup98, 3.18]. So if we let p be the first prime greater than $2n$, then we may perform all the matrix operations over the finite field $\mathbb{F}_p$, using the symmetric range modulo p to recover the integral traces with correct signs. Any required matrix multiplications can thus be performed very quickly in practice via the ATLAS library and Strassen's algorithm (see p. 24), even for rather high degree (on a typical computer, the product of two such 1000×1000 matrices takes under 0.5s).

Note also that the trace of AB for matrices $A, B \in \mathcal{M}_n(R)$ can be computed much more efficiently than by simply computing $C = AB$ and then $\mathrm{Tr}(C)$. If $A = [a_{i,j}]$ and $B = [b_{i,j}]$, then

$$\mathrm{Tr}(AB) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} \cdot b_{j,i},$$

which involves $O(n^2)$ sums and products of elements of R instead of $\mathsf{MM}(n)$ sums and products (where $\mathsf{MM}(n)$ denotes the complexity of matrix multiplication). We have implemented this method in Magma (as the function `TraceOfProduct(A, B)`). The above algorithm can then be improved even more as follows: as we compute successive elements of G and their images under ρ, we can store the elements of G as $[x_1, \ldots, x_s]$ and also the corresponding image matrices $[\rho(x_1), \ldots, \rho(x_s)]$. Then whenever we consider any new $y \in G$ and corresponding $\rho(y)$ we can also check whether the class of $y \cdot x_i$ for $1 \leq i \leq s$ has not been covered, and if so, we compute $\mathrm{Tr}(\rho(y) \cdot \rho(x_i))$ (using the fast trace-of-product method) and thus have a new character value for the class of $y \cdot x_i$. This can give us several extra character values which are relatively quick to compute, avoiding matrix multiplications. In the case that $F = \mathbb{Q}$, when the traces of the initial generators are not enough to determine the character uniquely (via the associated non-negative linear system), then it is still often the case that the traces of all products of the generators give enough values so that the system does have a unique solution, thus allowing the character to be determined without computing a single matrix multiplication.

3.4. The Integral Spin Algorithm

In the standard modular Meataxe, a fundamental subalgorithm is the so-called 'spin' procedure, which computes a basis for the invariant submodule generated some vectors under the action of some algebra (typically described by explicit matrices). This is easy to implement in the modular case with elementary linear algebra, because the growth of the matrix entries is not an issue. But in the characteristic zero case, a corresponding algorithm is much more difficult to make efficient, not just because of potential entry growth during the course of the computation, but because the choice of the final invariant basis can have a dramatic effect on the size of the entries in the matrices defining the reduced action on the submodule. We now describe how to implement such an algorithm so that these problems can be overcome in practice up to rather high dimension.

The 'integral spin' algorithm presented here computes the sublattice S of $\mathbb{Z}^n$ generated by some vectors under a given linear action ϕ on $\mathbb{Z}^n$, and the relevant reduced action on a suitable reduced basis of S. There are two simple stages to the algorithm. In the first stage, an invariant basis B is computed: this is done fairly easily by doing a 'modular spin' in parallel and only keeping the integral vectors which are independent modulo p. The second stage, which is typically much more expensive, invests substantial effort to ensure that the basis of the sublattice S is as reduced as possible, since this affects the quality of the final representation.

Algorithm INTEGRALSPIN($\{v_1, \ldots, v_m\}, \phi(v, i), k$)

INPUT:

- A set of vectors $\{v_1, \ldots, v_m\}$ lying in $\mathbb{Z}^n$.
- A linear 'action' function $\phi : \mathbb{Z}^n \times \{1, \ldots, k\} \to \mathbb{Z}^n$ for a $\mathbb{Z}$-algebra A acting on $\mathbb{Z}^n$ from the right and with k generators: ϕ takes a vector $v \in \mathbb{Z}^n$ and a generator number i with $1 \leq i \leq k$ and returns the result of acting on v by the i-th generator of A.

OUTPUT:

- A matrix B which is a reduced basis for the saturated invariant sublattice of $\mathbb{Z}^n$ generated by the v_i under the action of ϕ.
- Matrices $[X_1, \ldots, X_k]$ from $\mathcal{M}_r(\mathbb{Z})$ giving the reduced action of ϕ on B.

STEPS:

```
1. Set $[w_1, \ldots, w_s]$ to a basis of the lattice spanned by $[v_1, \ldots, v_m]$.
   Choose a prime $p$ such that $[w_1, \ldots, w_s]$ are independent modulo $p$.
   Set $r := s$ and $l := 1$. [$r$ is the current rank; $l$ is the next vector to process.]
   While $l \leq r$ and $r \leq n$ do:
   {
      For $i := 1$ to $k$ do:
        {
            Set $u := \phi(w_l, i)$.
            If $[w_1, \ldots, w_r, u]$ is independent mod $p$ then:
            {
                Set $r := r + 1$.
                Set $w_{r+1} := u$.
            }
        }
        Set $l := l + 1$.
    }
```

2. Set B to the matrix in $\mathcal{M}_{r \times n}(\mathbb{Z})$ whose rows are $[w_1, \ldots, w_r]$.
   Set $B :=$ SATURATION(B). [See p. 20.]
   Set $B :=$ HERMITEFORM(B). [See p. 18.]
   Set $B :=$ LLL(B). [See p. 21.]
   Set $B :=$ SEYSEN(B). [See p. 22.]
3. For $1 \leq i \leq k$, attempt to set $X_i \in \mathcal{M}_r(\mathbb{Z})$ to the solution of the matrix equation
   $$X_i \cdot B = \phi(B, i).$$
   If there is no solution for some i, then return to Step 2 and choose a new prime different to those chosen before (this p must have been bad).
4. Return B and $[X_1, \ldots, X_k]$.

Proposition 3.4.1. *Algorithm* INTEGRALSPIN *is correct.*

Proof. Step 1 clearly computes a basis $[w_1, \ldots, w_r]$ for a sublattice of $\mathbb{Z}^n$ which is invariant under ϕ modulo p and Step 2 does not change the $\mathbb{Q}$-span of the basis. If the basis is not invariant under ϕ (over $\mathbb{Q}$), then Step 3 will fail so the computation will be restarted with a new prime. There can only be a finite number of primes for which there is failure (they must divide the elementary divisors of the correct saturated invariant lattice). When Step 6 succeeds, B must describe an invariant lattice and must have minimal rank for an invariant lattice containing $[v_1, \ldots, v_m]$ (since it is such modulo p). Finally, the reduced action matrices $[X_1, \ldots, X_k]$ must be integral, by Prop. 1.7.11, since the lattice spanned by the rows of B is saturated. □

Remarks 3.4.2. We note the following points on our implementation:

1. The prime p should be chosen just as in the modular algorithm for computing a Hom-module (see p. 24), so the probability of hitting a bad prime is typically very low in practice.
2. There are 4 types of linear action which we will use below in practice. The simplest one is of course multiplication by an $n \times n$ matrix on the right, while the other 3 types of action are *permutation*, *induction* and *tensor*. We will apply each of these kinds in the context of condensation, and will explain the specific actions as they arise below. We just note here that the permutation action on a vector simply permutes its coordinates; obviously this is a lot faster than a general matrix action and needs very little memory

to store the action, and so is very efficient even for the high-degree permutation actions which will occur. The coefficients of the resulting vector are the same, too, so there is no growth of the entries at all in Step 1.

3. In Step 2, even if the initial matrix B has very small entries, the saturation algorithm may produce a matrix with some large entries: typically, the last row will have rather larger entries (possibly with many digits) than the other rows. Now applying the LLL algorithm directly to the new saturated basis B can be extremely slow: the upper rows with small entries will usually be reduced quickly, but the lower rows may involve an extremely large number of steps to reduce. Instead, we compute the Hermite form H of the saturated basis matrix first, and then apply LLL to H. In this case, the LLL algorithm always seems to work with a uniform number of steps to reduce each input row (include the final ones) and it does not slow down dramatically for the final rows. Even though H will often have very large entries, this method is practically always much faster and yields a basis with small entries (as small as the original non-saturated invariant basis).

   Note also that the modular Hermite algorithm (p. 18) is typically very fast when the elementary divisors of the input matrix are trivial but can be much slower when they are non-trivial. As a result, for the operations performed in Step 2, the call to SATURATION is usually the most expensive, since it calls the Hermite form algorithm on a matrix which typically has non-trivial elementary divisors (coming from the initial invariant basis), while the subsequent HERMITEFORM call is usually quite fast. This behaviour will be seen in examples below.

4. Each matrix equation computation in Step 3 is done by a CRT-based modular algorithm (solving the matrix equation over $\mathbb{Q}$ and then checking that the solution is integral), which is fast, so this step is always relatively quick (and is certainly much easier than computing the reduced basis).

5. This algorithm is useful when the input is already a basis of a submodule over $\mathbb{Q}$ under the action, since the algorithm will find a reduced basis so that the resulting action is integral and reduced. Thus it can be applied to the bases arising in the rational Meataxe in the previous chapter (the general eigenspaces of an algebra element or an endomorphism).

3.5. Condensation

Condensation is a very useful technique in module theory, whereby a large module for a large algebra is "condensed" to a small module for a small algebra, and information in the smaller module is more easily computed, hopefully yielding useful information about the original large module. Condensation has been used extensively in constructing modular representations (the original examples go back to Parker and Thackray in 1979 [Tha81]), but in our situation, we only need a fairly basic use of the theory and techniques.

We first state the basic definitions and results which we will need. For more detailed introductions to the basic concepts, we refer the reader to [Ryb90, Lux97, Mül04, Wil02]. In this section, let G be a finite group, F a field of characteristic zero and A the group algebra FG. We will only use **fixed-point** condensation, which is as follows. Suppose K

is a fixed subgroup of G (called the **condensation subgroup**). Define

$$e_K := \frac{1}{|K|} \sum_{k \in K} k \in A.$$

Then it is easy to see that e is an idempotent of A:

$$e_K^2 = \frac{1}{|K|^2} \sum_{k \in K} (k \sum_{l \in K} l) = \frac{1}{|K|} \sum_{k \in K} \frac{1}{|K|} \sum_{m \in K} m = e_K.$$

After setting $\tilde{A} = eAe$ and $\tilde{M} = Me$, it is elementary to show that $\tilde{M}$ is an $\tilde{A}$-module. $\tilde{A}$ is called the **condensed algebra** of A and $\tilde{M}$ is called the **condensed module** of M. The following standard results are mostly elementary.

Lemma 3.5.1. [Ryb90, Sec. 2] *Suppose that $A = FG$ and M is an A-module and $e = e_K$ as above for some subgroup K of G.*

1. *If S is a submodule of M, then Se is a submodule of Me.*
2. *If $\tilde{S}$ is a submodule of $\tilde{M} = Me$, then $\tilde{S} = Se$ for some submodule S of M.*
3. *If S is a simple submodule of M, then Se is either zero or simple (as an eAe-module).*
4. *If M is semisimple, then $\tilde{M} = Me$ is semisimple.*

Proposition 3.5.2. [MNRW02, 3.2]. *Let S, S' be simple A-modules, such that $\tilde{S} = Se \neq 0$ and $\tilde{S}' = S'e \neq 0$ and let $\tilde{A} = eAe$. Then $S \cong S'$ as A-modules if and only if $\tilde{S} \cong \tilde{S}'$ as $\tilde{A}$-modules.*

Definition 3.5.3. *If $\tilde{S}$ is a submodule of $\tilde{M} = Me$, then the computation of a submodule S of M such that $Se = \tilde{S}$ is called '**uncondensing** $\tilde{S}$'. Usually $\tilde{M}$ is represented in a reduced form, so there is an associated* **uncondensing map** *$\iota : \tilde{M} \rightarrow M$ giving the natural embedding of $\tilde{M}$ into M as vector spaces. We can thus simply compute S as the submodule of M generated by $\iota(\tilde{v})$ where $\tilde{v}$ loops over an F-basis of $\tilde{S}$. See* [MR99, 2.3] *for more discussion.*

Lemma 3.5.4. *(The Trace Formula) Let $e = e_K$ as above. Then there is a simple formula (first stated in* [SW97]*) for computing the trace of a condensed matrix which gives the action of ege on Me, as follows:*

$$\mathrm{Tr}_{Me}(ege) = \mathrm{Tr}_M(ege) = \mathrm{Tr}_M(gee) = \mathrm{Tr}_M(ge) = \\ \frac{1}{|K|} \sum_{k \in K} \mathrm{Tr}_M(gk) = \frac{1}{|K|} \sum_{k \in K} \chi_M(gk),$$

where χ_M is the character of M.

Corollary 3.5.5. *Setting g to the identity of G in the above formula, one can precompute the dimension of the condensed submodule $\tilde{S} = Se_K$ for a submodule S and a potential condensation subgroup K as*

$$\frac{1}{|K|} \sum_{k \in K} \chi_S(k) = \langle \chi_S \downarrow K, 1_K \rangle,$$

where χ_S is the character of S and 1_K is the trivial character of K.

Remarks 3.5.6. One non-trivial problem with the use of condensation is the so-called generation problem: given a set of elements for the algebra A, it is not clear in general whether the corresponding condensed elements generate the condensed algebra $\tilde{A}$. Noeske [Noe07] describes a method to determine whether one has enough generators of the condensed algebra; this was designed for modular representations. We will solve this problem in characteristic zero by use of the non-negative solutions algorithm from Sec. 3.2.

3.6. Generic Condensation Environments

3.6.1. Introduction. We now introduce a simple mechanism by which we can encapsulate various kinds of condensation (for characteristic zero) in a generic object and then apply the basic condensation operations generically in subsequent algorithms.

Definition 3.6.1. *Let M be an FG-module for a field F of characteristic zero, and K a subgroup of G. Call*

$$\mathscr{C} = (\mathsf{ImageMatrix}, \mathsf{Uncondense}, \mathsf{Action})$$

a **condensation environment** *for the condensed module $\tilde{M} = Me_K$ of M, if:*

- $\mathsf{ImageMatrix}(g)$ *is a function which takes $g \in G$ and returns the matrix of $e_K g e_K$, acting on the reduced $\tilde{M}$.*
- $\mathsf{Uncondense}(\tilde{v})$ *is a function which takes $\tilde{v} \in \tilde{M}$ and returns the vector $v = \iota(\tilde{v}) \in M$, where ι is the uncondensing map (as in Def. 3.5.3).*
- $\mathsf{Action}(v, g)$ *is a function which takes $v \in M$ and $g \in G$ and returns $vg \in M$ under the action of FG on M. (Note that typically the full matrix action of FG on M is not constructed explicitly, and so this operation is done by some special technique based on the particular kind of condensation.) This function will be passed to the algorithm* IntegralSpin *from Sec. 3.4 to compute the final uncondensed module.*

In the following subsections, we will show how to set up a condensation environment for the three different kinds of condensation which we will use. These will then be applied in a generic algorithm to compute irreducible representations automatically via condensation.

3.6.2. Permutation Condensation. The following algorithm sets up a condensation environment for the condensation of a permutation module of G over $\mathbb{Q}$, defined by a permutation representation $\phi : G \to P$ (recall that we are always using fixed-point condensation). Constructing a generator of the condensed algebra $\tilde{A}$ only involves counting the lengths of intersections of K-orbits for the given condensation subgroup K, so is quite fast in this case. The dimension d of the condensed module is the number of orbits of K. For more information and for proof of correctness of the constructions used in the following algorithm, see [MNRW02, 3.4] or [Wil02, 1.4].

Note that the entries of the matrices defining the condensed module will be positive integers bounded by d, so will be reasonably small in practice.

Algorithm PermutationCondensationSetup(ϕ, K)
Input:

- $\phi : G \to P$, a permutation representation of a finite group G.
- A condensation subgroup K of G.

```
OUTPUT:
  • A condensation environment $\mathscr{C}$ for condensation of the permutation module $\mathbb{Q}P$ at $K$.
STEPS:
  1. Set $\chi$ to the character of $\phi$.
     Let the $K$-orbits of $P$ be $\Omega_1, \dots, \Omega_d$.
  2. Set ImageMatrix := Function($g$)
     {
        Return the matrix in $\mathcal{M}_d(\mathbb{Z})$ whose $(i, j)$-th entry is
           $|\Omega_i\phi(g_k) \cap \Omega_j|/|\Omega_j|$.
     }
  3. Set Uncondense := Function($\tilde{v}$)
     {
        Set $v := (0, \dots, 0) \in \mathbb{Z}^n$, where $n$ is the degree of $P$.
        For $i := 1$ to $d$, for $j \in \Omega_i$ do: set $v[j] := \tilde{v}[i]$.
        Return $v$.
     }
  4. Set Action := Function($v, g$)
     {
        Return $v^g$ [the natural permutation of the coordinates of $v$ by $g$].
     }
  5. Set $\mathscr{C}$ := (ImageMatrix, Uncondense, Action) and return $\mathscr{C}$.
```

3.6.3. Induction Condensation. Suppose that H is a subgroup of a finite group G and $\rho_H : H \rightarrow \mathrm{GL}_n(F)$ is a representation of H. Let ρ_G be the **induced** representation $\rho_H \uparrow^G$. Tools for condensing ρ_G at a subgroup K of G and thus decomposing ρ_G (without explicitly constructing ρ_G) are described in [MR99]. We outline the main components here (slightly more concretely within our framework) and the associated methods for our condensation environment. For more details and for proof of correctness of the constructions used in the following algorithm, see the above reference.

Let M_H be the FH-module corresponding to ρ_H and let $M_G = M_H \uparrow^G$. A special transversal of G over H is first needed to define $M_G = M_H \uparrow^G$. Then we can set up the explicit reduced form of the condensed module $\tilde{M}_G = M_G e$ (where $e = e_K$, as above) and compute with it.

1. Let $\{g_i : i \in I\}$ be a set of H-K-double coset representatives in G and then for each $i \in I$, let $\{k_{ij} : j \in I_i\}$ be a set of right coset representatives for $H^{g_i} \cap K$ in K. Now set
$$T := \{g_i k_{ij} : i \in I, j \in I_i\}.$$
Then T is a set of right coset representatives for H in G. If $[v_1, \dots, v_d]$ is a basis of M_H as a vector space, then M_G has a vector space basis:
$$B := [v \otimes t : v \in [v_1, \dots, v_d], t \in T].$$

2. The action of G on M_G is as follows. Suppose $v \in M_H, t \in T, g \in G$. We compute the image $(v \otimes t)g$ as follows. There is a unique $t' \in T$ such that $Htg = Ht'$, and so there is some $h \in H$ with $tg = ht'$. Then $(v \otimes t)g = (vh) \otimes t'$. This covers the action of g on the basis B of M_G and thus on all of M_G by linear extension.
3. For each $i \in I$, set
$$H_i := H \cap g_i K g_i^{-1}, \quad e_i := \frac{1}{|H_i|} \sum_{h \in H_i} h.$$
The action of e_i maps M_H to $(M_H)e_i \subseteq M_H$, and, as a vector space, $(M_G)e$ is isomorphic to the direct sum of all the $(M_H)e_i$. The action of e on M_G is as follows. For $v \in M_H$ and $g_i k_{ij} \in T$, we have:
$$(v \otimes g_i k_{ij})e = \frac{|H_i|}{|K|} \left(ve_i \otimes \left(g_i \sum_{j \in I_i} k_{ij} \right) \right).$$
Note that the RHS is independent of the particular j on the LHS. Again, this covers the action of e on the whole basis of M_G and thus on all of M_G by linear extension.
4. Finally, it follows that $(M_G)e$ can be identified with the following subspace of M_G:
$$W := \bigoplus_{i \in I} \left((M_H)e_i \otimes \left(g_i \sum_{j \in I_i} k_{ij} \right) \right).$$
The uncondensing map ι simply injects W back into M_G.

We now apply the above for $F = \mathbb{Q}$ to set up an appropriate condensation environment for induction condensation. This setup operation is generally more expensive than for permutation condensation, but rarely takes a long time in our implementation, even when the degree is very large. The entries in the condensed module are of the same size roughly as the entries in the matrices defining ρ_H, and so are usually small when the degree of ρ_H is low, since ρ_H can generally be reduced to have very small entries.

Algorithm InductionCondensationSetup(G, M_H, K)
Input:

- A group G and a $\mathbb{Q}H$ module M_H for a subgroup H of G.
- A condensation subgroup K of G.

Output:

- A condensation environment $\mathscr{C}$ for condensation of the induced module $M_H \uparrow^G$ at K.

Steps:

1. Let $\{g_1, \ldots, g_l\}$ be a set of H-K-double coset representatives of G. Write $I := \{1, \ldots, l\}$.
2. For $1 \leq i \leq l$, let $\{k_{i,1}, \ldots, k_{i,l_i}\}$ be a set of $H^{g_i} \cap K$ right coset representatives in K. Write $I_i := \{k_{i,1}, \ldots, k_{i,l_i}\}$.

```
3. Set $T := \{g_i k_{ij} : i \in I, j \in I_i\}$ and let $f_T : T \times G \to T \times H$ be the map which, given
   $(t, g) \in T \times G$, returns $(t', h)$ where $t'$ is the unique element of $T$ with $Htg = Ht'$ and
   $h \in H$ with $tg = ht'$.
4. For $1 \leq i \leq l$, set $H_i := g_i H g_i^{-1} \cap K$, and $e_i := \frac{1}{|H_i|} \sum_{h \in H_i} h$. Set
       $W := \bigoplus_{i \in I} \left( M_H e_i \otimes \left( g_i \sum_{j \in I_i} k_{ij} \right) \right)$
   and let $\{w_1, \ldots, w_D\}$ be a basis of $W$ as a vector space (so $D$ is the dimension of $W$).
5. Write $M_G := (M_H)^G$. [The action algebra of $M_G$ is not explicitly constructed but
   understood to lie in the background theoretically in the following.]
   Set $U := F^D$ with standard basis $[u_1, \ldots, u_D]$ and let $\iota : U \to M_G$ be the embedding
   given by $u_i \mapsto w_i$ (the uncondensing map, with image $W$).
6. Set gAction := Function$(v, g)$ [Takes $v \in M_G, g \in G$ and returns $vg \in M_G$.]
   {
      Write $v = \sum_{i \in I} \sum_{j \in I_i} (v_{ij} \otimes t_{ij})$, with $v_{ij} \in M_H, t_{ij} \in T$.
      Set $(t'_{ij}, h_{ij}) := f_T(t_{ij}, g)$ for $i \in I, j \in I_i$.
      Return $\sum_{i \in I} \sum_{j \in I_i} ((v_{ij} h_{ij}) \otimes t'_{ij})$.
   }
7. Set eAction := Function$(v)$ [Takes $v \in M_G$ and returns $ve \in M_G$.]
   {
      Write $v = \sum_{i \in I} \sum_{j \in I_i} (v_{ij} \otimes g_i k_{ij})$, with $v_{ij} \in M_H$.
      Return $\frac{1}{|K|} \sum_{i \in I} |H_i| \sum_{j \in I_i} \left( v_{ij} e_i \otimes \left( g_i \sum_{j \in I_i} k_{ij} \right) \right)$.
   }
8. Set ImageMatrix := Function$(g)$
   {
      Return the matrix in $\mathcal{M}_D(F)$ whose $i$-th row (for $1 \leq i \leq D$) is
         $\iota^{-1}$(eAction(gAction$(\iota(u_i), g)$)).
   }
9. Set Uncondense := Function$(\tilde{v})$
   {
      Return $\iota(\tilde{v})$.
   }
10. Set Action := Function$(v, g)$
   {
      Return gAction$(v, g)$.
   }
11. Set $\mathscr{C}$ := (ImageMatrix, Uncondense, Action) and return $\mathscr{C}$.
```

3.6.4. Tensor Condensation. Suppose we have representations $\rho_1 : G \to \mathrm{GL}_{n_1}(F)$ and $\rho_2 : G \to \mathrm{GL}_{n_2}(F)$ of a finite group G and a field F. Let ρ be the **tensor product** representation $\rho_1 \otimes \rho_2$. Tools for condensing ρ at a subgroup K of G and thus decomposing ρ (without explicitly constructing ρ) are described in [LW98]. The authors concentrated on the case that F is a finite field, but we again apply this to the case that $F = \mathbb{Q}$ to construct a suitable corresponding condensation environment.

We first outline the basic setup. Suppose a semisimple A-module M has non-isomorphic constituents $S_1, \ldots, S_s$ and corresponding multiplicities $m_1, \ldots, m_s$. Then a **symmetry basis** of M is a basis

$$B = B_{11} \cup \ldots \cup B_{1m_1} \cup \ldots \cup B_{s1} \cup \ldots \cup B_{sm_s}$$

of the underlying vector space of M, where $B_{i\alpha}$ is a basis of the α-th simple submodule of M isomorphic to S_i, and such that the action of A on the submodule corresponding to $B_{i\alpha}$ is identical (not just equivalent) to the action of A on S_i.

Now let M_1 and M_2 be A-modules corresponding to the input representations ρ_1 and ρ_2, respectively. In the algorithm below, we first compute such a symmetry basis B for $M_1 \downarrow_K$ (from the corresponding S_i and m_i). Similarly, we compute a symmetry basis

$$C = C_{11} \cup \ldots \cup C_{1n_1} \cup \ldots \cup C_{s1} \cup \ldots \cup C_{sn_t}$$

of $M_2 \downarrow_K$, where the constituents of $M_2 \downarrow_K$ are $T_1, \ldots, T_t$ with corresponding multiplicities $n_1, \ldots, n_t$ and such that $T_i \cong S_i^*$ (the dual of S_i); note that some n_i may be zero. The basis B_T of the full tensor module $M_T = M_1 \otimes M_2$ is then given by the concatenation of all $B_{i\alpha} \otimes C_{j\beta}$, where $1 \le i \le s$, $1 \le \alpha \le m_i$, $1 \le j \le t$, $1 \le \beta \le n_i$ (unfolding the loops in that order). The rest of the construction is now described in the following algorithm; for more details and for proof of correctness of the constructions used in the following algorithm, see the above reference.

Algorithm TensorCondensationSetup(ρ_1, ρ_2, K)
Input:

- Rational representations ρ_1, ρ_2 of a group G.
- A condensation subgroup K of G.

Output:

- A condensation environment $\mathscr{C}$ for the condensation of $\rho_1 \otimes \rho_2$ at K.

Steps:

1. Let M_1, M_2 be the $\mathbb{Q}G$-modules corresponding to ρ_1, ρ_2 respectively.
   Set $d_1 := \mathrm{Dim}(M_1)$, $d_2 := \mathrm{Dim}(M_2)$.
2. Set $D_1 :=$ RationalMeataxe($M_1 \downarrow_K$).
   Using D_1, determine the pairwise non-isomorphic constituents $S_1, \ldots, S_s$ of $M_1 \downarrow_K$ with corresponding multiplicities $m_1, \ldots, m_s$.
   Let U_1 be the transformation matrix corresponding to a symmetric basis of M_1 w.r.t. the S_i and the s_i and set $M_1' := (M_1)^{U_1}$.

```
3. Let $T_i = S_i^*$ for $1 \le i \le s$.
   Set $D_2 :=$ RationalMeataxe$(M_2 \downarrow_K)$.
   Using $D_2$, determine $T_{s+1}, \ldots, T_t$ and $n_1, \ldots, n_t$ such that the pairwise non-isomorphic
   constituents of $M_2 \downarrow_K$ are $T_1, \ldots, T_t$ with corresponding multiplicities $n_1, \ldots, n_t$ (note
   that some $n_i$ may equal 0 for $1 \le i \le s$).
   Let $U_2$ be the transformation matrix corresponding to a symmetric basis of $M_2$ w.r.t.
   the $T_i$ and the $n_i$ and set $M_2' := (M_2)^{U_2}$.

4. For $1 \le i \le s$ do:
   {
      If $m_i = n_i = 0$ then skip to the next $i$.
      Set $e_i := \frac{1}{|H|} \sum_{h \in H} S_i(h) \otimes T_i(h)$.
      Set $q_i$ to the echelonized basis matrix over $\mathbb{Q}$ of the rowspace of $e_i$
          and let $Q_i$ be the rows of $q_i$ (i.e., a basis for the rowspace of $q_i$).
      Set $p_i$ to the unique matrix over $\mathbb{Q}$ such that $p_i q_i = e_i$.
   }
   Set $Q := \bigcup_{i=1}^{s} \bigcup_{\alpha=1}^{m_i} \bigcup_{\beta=1}^{n_i} Q_i$, where each copy of $Q_i$ corresponds to the tensor product
   of the $\alpha$-th copy of $S_i$ and the $\beta$-th copy of $T_i$.
   Set $d$ to the length of $Q$.

5. For $A \in \mathcal{M}_{d_1}(\mathbb{Q})$, let $A_{i\alpha j\gamma}$ denote the submatrix of $A$ indexed by the $(i, \alpha)$-th row
   block corresponding to the $\alpha$-th copy of $S_i$ in the symmetric basis of $M_1$ and the
   $(j, \gamma)$-th column block corresponding to the $\gamma$-th copy of $T_j$ in the symmetric basis of
   $M_2$; similarly for $B_{i\beta j\delta} \in \mathcal{M}_{d_2}(\mathbb{Q})$.
   For $\tilde{X} \in \mathcal{M}_d(\mathbb{Q})$, let $\tilde{X}_{i\alpha\beta j\gamma\delta}$ denote the submatrix of $\tilde{X}$ indexed by the $(i, \alpha, \beta)$-th
   row block and the $(j, \gamma, \delta)$-th column block (corresponding to the decomposition of $Q$
   above, which $\tilde{X}$ acts on).

   Set ImageMatrix := Function$(g)$
   {
      Set $\tilde{X}$ to the zero matrix of $\mathcal{M}_d(\mathbb{Q})$.
      Set $A := M_1'(g)$, $B := M_2'(g)$.
      For $i := 1$ to $s$, $\alpha := 1$ to $m_i$, $\beta := 1$ to $n_i$,
          $j := 1$ to $s$, $\gamma := 1$ to $m_i$, $\delta := 1$ to $n_i$ do:
      {
          Set $C := A_{i\alpha j\gamma} \otimes B_{i\beta j\delta}$.
          Set $\tilde{X}_{i\alpha\beta j\gamma\delta} := q_i \cdot C \cdot p_j$.
      }
      Return $\tilde{X}$.
   }

6. For $v \in \mathbb{Q}^{d_1 d_2}$, let $v_{i\alpha j\beta} \in \mathbb{Q}^{c_i^2}$ (where $c_i = \mathrm{Dim}(S_i)$) denote the subvector of $v$ with
   coordinates corresponding to the component $B_{i\alpha} \otimes C_{j\beta}$ of the basis $B_T$ of the full
   tensor module $M_T$.

   Set Uncondense := Function$(\tilde{v})$
   {
        [Input $\tilde{v} \in \mathbb{Q}^{d}$; output is $\iota(\tilde{v}) \in \mathbb{Q}^{d_1 d_2}$.]
        Set $v$ to the zero vector of $\mathbb{Q}^{d_1 d_2}$.
        Set $k := 1$.
        For $i := 1$ to $s$, $\alpha := 1$ to $m_i$, $\beta := 1$ to $n_i$ do:
        {
            Set $c_i := \mathrm{Dim}(S_i)$ and let $r_i$ be the number of rows in $q_i$.
            Let $u$ be the subvector $\tilde{v}[k, \ldots, k + r_i - 1] \in \mathbb{Q}^{r_i}$.
            Set $v_{i\alpha i\beta} := u \cdot q_i$.
            Set $k := k + r_i$.
        }
        Return $v$.
    }
7. Set Action := Function($v, g$)
    {
        Let $A$ be the $d_1 \times d_2$ matrix corresponding to $v$ (in row major order).
        Set $B := \rho_1(g)^{tr} \cdot A \cdot \rho_2(g)$.
        Return the vector of length $d_1 d_2$ corresponding to $B$.
    }
8. Set $\mathscr{C}$ := (ImageMatrix, Uncondense, Action) and return $\mathscr{C}$.
```

Remarks 3.6.2. We note the following points on the implementation for rational representations:

1. If $\rho_1 = \rho_2$, then we of course need only decompose ρ_1 and compute its symmetry basis and there are other basic optimizations which can be made. This case arises often (as can be seen in examples later).
2. If the condensation subgroup K is cyclic with generator g_K, then the decomposition of the restricted modules can be found easily by use of the primary rational form or generalized Jordan form of $\rho_1(g_K)$ and $\rho_2(g_K)$ respectively (we use the algorithm described in [Ste97]). The constituents can be matched by simply comparing the powers of irreducible polynomials which give the primary invariant factors of the matrices, and the symmetry bases can be constructed from the corresponding transformation matrices.
3. For the general case, where K is not cyclic, we have given a default method where we compute the decomposition of each of the restricted representations via the rational Meataxe. An alternative is to compute the characters of these restricted representations and decompose these w.r.t. $\mathrm{Irr}_{\mathbb{Q}}(K)$ and then compute irreducible rational representations corresponding to these irreducible characters, using the algorithm IRREDUCIBLERATIONALREPRESENTATIONS below (Sec. 3.8). Since K is very often rather small in practice (order typically under 100; see below), it will in general be very easy to compute the relevant irreducible representations of K. Then one can compute the Hom-module from each constituent to $M_1 \downarrow_K$ and $M_2 \downarrow_K$ to construct each of the symmetry bases. This variant has also been implemented and we find that it is preferable when at least one of the input representations has large degree (above 200).

4. For the action of the tensor product representation on a vector $v \in \mathbb{Q}^{(d_1 d_2)}$, v is written as a $d_1 \times d_2$ matrix (in row major order), and then this matrix is multiplied on the left by a $d_1 \times d_1$ matrix and on the right by a $d_2 \times d_2$ matrix. The (classical) complexity of this operation is thus ${d_1}^2 d_2 + d_1 {d_2}^2$ arithmetic operations (using only classical multiplication), which in general is significantly less than ${d_1}^2 {d_2}^2$, which would be the complexity of performing the standard vector-times-matrix multiplication in the full tensor product.
5. The setup operation is typically much more expensive than for permutation and induction condensation. Also, the entries in the matrices defining the condensed module may be rather large. Thus tensor condensation is generally less suitable when constructing representations directly via the splitting method in high degree. But tensor condensation can also be used in the hybrid algorithm which will be described in Chapter 6: in this situation, the entry size for the condensed module will not be an issue.

3.7. Automatic Condensation over the Rational Field

We now present the key algorithm AUTOMATICCONDENSATION which constructs a desired irreducible rational representation of a finite group G via condensation by extracting it as a constituent of a given virtual permutation, induced, or tensor representation σ of G. The algorithm automatically chooses a suitable condensation subgroup K so that the desired constituent of σ is not mapped to zero under condensation and the corresponding constituent of the condensed module $\tilde{M}$ can be identified in a decomposition of $\tilde{M}$. The desired irreducible representation can then be constructed by applying the INTEGRALSPIN algorithm to the corresponding uncondensed vectors.

The following notation will be used in this section:

1. Write $\texttt{Trace}(\chi, K, g) = \frac{1}{|K|}\sum_{k \in K} \chi(gk)$ for character χ, $K \leq G$ and $g \in G$ (using the Trace Formula from Lem. 3.5.4).
2. Write $\texttt{CondDim}(\chi, K) = \langle \chi \downarrow K, 1_K \rangle$ for character χ and for $K \leq G$ (giving the condensed dimension of χ w.r.t. K).
3. For fixed $\{x_1 = 1, x_2, \ldots, x_r\} \subseteq G$, and for a character χ, call
$$(\texttt{Trace}(\chi, K, x_1), \ldots, \texttt{Trace}(\chi, K, x_r)) \in \mathbb{Z}^r$$
the 'trace vector' of χ w.r.t. K. Note that the x_i need not be class representatives of G.

The heart of the algorithm is the search for a suitable condensation subgroup K. For such a K, let $\tilde{M}$ be the corresponding condensed module. The properties sought for K are:

1. The dimension of $\tilde{M}$ should be as small as possible, so that the rational Meataxe can decompose it easily.
2. The simple constituent $\tilde{S}$ of $\tilde{M}$ corresponding to χ must not condense to zero.
3. The simple constituent $\tilde{S}$ of $\tilde{M}$ corresponding to χ must be uniquely identifiable via traces. More precisely, if T_i gives the trace vector of the i-th constituent of $\tilde{M}$ (which can be computed by decomposing the character of M into irreducibles) and the index I corresponds to χ, then we require that T_I can only be expressed in one way as a non-negative linear combination of all the T_i.

The algorithm first searches for the best K, subject to these conditions. The search is over a suitable list L of small subgroups of G. Typically, L should contain subgroups which are small and easy to compute, such as the cyclic subgroups generated by all class representatives and the Sylow subgroups and all their subgroups up to conjugacy (one could also include all or a selection of the subgroups of G having order up to some bound such as 500). After finding the best K, the algorithm sets up the condensed module using a given generic Setup function (which calls one of the setup functions of the previous section with information defining σ and the chosen K), and then calls the rational Meataxe to extract the appropriate constituent, and uncondenses the submodule to construct the final representation. Finally, a verification step at the end detects the potential problem where the condensed algebra does not have enough generators. The full algorithm is as follows.

Algorithm AutomaticCondensation($G, \psi, \mathsf{Setup}(K), \chi$)

Input:

- A finite group G, a rational character ψ of G, and a generic function $\mathsf{Setup}(K)$ which takes a subgroup K of G and returns a condensation environment $\mathscr{C}$ for the condensation at the subgroup K of some underlying virtual representation $\sigma : G \longrightarrow \mathrm{GL}_n(\mathbb{Q})$ which affords ψ.
- A character $\chi \in \mathrm{Irr}_{\mathbb{Q}}(G)$, such that χ is a constituent of ψ.

Output:

- A rational representation ρ of G (which is integral if σ is integral) affording χ.

Steps:

1. Set L to a suitable list of small subgroups of G which at least contains the trivial subgroup (see the above discussion).
2. Set $r := 20$. Set x_1 to the identity element of G and set $[x_2, \dots, x_r]$ to $r-1$ distinct random elements of G.
3. Let
$$\psi = \sum_{i=1}^{k} m_i \cdot \chi_i, \quad \chi_i \in \mathrm{Irr}_{\mathbb{Q}}(G)$$
(with each $m_i > 0$) be the decomposition of ψ into irreducible rational characters and let I be the index such that $\chi = \chi_I$.
4. [*Find condensation subgroup K_{best} with smallest possible condensed dimension and such that the desired constituent does not collapse to zero w.r.t. it and the trace vectors $[T_1, \dots, T_k]$ corresponding to each condensed constituent can be uniquely identified.*]

   Set $K_{\mathrm{best}} := T_{\mathrm{best}} := \mathscr{C}_{\mathrm{best}} := 0, D_{\mathrm{best}} := \infty$.

   For each subgroup K in L do:

   {

   Set $D := \texttt{CondDim}(\psi, K)$. If $D \geq D_{\mathrm{best}}$ then skip to the next K.

   If $\texttt{CondDim}(\chi, K) = 0$ then skip to the next K (K is invalid).

   For $1 \leq i \leq k$, set $T_i := (\texttt{Trace}(\chi_i, K, x_1), \dots, \texttt{Trace}(\chi_i, K, x_r)) \in \mathbb{Z}^r$.

```
        Let $[i_1, \dots, i_t]$ be the indices from $[1, \dots, k]$ such that $T_{i_j}[1] > 0$ for each $j$
            (corresponding to all the constituents not mapped to zero).
        [*Check that the constituent for $\chi$ can be uniquely identified by traces.*]
        Set $L :=$ NonNegativeSolutions($[T_{i_1}, \dots, T_{i_t}], m_I \cdot T_I, 2$).
        If $\#L > 1$ then skip to the next $K$ ($K$ is invalid).
        [*Now we have a valid $K$ with new smallest dimension $D$.*]
        Set $K_{\text{best}} := K, D_{\text{best}} := D, \mathscr{C}_{\text{best}} := \mathscr{C}, T_{\text{best}} := [T_1, \dots, T_k]$.
        If $D_{\text{best}} <$ MinDimBound then break.
    }
    Set $K := K_{\text{best}}, \mathscr{C} := \mathscr{C}_{\text{best}}, [T_1, \dots, T_k] := T_{\text{best}}$.
5.  Set $\mathscr{C} :=$ Setup($K$). For $i := 1$ to $r$ do: Set $\tilde{X}_i := \mathscr{C}$.ImageMatrix($x_i$).
6.  Set $\tilde{A}$ to the $\mathbb{Q}$-algebra with generators $[\tilde{X}_1, \dots, \tilde{X}_r]$, set $\tilde{M}$ to the corresponding
    condensed $\tilde{A}$-module and set TraceInfo := $\langle m_I, T_I \rangle$.
    Set $[\tilde{S}_1, \dots, \tilde{S}_s] :=$ RationalMeataxe($\tilde{M}$, TraceInfo).
7.  Let $i$ be such that the $r$ traces of the generators of the action on $\tilde{S}_i$ equals $T_I$. If there
    is none such, then go to Step 9 (condensed algebra was bad).
8.  Set $U := \{\mathscr{C}$.Uncondense($\tilde{u}$) : $\tilde{u} \in \tilde{U}\}$, where $\tilde{U}$ is a basis of $\tilde{S}_i$ w.r.t. the embedding
    of $\tilde{S}_i$ into $\tilde{M}$.
    Set $\phi :=$ Function($v, j$) { Return $\mathscr{C}$.Action($v, g_j$). }
    Set $B, [A_1, \dots, A_n] :=$ IntegralSpin($U, \phi, n$).
    Set $\rho$ to the representation of $G$ given by $\rho(g_j) = A_j$ for each $j$.
    If the character of $\rho$ equals $\chi$ then return $\rho$.
9.  The condensed algebra must have been bad (not enough generators). So set $r' :=$
    $r + 10$, choose random $x_{r+1}, \dots, x_{r'} \in G$, extend each $T_i$ with the traces for the new
    coordinates, set $\tilde{X}_{r+1}, \dots, \tilde{X}_{r'}$ as in Step 5, then set $r := r'$ and go to Step 6.
10. Return $[\rho_1, \dots, \rho_l]$.
```

Theorem 3.7.1. *Algorithm* AutomaticCondensation *is correct.*

Proof. After basic initialization, the critical part of the algorithm is the loop in Step 4 which searches for the best condensation subgroup K (giving the smallest condensed dimension) which satisfies the conditions listed on p. 60. Suppose that $\tilde{M}$ is the condensed $\tilde{A}$-module corresponding to a potential K. The first condition on K applies Cor. 3.5.5 to check that the desired constituent of M corresponding to χ does not condense to zero inside $\tilde{M}$. The more complex condition on K involves the traces of the action of $\tilde{A}$ on the constituents of $\tilde{M}$. For each i, T_i is set to the trace vector of χ_i w.r.t. K and since $x_1 = 1$, the first coordinate of T_i gives the dimension of the constituent of $\tilde{M}$ corresponding to χ_i, and this is positive at least for $i = I$ by the first condition on K (where I is such that $\chi = \chi_I$). The call to NonNegativeSolutions checks that the trace vector of $m_I \cdot \chi_I$ can only be expressed in exactly one way as a non-negative linear combination of $T_{i_1}, \dots, T_{i_t}$ (the trace vectors of the non-zero condensed constituents), so the homogeneous constituent of $\tilde{M}$ whose trace vector equals $m_I \cdot T_I$ can be uniquely identified in a homogeneous decomposition of the full condensed module $\tilde{M}$ corresponding to K. (The bound 2 is passed for the maximum number of solutions desired, since we only need to know whether the solution is unique or not.) Taking K to be the trivial subgroup of G clearly satisfies all the conditions, so K_{best} must be set to some subgroup K satisfying the conditions when the loop is exited.

Steps 5 to 6 clearly set up the condensed $\tilde{A}$-module $\tilde{M}$ w.r.t. the best condensation subgroup K and decompose this via the rational Meataxe, using the trace information matching the desired characters. Assume first that there are enough generators of $\tilde{A}$, so by Lem. 3.5.1 and the condition that $\texttt{CondDim}(\chi, K) \neq 0$, there must be a simple submodule $\tilde{S}_i$ of $\tilde{M}$ which is the condensation of a submodule of the full module corresponding to σ whose character is χ. Now T_I (the trace vector of $\chi = \chi_I$) uniquely identifies $\tilde{S}_i$ because of the condition on the unique solution in the preceding call to NonNegativeSolutions on $[T_{i_1}, \ldots, T_{i_t}]$ and $m_I \cdot T_I$ for this K. So there must be a simple $\tilde{S}_i$ returned by the rational Meataxe whose trace vector equals T_I. (Either the rational Meataxe will return such a constituent alone if the heuristic method using the trace information succeeds or simply a full decomposition, and either case, the relevant constituent must be present and it alone can have trace vector T_i.) Thus in Step 8, ρ must be set to a valid representation affording χ so the check on the character of ρ must succeed and the output is correct.

On the other hand, if there are not enough generators of the condensed algebra $\tilde{A}$, then it can happen that the condensed module $\tilde{M}$ decomposes more than it should. In such a case, this must be detected because either the simple constituent $\tilde{S}_i$ with the appropriate trace vector will not be found, or the character test on ρ will fail (the final representation will typically be the sum of irreducible representations in this bad case and this can also be detected in the integral spin before computing the full character). So in this case, the algorithm adds more random generators of G and recomputes the condensed module with the same K but with the enlarged condensed algebra $\tilde{A}$. Eventually the correct condensed algebra must be generated and so the algorithm will terminate. □

Remarks 3.7.2. We note the following points on the implementation:

1. One can break out of the loop in Step 4 as soon as some K is found such that the corresponding condensed dimension is less than some bound B, under the assumption that the rational Meataxe will be fast for modules with dimension up to B. We take $B = 200$ in the implementation.
2. The rather strict condition involving the call to NonNegativeSolutions in Step 4 is of critical important in practice. If K is a potential subgroup such that the condensation of the desired constituent is not zero, while the condensed module $\tilde{M}$ has small dimension (which often happens), then there is good chance that several distinct elements of G will map to elements of $e\mathbb{Q}Ge$ having the same trace, so the trace vectors of the constituents of the condensed module will have much repetition and will be very similar on most coordinates. There is then a very good chance that there is more than one non-negative solution to the associated linear system and so this K must be rejected. Thus the use of NonNegativeSolutions is critical, and its efficiency (via the pruning of bounds) is very important too, so we can quickly determine whether a potential condensation subgroup K cannot be used.

3. Note that for computing $\texttt{Trace}(\chi_i, K, g)$ for fixed K and g but for differing χ_i in Step 4, we can first compute the class map values for each element of the coset gK once and then for each χ_i, we can compute the traces more quickly. Clearly this involves a loop over K and so as the size of K increases, the computation of the traces can become expensive but a larger K typically implies a condensed module of smaller dimension (and a corresponding speedup in setting that up and decomposing it by the rational Meataxe), so using a larger K is preferable when χ has large degree, even when the trace computations are non-trivial.
4. The case that the best K must equal the trivial subgroup does arise; for example, when the endomorphism ring of the desired representation has a large centre and non-trivial multiplicity (since the endomorphism ring of the condensed module must be the same, there is often no non-trivial condensation subgroup without collapsing to zero). In this case, in Step 5 we immediately set $\tilde{M}$ to the $\mathbb{Q}G$-module corresponding to the full virtual representation (thus skipping the condensation machinery) and use the Meataxe to extract the desired constituent (and the trace information uses the normal characters). Because $\tilde{M}$ is a $\mathbb{Q}G$-module in this case, the algorithm SplitHomogeneousByMinimalField can be used if the splitting of the homogeneous module is difficult.
5. For simplicity of exposition, we have presented the algorithm so that a single representation is requested and constructed. But in our implementation, the algorithm allows several characters to be given, so that corresponding representations are constructed (the conditions on K apply to all of the characters). This means that only one condensed module has to be constructed and split by the Meataxe (and that uses the heuristic method with the trace information for each desired constituent). Thus several representations can be efficiently constructed from the one virtual representation via condensation.
6. Note that the character test in Step 8 can in fact be done modularly within IntegralSpin: assuming the prime p is greater than the usual bound, then after the initial modular spin in that algorithm, one can immediately compute the character modulo p to verify that it is correct (still using the advanced algorithm of Sec. 3.3) before constructing the integral representation. This means that the cost of the character test is generally trivial in practice.

Example 3.7.3. Let G be the sporadic simple Fischer group Fi_{23}, of order

$$4089470473293004800 = 2^{18}.3^{13}.5^{2}.7.11.13.17.23.$$

A minimal-degree faithful representation of G has degree 782, which can be realized over $\mathbb{Q}$. We computed such a representation as follows (table entry on p. 181).

A degree-31671 permutation representation from the online ATLAS [WWT+] was used to define G. The corresponding permutation module M splits as $1 + 782 + 30888$. AutomaticCondensation was called with this permutation representation and the desired character χ of degree 782.

After searching in 98 subgroups generated by the class representatives and elements of Sylow subgroups (2.5s), a condensation subgroup K of order 243 was selected so that the corresponding condensed module $\tilde{M}$ had dimension 185 (constructed in 2.7s, via 20 random elements of G). The constituents of M condense to submodules of dimension 1, 10, 174 respectively, and the dimension-10 condensed constituent $\tilde{S}$ corresponded to χ. Then the rational Meataxe was called on $\tilde{M}$ with the corresponding trace information. That first computed the primary invariant spaces of a random linear combination of the algebra generators (1.2s); the invariant spaces had dimensions 1, 10, 174 corresponding to the full split. It was then verified (in 0.1s) that the dimension-10 space was a submodule $\tilde{S}$ of $\tilde{M}$. The trace of the action of $\tilde{S}$ matched the desired trace information, so $\tilde{S}$ was returned immediately.

The uncondensed vectors were passed to IntegralSpin, with the permutation action of degree 31671. The initial basis with the modular spin took 36.4s. This yielded a 782×31671 integral basis matrix B. The following operations were then done, each on an integral matrix of the same shape:

- B was set to Saturation(B) in 185.1 secs.
- B was then set to HermiteForm(B) in 46.2 secs.
- B was then set to LLL(B) in 55.7 secs.
- B was then set to Seysen(B) in 46.3 secs.

Finally, the reduced action of the permutation action on B was computed in 22.2 secs, yielding two 782×782 integral matrices defining the desired representation of G. The character of the representation was then computed instantly (since the combination of irreducible characters was unique, based on the dimension alone), verifying that the condensed algebra had enough generators.

The whole computation took 596 seconds, and the images of the standard generators in the resulting representation have integral entries whose absolute values have maximum value 214 and average 4.2. Note that if the Seysen reduction step is omitted, then the generator images have integral entries whose absolute values have maximum value 2576 and average 11.5, so the Seysen reduction is well worth doing to reduce the entries.

3.8. Constructing Irreducible Rational Representations

We can now present a completely automatic algorithm which, given a finite group G and a set of characters from $\mathrm{Irr}_{\mathbb{Q}}(G)$, computes corresponding irreducible rational representations of G. The returned representations are in fact always integral, which helps to keep the size of the entries small in general. The algorithm is the critical 'base engine' on which most of the later algorithms to compute representations are based.

The basic idea is to extract the representations as irreducible constituents of various virtual representations of G, using the AutomaticCondensation algorithm from the previous section. The virtual representations are selected by means of a priority queue of potential representations to be decomposed. Each entry of the queue contains information for a method for constructing a new (generally reducible) representation and the character of that representation. The queue is sorted by difficulty, based on the degree of the virtual representation, so smaller-degree representations are considered first. As a new potential representation is removed from the head of the queue, the decomposition of its character is computed, and if there are any irreducible characters in the decomposition corresponding to representations which have not yet been found, then the method is applied to find such representations.

More precisely, the priority queue contains triples of the form $\langle \psi, t, I \rangle$ where ψ is the character of the virtual representation, t is a tag indicating the kind of representation (PERM, IND or TENS), and I is other information depending on the kind. When condensation of a virtual representation w.r.t. a condensation subgroup K is to be used, the algorithm calls the appropriate condensation environment setup function from the previous sections. The particular cases for a triple $\langle \psi, t, I \rangle$ are as follows:

- $t =$ PERM: Here I is a permutation representation with character ψ, so the setup function calls PermutationCondensationSetup on I and K.
- $t =$ IND: Here $I = \langle H, \chi_H \rangle$ is a pair such that H is a subgroup of G, $\chi_H \in \mathrm{Irr}_{\mathbb{Q}}(H)$ and $\psi = \chi_H \uparrow^{G}$, so the setup function calls InductionCondensationSetup on ρ_H (which affords χ_H) and K, after ρ_H has first been recursively constructed.
- $t =$ TENS: Here $I = \langle \rho_1, \rho_2 \rangle$, where ρ_1, ρ_2 are representations of G which have already been constructed and ψ is the character of $\rho_1 \otimes \rho_2$, so the setup function calls TensorCondensationSetup on ρ_1, ρ_2 and K.

We also define the degree of a triple $\langle \psi, t, I \rangle$ to be the degree of ψ and always select the next triple with smallest degree which will yield a new representation.

Apart from using the automatic condensation algorithm on the above virtual representations, the algorithm also immediately constructs the tensor product, exterior tensor square or symmetric tensor square of representations when they are constructed, if such representations afford one of the desired characters. (In the following, we use 'ExteriorSquare' and 'SymmetricSquare' to denote the latter two operations, acting on a character or representation).

Note also that the algorithm in practice always returns integral representations, since it only extracts constituents of integral representations (permutation or induction of integral representations by recursion) and it always saturates the invariant basis when creating a submodule. But since we do not consider the issue of inequivalence of integral representations in this thesis, we will continue to call the resulting representations rational, to make it clear that we are only considering equivalence over $\mathbb{Q}$.

Algorithm IrreducibleRationalRepresentations($[\chi_1, \ldots, \chi_k]$)
Input:

- Distinct characters $[\chi_1, \ldots, \chi_k]$ from $\mathrm{Irr}_{\mathbb{Q}}(G)$, for a finite group G.

Output:

- Irreducible rational representations $[\rho_1, \ldots, \rho_k]$ of G affording $[\chi_1, \ldots, \chi_k]$ respectively. The representations will always be integral.

Steps:

1. Set SubgroupIndex := 100 (or some other initial value; determines the initial index range of subgroups to be considered).
   Set Q to an empty priority queue of triples (see above discussion).
   Set ρ_i to 0 for $1 \leq i \leq k$.

```
2. [Extend queue $Q$ using higher index subgroups if necessary.]
   While $Q$ is empty, or the degree of the head of $Q \geq$ SubgroupIndex do:
   {
      Set $L$ to a list of the subgroups of $G$ (up to conjugacy) with
          index in [SubgroupIndex $\ldots 2 \cdot$ SubgroupIndex $- 1$],
          sorted by index in $G$ (with smallest index first).
      Set SubgroupIndex $:= 2 \cdot$ SubgroupIndex.
      For $H$ in $L$ do:
      {
          [Include new representations obtainable from $H$ in queue.]
          Let $f : G \to P$ be the permutation representation of $G$ given by the action
          of $G$ on the right cosets of $H$ and let $\psi$ be the character of $f$ and then
              include $\langle \psi, \mathrm{PERM}, f \rangle$ in $Q$.
          Compute $\mathrm{Irr}_{\mathbb{Q}}(H)$ and then for each $\chi_H \in \mathrm{Irr}_{\mathbb{Q}}(H)$ do:
              Include $\langle \chi_H \uparrow^G, \mathrm{IND}, \langle H, \chi_H \rangle \rangle$ in $Q$.
      }
   }

3. [Find smallest virtual representation in $Q$ which will give something new.]
   Set $c := 0$.
   Sort $Q$ by degree of first components, with smallest first.
   While $Q$ is non-empty, and the degree of the head of $Q <$ SubgroupIndex do:
   {
      Remove $T = \langle \psi, t, I \rangle$ from the head of $Q$.
      If there is an $i$ with $1 \leq i \leq k$ such that $\rho_i = 0$ and $\chi_i$ is a component of
          $\psi$ (w.r.t. $\mathrm{Irr}_{\mathbb{Q}}(G)$), then set $c := i$ and break out of the loop.
   }
   If $c = 0$ (nothing new found) then go to Step 2.

4. [Now $T = \langle \psi, t, I \rangle$ must provide a representation for $\chi_c$. Set Setup($K$) to be the
   function which takes condensation subgroup $K$ and calls the appropriate function to
   set up a condensation environment for $\psi$ and $K$.]
   If $t =$ PERM then:
   {
      Set $f := I$ [the permutation representation].
      Set Setup := Function($K$)
          { Return PermutationCondensationSetup($f, K$). }
   }
   Else if $t =$ IND then:
   {
      Write $I = \langle H, \chi_H \rangle$.
      Set $[\rho_H] :=$ IrreducibleRationalRepresentations($[\chi_H]$).
      Set Setup := Function($K$)
          { Return InductionCondensationSetup($G, \rho_H, K$). }
   }
   Else ($t =$ TENS):
   {
        Write $I = \langle \rho_1, \rho_2 \rangle$.
        Set Setup := Function($K$)
            { Return TensorCondensationSetup($\rho_1, \rho_2, K$). }
    }
5.  [Create representation affording $\chi_c$.]
    If $t$ = IND and $\psi = \chi_c$ then:
        Set $\rho_c := \rho_H \uparrow^G$. [Exact induction; skip condensation]
    Else:
        Set $\rho_c :=$ AutomaticCondensation($G, \psi$, Setup, $\chi_c$).
6.  [Consider the tensor product of $\rho_c$ and each other existing representation.]
    For each $s$ with $1 \le s \le k$ and $\rho_s \ne 0$ do:
    {
        Set $\psi := \chi_c \cdot \chi_s$.
        If $\psi = \chi_t$ for some $1 \le t \le k$ then:
        {
            If $\rho_t = 0$ then set $\rho_t := \rho_c \otimes \rho_s$.
        }
        Else:
            Include $\langle \psi$, TENS, $\langle \rho_c, \rho_s \rangle \rangle$ in $Q$.
    }
    If ExteriorSquare($\chi_c$) $= \chi_t$ and $\rho_t = 0$ for some $1 \le t \le k$ then:
        Set $\rho_t :=$ ExteriorSquare($\rho_c$).
    If SymmetricSquare($\chi_c$) $= \chi_t$ and $\rho_t = 0$ for some $1 \le t \le k$ then:
        Set $\rho_t :=$ SymmetricSquare($\rho_c$).
7.  If at least one of $\rho_1, \dots, \rho_k$ is 0 then go to Step 2. Otherwise, return $[\rho_1, \dots, \rho_k]$.
```

Theorem 3.8.1. *Algorithm* IrreducibleRationalRepresentations *is correct.*

Proof. The correctness of the algorithm essentially follows from the correctness of the preceding condensation algorithms which are called.

Step 2 expands Q so that it has information for all permutation or induced virtual representations up to the current index limit (and that limit is increased while the queue is empty). Step 3 finds a tuple $T = \langle \psi, t, I \rangle$ such that ψ includes as a component a character χ_c for one the desired representations which is not already computed. Then **Setup** is assigned in Step 4 to the appropriate generic function to set up the condensation of the virtual representation σ affording ψ, so AutomaticCondensation can call the setup function for the particular condensation subgroup K which it chooses. Thus in Step 5, ρ_c must be set to representation affording χ_c which is a constituent of the virtual representation σ: in the case that induction is to be performed and $\psi = \chi_H \uparrow^G = \chi_c$, then clearly ρ_c can be set immediately to $\rho_H \uparrow^G$; otherwise the automatic condensation algorithm is used. In Step 6, the loop over s clearly checks whether a desired representation affording χ_t can be formed by the exact tensor product of ρ_c with another existing representation immediately (the correctness clearly follows from the check on the corresponding characters), and the loop also inserts the information into the queue Q corresponding to all other potential tensor products involving ρ_c and currently constructed representations.

As for termination, note that when Step 3 starts to search for a suitable T in Q, clearly Q will contain the information (not already considered) for at least all permutation representations of G of degree up to `SubgroupIndex` and all induced representations for subgroups of index up to `SubgroupIndex`. In the worst case, the index limit variable `SubgroupIndex` will eventually reach the order of G, so the regular permutation representation of G will be inserted in the queue, and since this contains all irreducible representations of G, all desired representations must eventually be constructed. □

Remarks 3.8.2. There are very many parameters and options in the implementation, which are useful for handling different kinds of groups. We note the most important of these.

1. One can set a limit on the degree of a virtual representation which will be considered, so that, for example, $\chi_H \uparrow^G$ will not be considered if its degree is too large.
2. The variable `SubgroupIndex` can of course initially be set to a larger value, and successively increased by a greater ratio, depending on G. The user can also pass in an explicit list of subgroups to be used, or a list of indices, so that only subgroups whose index in G is in this list are used.
3. Reaching index $|G|$ and thus splitting the full regular representation is not as impractical as it sounds: for groups up to order a few thousand, say, it can be very fast. The point is that the degree of the virtual representation can be very much larger than the degree of the desired representations.
4. A basic issue is computing the relevant subgroups of G. In our implementation, we compute the subgroups of a group G by the algorithm described in [CHSS05]. If it is easy to compute all subgroups whose index in G is moderate (say up to index about 100000), then the algorithm is very effective. This covers a vast range of groups. If there are no subgroups of reasonably small index, then this algorithm will fail in practice, but the extension algorithms later in the thesis will handle this situation well.
5. When recursing in the induction case to construct a representation of a subgroup H, the inner call uses the algorithm with default options, and thus potentially recurses again to construct the representation of H via induction. This situation happens often in our implementation and so multiple levels of recursion can occur.
6. The user can give irreducible rational representations as extra input. This can help in that the tensor product of such representations with each other or with easily constructed representations within the algorithm may yield the desired representations. The algorithm itself can also construct easy representations at the beginning, such as linear representations; these may give some other representations for free. Going further, for each absolutely irreducible linear character χ of G, one can instantly construct a representation affording χ, and then compute the restriction to scalars of this representation to $\mathbb{Q}$ to yield an irreducible rational representation of degree d, where d is $s_{\mathbb{Q}}(\chi)$ times the degree of the character field $\mathbb{Q}(\chi)$.

7. The user can also give explicit irreducible representations of subgroups of G, so that induction condensation will be automatically applied to such representations, thus avoiding the search for representations of subgroups to induce to G.
8. If the virtual representation with character ψ has very small degree (say under 100), then one can use the rational Meataxe to decompose the full $\mathbb{Q}G$-module directly, instead of using the condensation tools.
9. Note that induction condensation is very useful for condensing and thus decomposing a monomial rational representation of G (where the corresponding representation of H is linear); this occurs very often and the algorithm IntegralSpin will be applied to a space having half the degree of the equivalent permutation representation, thus potentially making the saturation, LLL and Seysen operations run much faster.
10. The advantage of using tensor condensation is that it sometimes yields representations at little cost without needing a search in many subgroups of G for suitable permutation or induced representations. It is easy to compute initially the tensor products of the irreducible rational characters of G and check whether a desired representation occurs in a reasonable tensor product and then compute the contributing representations first. It seems that as the composition length of G grows, then useful tensor representations occur more often (not only do the exact tensor products occur often, but condensation of reducible tensor products becomes more worth using). For the large database of quasi-simple representations presented later (see Chapter 9), we see that tensor products are only used occasionally for the construction of the final representation of G, because G has composition length at most 2. But the algorithm often uses induction of a representation ρ_H of a subgroup H and since H can have arbitrary composition length, tensor products are used more often in constructing the representations of the subgroups.
11. As each new representation is constructed, our implementation applies the algorithm for entry reduction of an integral representation (Sec. 1.10) if its degree is less than 100 since this often makes the representation have even smaller entries (for higher degree, it has less effect and may take a long time). Note that in the induction case, the representation of H typically has very small degree (often less than 10) and so the reduced version will be very sparse which helps control the entry size in the induced representation.
12. Note that if we have a choice of different kinds of condensation of the same degree d, then it is always better to use permutation condensation if possible, since it is much faster to set up and the basis underlying the final integral spin tends to have smaller entries (since it consists of only permutations of the original uncondensed vectors). So in our implementation, permutation condensation of degree d is preferred over induction condensation of degree d. Since tensor condensation is the most expensive method, it is weighted so that it is preferred even less when comparing degrees. Thus whenever we compare entries of the queue Q while sorting Q, we first multiply the degree of the full character of each entry by a weight W, depending on the type of the relevant condensation. The current implementation uses $W = 1$ for permutation condensation, $W = 1.2$ for induction condensation and $W = 2$ for tensor condensation.

Example 3.8.3. Let $G = \mathrm{Sz}(8)$. The irreducible rational representations of G have degrees

$$1, 28, 64, 91, 105, 195.$$

All of these representations are easily computed in one go by calling IRREDUCIBLERATIONALREPRESENTATIONS on all the irreducible rational characters. There are 4 calls to AUTOMATICCONDENSATION, in this order (note how the degree of the virtual representation increases each time):

1. The degree-64 representation is extracted from a degree-65 permutation representation of G (condensation dimension 2; 0.1s).
2. The degree-195 representation is extracted from the induction to G of a degree-6 representation of a degree-65 subgroup (condensation dimension 12; 2.7s).
3. The degree-91 and -105 representations are both extracted from a degree-520 permutation representation of G (condensation dimension 12 for both; 1.2s).
4. The degree-28 representation is extracted from the induction to G of a degree-2 representation of a degree-560 subgroup (condensation dimension 88; 1.0s).

The degree-105 and -195 representations have 2-digit entries in their defining matrices while the other representations have 1-digit entries. The total time taken is only 5.6s.

Example 3.8.4. Let G be the perfect group of order 115248 with centre of order 7 and label '$\mathrm{L}_3(2)\ 2^1\ 7^2\ \mathbf{C}\ 7^1$' in the notation of [HP89]. G has inequivalent irreducible rational representations of the following degrees:

$$1, 6, 6, 7, 8, 8, 32, 42, 48, 48, 96, 96, 96, 126, 126, 168, 168, 252, 294, 336, 336, 504.$$

The degree-32 and one of the degree-48 representations have Schur index 2, while all of the others have Schur index 1. All of these representations can be computed by calling IRREDUCIBLERATIONALREPRESENTATIONS on all the irreducible rational characters. Most of the representations are computed very easily (each in less than a second) by automatic condensation of small-degree permutation or induced representations (and one of the degree-126 representations can be constructed by condensation of the tensor product of representations of degree 6 and 42). The only really non-trivial calls of AUTOMATICCONDENSATION are the following (out of a total time of 73.1s):

1. Degree 336 (16.7s): computed by condensation of the induction to G of a representation ρ_H of degree 12 for an index-56 subgroup H (ρ_H computed recursively in only 0.1s via the exact tensor product of the restriction of scalars to $\mathbb{Q}$ of absolutely irreducible linear representations of H). The condensation subgroup had order 16, the condensed module $\tilde{M}$ had dimension 48 (setup 0.5s) and the condensed constituent $\tilde{S}$ had dimension 24 (Meataxe time 0.8s, needing a maximal order basis to split the endomorphism ring). The integral spin was as follows: initial basis via modular spin with degree-672 induced action in 0.2s, saturation in 3.0s, Hermite form in 1.9s, LLL reduction in 0.4s and Seysen reduction in 8.9s. The resulting representation's defining matrices have absolute maximum entry 288, with average 10.5. (Without using Seysen reduction the absolute maximum entry is 28966, with average 457.1.)
2. Degree 504 (48.8s): computed by condensation of the induction to G of a representation ρ_H of degree 18 for an index-49 subgroup H (ρ_H computed recursively in 0.3s via a degree-168 permutation representation of H). The condensation subgroup had

order 98, the condensed module $\tilde{M}$ had dimension 12 (setup 0.1s) and the condensed constituent $\tilde{S}$ had dimension 6 (Meataxe time 0.02s). The integral spin was as follows: initial basis via modular spin with degree-882 induced action in 0.4s, saturation in 2.1s, Hermite form in 2.5s, LLL reduction in 1.3s and Seysen reduction in 33.5s. The resulting representation's defining matrices have absolute maximum entry 91, with average 2.7 (Without using Seysen reduction the absolute maximum entry is 2407, with average 20.7.)

Example 3.8.5. The first table in Chapter 9 describes our database of irreducible ordinary representations of quasi-simple groups up to degree 250 (matching the classification of Hiss & Malle [HM02]). There are 669 representations in total, and the representations are always realized over a minimal field. Of these, 353 are rational representations, of which 323 were computed by the algorithm IrreducibleRationalRepresentations (the tag 'IRR' in the method field indicates that this algorithm was used; see Chapter 8 for more information). The different kinds of condensation used by the algorithm for these 323 representations are as follows:

- 196 representations were computed by permutation condensation [IRR perm].
- 124 representations were computed by induction condensation [IRR ind].
- 3 representations were computed by tensor condensation [IRR $\ldots \otimes \ldots$].

The 29 other rational representations were computed by other algorithms described later.

Example 3.8.6. One of the higher-degree irreducible rational representations which was constructed by IrreducibleRationalRepresentations is the degree-1485 rational representation of $G = \mathrm{A}_{12}$ (table entry on p. 183; this representation was subsequently used in constructing the degree-3344 representation of HN). The algorithm proceeded as follows. After inserting many possibilities into the priority queue, the best choice used automatic condensation of the induction to G of a degree-42 representation ρ_H of an index-66 subgroup H. First ρ_H was constructed by a recursive call in only 1.4s (from a degree-252 permutation representation of H), then AutomaticCondensation selected a subgroup K of order 256, with a corresponding condensed module $\tilde{M}$ of dimension 33 (0.8s). The rational Meataxe split out the desired dimension-16 submodule $\tilde{S}$ in 0.2s. The modular spin with parallel operations on the integral vectors then took 33s, and the saturation, Hermite form, LLL and Seysen operations on each 1485×2772 integral matrix took 1018s, 69s, 28s and 910s respectively. Computing the reduced action took 40s, for a total time of 2126s. The resulting representation is integral, with both image matrices having at most 2-digit entries and density 45%.

Some other examples of higher-degree rational irreducible representations which can be constructed by this algorithm (all with small integral entries) are:

- The degree-825 representation of HS in 72s (p. 181).
- The degree-1300 representation of ${}^2\mathrm{F}_4(2)'$ in 1.0h (p. 183).
- The degree-1750 representation of McL in 1.4h (p. 183).
- The degree-2024 representation of M_{23} in 1.3h (p. 184).
- The degree-2024 and -2227 representations of Co_2 in 1.2h and 6.1h (p. 184).

We thus see that IrreducibleRationalRepresentations can be very effective for representations of very high degree and even for groups which are very large.

3.9. Constructing Absolutely Irreducible Representations

We now present an algorithm to construct an absolutely irreducible representation affording a given character χ, by first forming the corresponding irreducible rational representation and then computing the reduced action on a reduced basis of a suitable eigenspace. The major challenge is to control the size of the entries in the result. First we give a heuristic subalgorithm to compute a suitable reduced basis of the eigenspace, such that the denominators of the resulting representation are as small as possible.

Algorithm REDUCEDBASISFORACTION($[v_1, \ldots, v_r]$)

INPUT:

- A basis $[v_1, \ldots, v_r]$ of a subspace S of F^n where F is a number field.

OUTPUT:

- A reduced basis $[w_1, \ldots, w_r]$ of S.

STEPS:

```
1. Write $F = \mathbb{Q}(\alpha)$, let $f$ be the minimal polynomial of $\alpha$ and let $d = \mathrm{Deg}_{\mathbb{Q}}(f)$.
   Let $\phi : F^n \to \mathbb{Q}^{dn}$ be the natural $\mathbb{Q}$-vector space isomorphism, viewing $F$ as a $\mathbb{Q}$-vector
   space with basis $[1, \alpha, \ldots, \alpha^{d-1}]$.
2. Let $S_{\mathbb{Q}}$ be the $(dr)$-dimensional subspace of $\mathbb{Q}^{dn}$ generated by
                 $\{\phi(v_i \cdot \alpha^j) : 0 \le j \le d-1, 1 \le i \le r\}$.
   Set $L := (l_1, \ldots, l_{dr})$ to a LLL-reduced basis of the saturation of $S_{\mathbb{Q}}$, sorted with the
   shortest vectors first.
3. Set $W_{\text{best}} := 0$, $E_{\text{best}} := [\infty : 1 \le i \le r]$.
   For $c := 1$ to 10 do:
   {
      If $c = 1$ then set $U := L$; otherwise set $U$ to a random shuffle of $L$.
      Write $U = [u_1, \ldots, u_{dr}]$.
      Let $1 \le i_1 < i_2 < \ldots < i_r \le dr$ be minimal such that
          $(w_1, \ldots, w_r) = (\phi^{-1}(u_{i_1}), \ldots, \phi^{-1}(u_{i_r}))$ is an $F$-basis.
      Let $M$ be the matrix whose rows are $\phi(w_i \cdot \alpha^j)$ for $0 \le j \le d-1, 1 \le i \le r$.
      Set $E :=$ ELEMENTARYDIVISORS($M$).
      If $E < E_{\text{best}}$ (using lexicographical order backwards) then:
      {
         Set $W_{\text{best}} := (w_1, \ldots, w_r)$.
         Set $E_{\text{best}} := E$.
         If $E = [1, \ldots, 1]$ then break out of the loop.
      }
   }
   Return $W_{\text{best}}$.
```

Proposition 3.9.1. *Subalgorithm* REDUCEDBASISFORACTION *is correct.*

Proof. Clearly Step 2 sets $S_{\mathbb{Q}}$ to the image under ϕ of S regarded as a vector space over $\mathbb{Q}$ and since the saturation and LLL operations only perform invertible row transformations over $\mathbb{Q}$, the vectors $l_1, \dots l_{dr}$ must form a $\mathbb{Q}$-basis of $S_{\mathbb{Q}}$. Thus in each execution of the loop in Step 3, there exist vectors $u_{i_1}, \dots, u_{i_r}$ whose inverse images under ϕ are F-independent, and W_{best} will be set to one of these, so the returned result is an F-basis of S. □

Remarks 3.9.2. The point of computing the elementary divisors each time is that for a given choice of $W = (w_1, \dots, w_r)$, if E is the list of elementary divisors of the corresponding integral matrix M, then by Prop. 1.7.11, the largest (last) entry d of E gives the denominator introduced into the reduced action matrix corresponding to a matrix X acting on M by multiplication on the right; in a moment we will apply this to the case that X is the expansion under $\mathcal{B}_{F/\mathbb{Q}}$ [Def. 1.6.1] of a matrix with entries in F. Having a small maximum elementary divisor d not only gives a small denominator, but tends also to reduce the numerators which occur also in the coefficients of the number field elements. If the initial basis L is sparse (close to orthogonal), then the first try often gives $d = 1$ and we break out of the loop immediately and the number field entries in the reduced action is usually sparse with small entries.

Based on this special basis reduction algorithm, the following algorithms allow the construction of an absolutely irreducible representation.

Algorithm SPLITBYEIGENSPACE(M, e)
INPUT:

- An irreducible $\mathbb{Q}G$-module M of dimension n.
- A matrix $e \in \text{End}_{\mathbb{Q}G}(M)$ with minimal polynomial $f(x) \in \mathbb{Q}[x]$ such that f is irreducible over $\mathbb{Q}$.

OUTPUT:

- A submodule S_F of $M_F = M^F$ of dimension $\frac{n}{\text{Deg}(f)}$, where F is the number field $\mathbb{Q}(\alpha)$ with the minimal polynomial of α equal to f.

STEPS:

```
1. Set $F$ to the number field $\mathbb{Q}(\alpha)$ where the minimal polynomial of $\alpha$ is $f$.
2. Set $[v_1, \dots, v_d]$ to a basis of the nullspace of $(e - \alpha) \in \mathcal{M}_n(F)$.
   Set $[w_1, \dots, w_d]$ := REDUCEDBASISFORACTION$([v_1, \dots, v_d])$.
3. Set $S_F$ to the submodule of $M_F = M^F$ whose basis as a vector space is $[w_1, \dots, w_d]$
   and return $S_F$.
```

Algorithm ABSOLUTELYIRREDUCIBLEREPRESENTATION(χ)
INPUT:

- An absolutely irreducible character $\chi \in \text{Irr}(G)$ for a finite group G.

OUTPUT:

- A representation $\rho : G \to \mathrm{GL}_n(F)$ affording χ, such that F is a minimal field for χ.

STEPS:

1. Set $\chi_{\mathbb{Q}}$ to $s_{\mathbb{Q}}(\chi) \cdot \mathrm{GalSum}_{\mathbb{Q}}(\chi)$ where $C = \mathbb{Q}(\chi)$ (so $\chi_{\mathbb{Q}}$ equals the element of $\mathrm{Irr}_{\mathbb{Q}}(G)$ which contains χ as a constituent).
   Set $[\rho_{\mathbb{Q}}] :=$ IRREDUCIBLERATIONALREPRESENTATIONS($[\chi_{\mathbb{Q}}]$).
2. Set E to the endomorphism ring of $\rho_{\mathbb{Q}}$.
   Set e to a generator of a maximal subfield of E which is isomorphic to $F = \mathbb{Q}(\alpha)$.
3. Let $M_{\mathbb{Q}}$ be the $\mathbb{Q}G$-module corresponding to $\rho_{\mathbb{Q}}$.
   Set $M_F :=$ SPLITBYEIGENSPACE($M_{\mathbb{Q}}, e$).
4. Let $\rho : G \to \mathrm{GL}_n(F)$ be the representation corresponding to M_F.
   Embed $\mathbb{Q}(\chi)$ in F via Lem. 1.5.4 so that the character of ρ equals χ, then return ρ.

Theorem 3.9.3. *Algorithms* SPLITBYEIGENSPACE *and* ABSOLUTELYIRREDUCIBLEREPRESENTATION *are correct.*

Proof. Algorithm SPLITBYEIGENSPACE applies Lem. 1.5.3 directly. For algorithm ABSOLUTELYIRREDUCIBLEREPRESENTATION, $\rho : G \to \mathrm{GL}_n(F)$ must afford an $F/\mathbb{Q}$-Galois conjugate χ' of χ and F is minimal for χ' by Cor. 1.5.5, so FINDCONJUGATE is passed correct input and the returned representation must afford χ and be realized over the minimal field F. □

Remarks 3.9.4. We note the following points on ABSOLUTELYIRREDUCIBLEREPRESENTATION and its implementation:

1. If the Schur index s of χ is 1, then the field F is essentially unique but if $s > 1$, then it is not unique, of course. As an option, one can specify a particular field F to be used in Step 3, based on an element of the endomorphism ring. In our implementation, we also have an option so that the rational representation $\rho_{\mathbb{Q}}$ may be passed in, since it may be first constructed by other means, of course.
2. The quality of the resulting representation depends very strongly on how reduced (close to orthogonal) the initial reduced integral basis L is, in the subalgorithm REDUCEDBASISFORACTION. If the basis is sparse and highly reduced (which often happens when the rational representation $\rho_{\mathbb{Q}}$ is very sparse), then the resulting complex representation will tend to have high quality. But it is often the case that the basis L cannot be reduced much, even when the rational representation $\rho_{\mathbb{Q}}$ is sparse (and hardly ever when $\rho_{\mathbb{Q}}$ is only moderately dense and its degree is above 100). So the major limitation of the algorithm is that even after much searching in REDUCEDBASISFORACTION for the best reduced basis, the corresponding reduced action over F may still have very large entries (and take a long time to compute). See Ex. 3.9.7 below for an example.
3. The embedding of $\mathbb{Q}(\chi)$ in F in Step 4 typically takes very little time. Just as for the general algorithm in Sec. 3.3 for computing the character of a representation, we can

of course first evaluate traces of ρ evaluated at the generators of G, then products of these and general random elements, and when enough class representatives are found which determine the correct images for the embedding, then the algorithm can exit early instead of having to evaluate ρ at all the class representatives. In practice, this algorithm typically only takes a second or two even in high dimensions, since only a very small number of evaluations are needed to determine the correct embedding.

Example 3.9.5. Let $G = 6.\mathrm{A}_7$ and let χ be one of the absolutely irreducible characters of G of degree 36; χ has character field $F = \mathbb{Q}(\zeta_3)$ and Schur index 1. In [DD10, Sec. 2], the authors found it difficult to construct a representation affording χ using their methods. But we can construct it easily using AbsolutelyIrreducibleRepresentation in under 3 seconds, as follows (table entry on p. 163). The initial call to IrreducibleRationalRepresentations on the irreducible rational character containing χ yields a degree-72 representation $\rho_{\mathbb{Q}}$ over $\mathbb{Z}$ in 2.6s (derived from the induction to G of a degree-8 integral representation of a subgroup of index 21), with absolute maximum entry 7. It then takes only 0.3s to do the remaining Steps 2 to 4 of AbsolutelyIrreducibleRepresentation, as follows:

- The element e in the endomorphism ring of $\rho_{\mathbb{Q}}$ is instantly found, with minimal polynomial $x^2 + x + 1$; e has density 36.6% and absolute maximum entry 6.
- The dimension-36 nullspace $N \subset F^{72}$ of $(e - \alpha) \in \mathcal{M}_{72}(F)$ is computed in Step 2 of SplitByEigenspace.
- In Step 4 of ReducedBasisForAction, $S \in \mathcal{M}_{72\times 144}(\mathbb{Z})$ is set to the saturation of the expansion of N.
- The LLL-reduced basis L of the rows of S has vectors in $\mathbb{Z}^{144}$ whose norms range from 9 to 52, so the basis is rather sparse. The loop in Step 7 immediately finds that the lexicographically-first subset of L which yields an F-independent set has maximum elementary divisor 1, so the loop is exited immediately and then the reduced action on the basis is computed.
- The computation of the embedding of $\mathbb{Q}(\chi)$ in F needs an evaluation at one conjugacy class (not covered by the generators).

The resulting images of the generators have density 84% and 74% respectively and all entries have the form $a + b\zeta_3$, with $a, b \in \mathbb{Z}$, $|a|, |b| \leq 17$ and the denominator of all entries is 1 since the maximum elementary divisor of the basis in ReducedBasisForAction was 1.

Example 3.9.6. For the first table in Chapter 9 describing irreducible representations of quasi-simple groups up to degree 250 there are 669 representations in total. Of these, there are 316 irrational representations and 117 of these were computed by the algorithm AbsolutelyIrreducibleRepresentation since it returned a representation with very small entries (the tag 'AIR' in the method field indicates that this algorithm was used; see Chapter 8 for more information). The different kinds of condensation used in the initial call to IrreducibleRationalRepresentations in Step 1 for these 117 representations are as follows (with the corresponding tag in the table given in brackets):

- 19 representations were computed by permutation condensation [AIR perm].
- 97 representations were computed by induction condensation [AIR ind].
- 1 representation was computed by tensor condensation [AIR $\ldots \otimes \ldots$].

A large example is the degree-216 representation over $\mathbb{Q}(\sqrt{-1})$ of $2.\mathrm{J}_2$ for which the character is rational, but has Schur index 2 (table entry on p. 175). The absolute maximum numerator is 187 and the denominator LCM is 1 (the density of both generators is 38%).

Example 3.9.7. Let G be the sporadic simple group J_3. A minimal-degree faithful representation of G has degree 85 and can be realized over the quadratic field $F = \mathbb{Q}(\sqrt{-19})$. Let χ be one of the corresponding characters. If we call AbsolutelyIrreducibleRepresentation on G and χ to construct a representation ρ affording χ, then the initial construction of the corresponding degree-170 irreducible rational representation $\rho_{\mathbb{Q}}$ is not difficult (via condensation of a degree-14688 permutation representation) and takes 206s. But when the rest of the algorithm constructs an absolutely irreducible representation $\rho : G \to \mathrm{GL}_{85}(F)$ affording χ (in 79s), the resulting image matrices have entries with 73-digit numerators (and denominator 1); further searching in ReducedBasisForAction hardly improves this. So this is a case where the algorithm cannot construct a reasonably reduced representation. But we will later see that the hybrid algorithm of Chapter 6 can construct a representation affording χ with very small entries and in much less time; see Ex. 6.4.1 (p. 138).

Similar examples are the degree-80 faithful irreducible representations of $4_1.\mathrm{L}_3(4)$ and $4_2.\mathrm{L}_3(4)$, which are both realized over a minimal field of degree 4. The algorithm AbsolutelyIrreducibleRepresentation can only produce representations with 93-digit and 89-digit numerators (denominator 1) respectively, taking about an hour in each case. Again, the hybrid algorithm will easily construct appropriate representations with small entries in very little time (the results are on p. 167).

3.10. Constructing Irreducible Representations over a Given Field

The following algorithm computes F-irreducible representations of a group G for any given number field F which is normal over $\mathbb{Q}$. This algorithm will have important application in the extension-based algorithms later.

Algorithm IrreducibleRepresentationsOverField($[\chi_1, \ldots, \chi_k], F$)

Input:

- Characters $[\chi_1, \ldots, \chi_k]$ of a finite group G and a field F which is normal over $\mathbb{Q}$, such that $\chi_i \in \mathrm{Irr}_F(G)$ for $1 \le i \le k$.

Output:

- F-representations $[\rho_1, \ldots, \rho_k]$ such that $\rho_i : G \to \mathrm{GL}_{n_i}(F)$ affords χ_i for $1 \le i \le k$.

Steps:

```
1. For $1 \le i \le k$, set $\psi_i$ to the element of $\mathrm{Irr}_{\mathbb{Q}}(G)$ which contains $\chi_i$.
2. Collect distinct elements of $[\psi_1, \ldots, \psi_k]$ and then call IrreducibleRationalRepresentations on these to obtain rational representations $\sigma_1, \ldots, \sigma_k$ which afford $\psi_1, \ldots, \psi_k$ respectively.
3. For $i := 1$ to $k$ do:
   {
      Set $r := \mathrm{Deg}(\chi_i)$, $n := \mathrm{Deg}(\sigma_i)$, $d := \frac{n}{r}$.
```

```
        If $d = 1$ then set $\rho_i := (\sigma_i)^F$ and skip to the next $i$.
        Let $M$ be the $\mathbb{Q}G$-module corresponding to $\sigma_i$.
        Search for an $e \in \mathrm{End}_{\mathbb{Q}G}(M)$ with minimal polynomial $f_e \in \mathbb{Q}[x]$ of degree $d$ such that
            $f_e$ has a root in $F$ (first try each element of a basis $B$, and then 100 linear
            combinations with coefficients in $[-10 \ldots 10]$ of the elements of $B$).
        If such an $e$ is found then:
        {
            Set $M_S :=$ SPLITBYEIGENSPACE$(M, e)$ (written over $S = \mathbb{Q}(\beta)$, $f_e(\beta) = 0$).
            Let $\sigma_S$ be the representation corresponding to $M_S$.
            Let $\chi_S$ be the character of $\sigma_S$ and let $\phi$ be the embedding of $\mathbb{Q}(\chi_S)$ into
                $\mathbb{Q}(\chi_i)$ (a subfield of $F$) so that $\chi_S$ equals $\chi_i$ under this embedding
                (as in Lem. 1.5.4).
            Embed $S$ into $F$ so that the embedding equals $\phi$ on the subfield $\mathbb{Q}(\chi_S)$ and
                then let $\rho_i : G \to \mathrm{GL}_r(F)$ equal $\sigma_S$ lifted to $F$ via this embedding.
        }
        Else:
        {
            If $\chi_i$ is absolutely irreducible then:
            {
                Set $\sigma :=$ ABSOLUTELYIRREDUCIBLEREPRESENTATION$(\chi)$.
                Set $\rho_i$ to a representation over $F$ which is equivalent to $\sigma$ by Fieker's
                    algorithm [Subsec. 2.3.4].
            }
            Else:
            {
                Set $m := \mathrm{Deg}_{\mathbb{Q}}(F) \cdot \frac{r}{n}$ and $M_m := \oplus_{i=1}^{m} M$.
                Search for an $e \in \mathrm{End}_{\mathbb{Q}G}(M_m)$ which generates a subfield $S$ isomorphic
                    to $F$ by exhaustive search with increasing integral coordinates w.r.t. a basis.
                Set $M_S :=$ SPLITBYEIGENSPACE$(M_m, e)$.
                Let $\sigma_S$ be the representation corresponding to $M_S$.
                Let $\chi_S$ be the character of $\sigma_S$ and let $\phi$ be the embedding of $\mathbb{Q}(\chi_S)$ into
                    $\mathbb{Q}(\chi_i)$ so that $\chi_S$ equals $\chi_i$ under this embedding (as in Lem. 1.5.4).
                Embed $S$ into $F$ so that the embedding equals $\phi$ on the subfield $\mathbb{Q}(\chi_S)$ and
                    then let $\rho_i : G \to \mathrm{GL}_r(F)$ equal $\sigma_S$ lifted to $F$ via this embedding.
            }
        }
    }
4. Return $[\rho_1, \ldots, \rho_k]$.
```

Theorem 3.10.1. *Algorithm* IRREDUCIBLEREPRESENTATIONSOVERFIELD *is correct.*

Proof. After Step 2, for each i with $1 \leq i \leq k$, σ_i affords ψ_i, where ψ_i is the irreducible rational character containing χ_i. We now show that for each i, the body of the loop in Step 3 sets ρ_i to an F-representation affording χ_i. Fix such an i.

First note that since F is normal over $\mathbb{Q}$, $\mathrm{GalSum}_{F/\mathbb{Q}}(\chi_i)$ equals an integer multiple of ψ_i, and it it is easy to see that for any integer $m \geq 1$, any character in $\mathrm{Irr}_F(G)$ which is a constituent of $m \cdot \psi_i$ must be an $(F/\mathbb{Q})$-conjugate of χ_i. Thus for any constituent of $(m \cdot \sigma_i)^F$ (for $m \geq 1$) which has degree $r = \chi_i(1)$, its character must be an $(F/\mathbb{Q})$-conjugate of χ.

Suppose first that the first case is taken in the main if-statement, so an endomorphism e is found with minimal polynomial f_e of degree $d = \frac{n}{r}$ ($n = \psi_i(1)$, $r = \chi_i(1)$), where f_e has a root in F, and e generates a subfield which is isomorphic to S which can be embedded into F. Then by Lemma 1.5.3, the constructed σ_S has degree $\frac{n}{d} = r$ and under any choice of embedding of S into F such that the character of σ_S embeds into $\mathbb{Q}(\chi_i)$, $(\sigma_S)^F$ has degree r and so will have character $(F/\mathbb{Q})$-conjugate to χ_i by the observation of the previous paragraph. Thus under a suitable choice of embedding, $\rho_i = (\sigma_S)^F$ affords χ_i.

The else-part of the main if-statement is executed when no such subfield S can be found after some searching (it may not exist in general). In the case that χ_i is absolutely irreducible, then clearly AbsolutelyIrreducibleRepresentation will return σ affording χ over some field and Fieker's algorithm will rewrite this to be over F. For the final case, there must exist some representation ρ_1 over F which affords χ_i, by the assumptions on the input. Now if $\rho_{\mathbb{Q}}$ is the restriction of scalars representation of ρ_1 from F to $\mathbb{Q}$, then $\rho_{\mathbb{Q}}$ is a homogeneous rational representation of degree $r \cdot \mathrm{Deg}_{\mathbb{Q}}(F)$ and must have character $m \cdot \psi_i$, where $m = \frac{r}{n} \cdot \mathrm{Deg}_{\mathbb{Q}}(F)$ (by Prop. 1.6.2), and the endomorphism ring of $\rho_{\mathbb{Q}}$ must contain a subfield isomorphic to F. By construction, the character of the representation corresponding to M_m equals the character of $\rho_{\mathbb{Q}}$, so the search for the subfield S in the endomorphism ring of M_m must eventually succeed. The remaining statements are similar to the first case above and clearly set up a corresponding F-representation ρ_i which affords χ_i. □

Remarks 3.10.2. We note the following points on the implementation:

1. This algorithm couples well with a single call to IrreducibleRationalRepresentations when there are several representations to construct, since that algorithm does only one search to construct all the representations (and some may be easily derived from others via tensor products).
2. It is worth checking first for each χ_i whether a representation can be constructed by direct induction from a representation from a subgroup (and one can then call the algorithm recursively on a smaller degree character for a proper subgroup).
3. When more than one desired F-representations are constituents of the same irreducible rational representation, then after the first one is constructed, the other ones can of course just be computed as conjugates, instead of doing the body of the loop in Step 3 again each time.
4. The former case in Step 3 nearly always happens for the applications we have made of this algorithm: one can nearly always find an endomorphism generating a subfield of the right degree which can be embedded into F. One should also use the basis of a maximal order of the endomorphism ring to find endomorphisms with small entries. The second case arises occasionally when a Schur index $s_{\mathbb{Q}}(\chi_i)$ is non-trivial; an example of this situation will be seen later in Ex. 6.4 (p. 140). One could also use methods based

on solving conics instead of Fieker's algorithm to find suitable endomorphisms in the last case where χ_i is not absolutely irreducible.

Examples of the use of this algorithm will be given later where it is needed in the extension-based algorithms, where several irreducible representations over a given field F may need to be computed, where F is intermediate between $\mathbb{Q}$ and a minimal field for an absolutely irreducible representation.

3.11. Rewriting a Representation over a Minimal Field

A simple modification of AbsolutelyIrreducibleRepresentation also yields the following straightforward algorithm to rewrite a given absolutely irreducible representation over a minimal field.

Algorithm RewriteOverMinimalField(ρ_0)
Input:

- An absolutely irreducible representation $\rho_0 : G \longrightarrow \mathrm{GL}_n(F_0)$ of a finite group G affording χ, where F_0 is not necessarily minimal for χ.

Output:

- An equivalent representation $\rho : G \longrightarrow \mathrm{GL}_n(F)$ affording χ, such that F is a minimal field for χ.

Steps:

1. Let $\rho_{\mathbb{Q}}$ be the restriction of scalars representation of ρ_0 from F_0 to $\mathbb{Q}$ (using $\mathcal{B}_{F_0/\mathbb{Q}}$, as in Prop. 1.6.2).
   Let $M_{\mathbb{Q}}$ be the $\mathbb{Q}G$-module corresponding to $\rho_{\mathbb{Q}}$.
   Set $[S_1, \ldots, S_m] :=$ RationalMeataxe($M_{\mathbb{Q}}$).
2. Set E to the endomorphism ring of S_1.
   Set e to a generator of a maximal subfield F of E.
   Set $M :=$ SplitByEigenspace(S_1, e).
   Let $\rho : G \longrightarrow \mathrm{GL}_n(F)$ be the representation corresponding to M.
   Embed $\mathbb{Q}(\chi)$ in F via Lem. 1.5.4 so that the character of ρ equals χ, then return ρ.

Proposition 3.11.1. *Algorithm* RewriteOverMinimalField *is correct.*

Proof. Since ρ_0 is absolutely irreducible, $M_{\mathbb{Q}}$ must be homogeneous and so equal the sum of m copies of a simple $\mathbb{Q}G$-module. Thus the character of $M_{\mathbb{Q}}$ equals $m\chi_{\mathbb{Q}}$ for some $\chi_{\mathbb{Q}} \in \mathrm{Irr}_{\mathbb{Q}}(G)$, so after Step 1, the character of S_1 must be $\chi_{\mathbb{Q}}$. Then we can apply Cor. 1.5.5 again and are in the same situation as algorithm AbsolutelyIrreducibleRepresentation, so Steps 2 and 3 proceed the same as in that algorithm. □

Remarks 3.11.2. 1. This algorithm works extremely well in practice when the minimal field F does not have very large degree, thus avoiding the non-trivial number theory which is needed in Fieker's method.

2. Instead of the call to SPLITBYEIGENSPACE, we will give an alternative method below (p. 128) which can be used when the degree is large or SPLITBYEIGENSPACE does not give a result with small entries.

3.12. Conclusion

We summarize the main features of the condensation-based splitting approach. Some of the key advantages are the following:

1. For computing irreducible rational representations of rather high degree (say up to degree 1000), this method yields an integral representation with very small entries in practice, even when the virtual representation σ from which the constituents are extracted has degree up to about 100,000.
2. The method is completely automatic and guarantees that the resulting representation(s) are always realized over a minimal field (because the corresponding rational representations are irreducible). It does not require an initial choice of a suitable subgroup H which is required by the extension-based algorithms (in the following chapters).
3. When one needs several irreducible F-representations of G, then the splitting approach can often construct them together easily (e.g., several representations can be extracted from the one virtual representation, and tensor products can yield representations for free) and this can be much more efficient than using the extension-based methods below separately for each representation.

Some of the limitations are the following:

1. If G has no proper subgroups of moderate index, then one cannot find a representation σ which it is feasible to split, so this method fails.
2. If χ has very high degree (say over 1000), then the operations on integral matrices to compute the reduced basis in the integral spin algorithm (saturation, Hermite form, LLL, Seysen) become very expensive.
3. If the final representation ρ cannot be realized over $\mathbb{Q}$, then it may be impossible to find a reduced basis of the eigenspace over the number field so that ρ has reasonably reduced entries, even when the degree is rather small. So this method often fails to construct irrational representations with reasonably small entries.

CHAPTER 4

Irreducible Extension

4.1. Introduction

In this chapter we start to describe the extension approach, considering first the case of irreducible extension. We show how a well-known algorithm for irreducible extension, based on linear algebra, can be made very efficient. Several important techniques which are developed here will be again used in the next chapter in the algorithm for general extension.

4.2. Existing Methods

Let χ be an absolutely irreducible character of a finite group G. Suppose that H is a subgroup of G such that $\chi_H = \chi \downarrow_H$ is also absolutely irreducible and suppose that $\rho_H : H \to \mathrm{GL}_n(F)$ affords χ_H, where $F = F(\chi)$. Then ρ_H can be uniquely extended to a representation $\rho : G \to \mathrm{GL}_n(F)$ affording χ, so that $\rho \downarrow_H = \rho_H$. We call this operation **irreducible extension**.

Minkwitz presented the following explicit formula for irreducible extension, which involves looping over the subgroup H.

Theorem 4.2.1. [Min96, Thm. 1] *Let $\chi \in \mathrm{Irr}(G)$ and let H be a subgroup of G such that $\chi_H = \chi \downarrow_H$ is absolutely irreducible and suppose that $\rho_H : H \to \mathrm{GL}_n(F)$ affords χ_H. Let $E = F(\chi)$ and define a representation $\rho : G \to \mathrm{GL}_n(E)$ of G by:*

$$\rho(g) := \frac{\chi(1)}{|H|} \sum_{h \in H} \chi(h^{-1}g)\rho_H(h) \quad \textit{for } g \in G.$$

Then ρ affords χ and $\rho \downarrow_H = \rho_H$. Thus given a representation ρ_H affording χ_H, one can construct a representation ρ affording χ with $\rho \downarrow_H = \rho_H$ by evaluating the above sum for elements $\{g_1, \ldots, g_k\}$ of G where $G = \langle H, g_1, \ldots, g_k \}$.

The obvious practical limitation of this formula is that it requires the evaluation of ρ_H at every element of H, so it can only be used when H is rather small. Grassl constructed some representations up to degree 124 using this formula for some large groups [Gra06], but the computations took a very long time for larger examples (e.g., a degree-78 absolutely irreducible representation of Fi_{22} was constructed as the extension of a degree-78 representation of $\mathrm{G}_2(3)$ in about 40 hours).

Plesken & Souvignier [PS98, 3.1] proposed an alternative method which does not require looping over the subgroup H, but involves writing the image of $g \in G$ as a linear combination of n^2 images of elements of H under ρ_H. An equivalent formulation based on linear algebra was given by Dabbaghian-Abdoly as follows.

Theorem 4.2.2. [DA05, 2.2–2.3] *Let $\chi \in \mathrm{Irr}(G)$ and let H be a subgroup of G such that $\chi_H = \chi \downarrow_H$ is absolutely irreducible and suppose that ρ_H affords χ_H. Let n be the degree of χ. By a theorem of Burnside there exist $w_1, \ldots, w_{n^2} \in H$ such that $\{\rho_H(w_1), \ldots, \rho_H(w_{n^2})\}$ is a basis for the full matrix algebra $\mathcal{M}_n(F)$. Then ρ_H can be extended uniquely to a representation ρ of G affording χ and the entries of $\rho(g)$ for $g \in G$ are determined by the equations:*

$$\chi(w_k g) = \mathrm{Tr}(\rho_H(w_k)\rho(g)) \quad \text{for } k = 1, \ldots n^2.$$

Furthermore, on average, selection of at most $2n^2$ random elements of H yield a corresponding basis (or equivalently, yield enough relations from the above formula involving traces to determine $\rho(g)$ uniquely for any $g \in G$).

4.3. Using a Normalized Subgroup

W. Unger [Ung10] noted that the linear irreducible extension method can be greatly improved by using a subgroup L of H which is normalized by an element of g outside of H (this idea was motivated by the use of normalizers in [Wil99]). The basic idea is given in the following lemma, and immediately suggests the auxiliary algorithm which follows.

Proposition 4.3.1. [CR81, 9.24] *Let χ_1, χ_2 be characters for G which are afforded by $\rho_1 : G \to \mathrm{GL}_{n_1}(F)$ and $\rho_2 : G \to \mathrm{GL}_{n_2}(F)$ respectively. Then $\mathrm{Dim}_F(\mathrm{Hom}_{FG}(\rho_1, \rho_2)) = \langle \chi_1, \chi_2 \rangle_G$ (the inner product of χ_1 and χ_2).*

Lemma 4.3.2. *Suppose that χ is a character of G (not necessarily irreducible), F is a field over which χ may be realized, H is a subgroup of G, $g \in G$, $G = \langle H, g \rangle$ and $\rho_H : H \to \mathrm{GL}_n(F)$ affords $\chi_H = \chi \downarrow_H$. Suppose also that L is a subgroup of H such that $L^g = L$ (i.e., g normalizes L). Let $\rho_L = (\rho_H) \downarrow_L$ and define a new representation $\rho'_L : L \to \mathrm{GL}_n(F)$ by*

$$\rho'_L(x) := \rho_L(x^g).$$

Then if ρ is any extension of ρ_H to G which affords χ, then

$$\rho(g) \in \mathrm{Hom}_{FL}(\rho_L, \rho'_L).$$

Also, the dimension of this Hom-module as an F-vector space equals $\langle \chi \downarrow_L, \chi \downarrow_L \rangle_L$ (the norm of $\chi \downarrow_L$ w.r.t. L).

Proof. For any $x \in L$, we have

$$\rho'_L(x) = \rho_L(x^g) = \rho(x^g) = \rho(g^{-1}xg) = \rho(g)^{-1}\rho_L(x)\rho(g),$$

so $\rho(g)$ is in $\mathrm{Hom}_{FL}(\rho_L, \rho'_L)$. The statement on the dimension follows from Prop. 4.3.1. □

Remarks 4.3.3. Note that taking L to be the trivial group reduces to the original method: in this case, $\mathrm{Hom}_{FL}(\rho_L, \rho'_L)$ has dimension n^2 with basis consisting of the unit matrices, where n is the degree of χ.

Algorithm ExtensionImageSetup(G, ρ_H)
Input:

- A finite group G and a representation $\rho_H : H \to \mathrm{GL}_n(F)$ for a maximal subgroup H of a group G and a field F (where ρ_H is not necessarily irreducible over F).

OUTPUT:

- An element $g \in G \setminus H$ and matrices $[A_1, \dots, A_l] \in \mathcal{M}_n(F)$ such that for any representation $\rho : G \to \mathrm{GL}_n(F)$ with $\rho \downarrow_H = \rho_H$, $\rho(g)$ must equal an F-linear combination of the A_i.

STEPS:

1. Set L to a subgroup of H with largest possible order such that $N_G(L) \not\subseteq H$ and set g to an element of $N_G(L) \setminus H$.
2. Set ρ_L to $(\rho_H) \downarrow_L$ and define a new representation $\rho'_L : L \to \mathrm{GL}_n(F)$ by
$$\rho'_L(x) = \rho_L(x^g).$$
3. Set $[A_1, \dots, A_l]$ to an echelonized basis of $\mathrm{Hom}_{FL}(\rho_L, \rho'_L)$ (as matrices acting on the standard basis of the natural module corresponding to ρ_H). Return g and $[A_1, \dots, A_l]$.

Remarks 4.3.4. We note the following points on the implementation:

1. It is highly desirable to minimize the dimension of the Hom-module associated to L, since this directly affects the cost of later algorithms. So instead of stopping at the first valid L, one could loop over all subgroups of S and for each potential L for which there is a normalizing element outside H, one could compute the corresponding dimension as $\langle \chi \downarrow_L, \chi \downarrow_L \rangle_L$ and choose an L for which the corresponding dimension is minimal. However, this may be very expensive for larger groups (mainly because computing $\chi \downarrow_L$ involves setting up the fusion of classes of L in H) so in such a case, we simply choose the first valid L (proceeding from biggest to smallest) and stop immediately, as in the algorithm.
2. The other major issue is the cost of computing the normalizer $N_G(L)$. For permutation groups, MAGMA has an efficient backtrack search algorithm, so it is not a major issue here. But for matrix groups, computing the normalizer is a much harder problem, and is currently impossible if one cannot compute a base and strong generating set (BSGS) for G. So we will later describe an advanced version of this algorithm (in Subsec. 5.4.8) which does not need a BSGS for G and so will be suitable for the large sporadic simple groups which have to be defined in practice by large-degree matrix groups over finite fields.
3. The Hom-module can be computed efficiently using the algorithm described on p. 23, even when F is a number field. In this chapter, ρ_H will always be irreducible, but in the next chapter this algorithm will be applied to a representation ρ_H which is not necessarily irreducible over F but may be a block diagonal sum of irreducible F-representations. In this situation, the restriction of ρ_H to L preserves the block structure, so we note that the computation of the basis of homomorphisms can be sped up greatly by exploiting the block structure of ρ_L (and the resulting matrices can also be returned in block form).

4.4. The Irreducible Extension Algorithm

We can now present the improved version of the linear algebra-based algorithm to extend an irreducible representation of a subgroup H to one for G. We first separate out a subalgorithm LinearTraceReduction to gather linear relations based on random elements of H; since this subalgorithm will also be used in the next chapter in the case of general extension, it does not require that the character χ of G is irreducible.

Algorithm LinearTraceReduction($\chi, \rho_H, g, [A_1, \dots, A_l]$, `MaxTries`)
Input:

- A character χ (not necessarily irreducible) of a finite group G.
- A representation ρ_H affording $\chi \downarrow_H$, where H is a subgroup of G.
- An element $g \in G$ with $G = \langle H, g \rangle$.
- Matrices $[A_1, \dots, A_l] \in \mathcal{M}_n(F)$ such that for any representation $\rho : G \to \mathrm{GL}_n(F)$ which affords χ with $\rho \downarrow_H = \rho_H$, $\rho(g)$ must equal an F-linear combination of the A_i.
- A stopping limit `MaxTries` (which may equal ∞ if ρ_H is absolutely irreducible).

Output:

- Matrices $[A_0, A_1, \dots, A_k] \in \mathcal{M}_n(F)$ such that for any representation $\rho : G \to \mathrm{GL}_n(F)$ which affords χ with $\rho \downarrow_H = \rho_H$, $\rho(g)$ must equal A_0 plus an F-linear combination of $[A_1, \dots, A_k]$. (The algorithm proceeds until $k = 0$ or there are `MaxTries` consecutive random elements of H which give no new independent relations.)

Steps:

```
1. Set $C := 0$ and set $A_0 := 0 \in \mathcal{M}_n(F)$. Set $k := l$.
2. Loop forever:
   {
      Set $h$ to a random element of $H$ and $B := \rho_H(h)$.
      Set $c_0 := \chi(h \cdot g)$ and $c_i := \mathrm{Tr}(B \cdot A_i)$ for $1 \le i \le k$.
      [This implies the linear relation $\sum_{i=1}^{k} c_i \cdot x_i = c_0$.]
      If $c_i = 0$ for all $i$ with $1 \le i \le k$ then:
      {
         Assert that $c_0 = 0$ (as a check; this relation yields nothing).
         Set $C := C + 1$.
         If $C =$ MaxTries then break out of the loop.
         Skip to the top of the loop.
      }
      Let $l$ be maximal such that $c_l \neq 0$.
      [The relation can be written $x_l = \frac{1}{c_l}(c_0 - \sum_{i=1}^{l-1} c_i x_i)$.]
      Set $A_0 := A_0 + \frac{c_0}{c_l} A_l$.
      For $i := 1$ to $l - 1$ do: set $A_i := A_i - \frac{c_i}{c_l} A_l$.
      Set $[A_1, \dots, A_k] := [A_1, \dots, A_{l-1}, A_{l+1}, \dots A_k]$ and set $k := k - 1$.
      If $k = 0$ then break out of the loop.
   }
3. Return $[A_0, A_1, \dots, A_k]$.
```

```
Algorithm IRREDUCIBLEEXTENSION($\chi, \rho_H$)
INPUT:
  • An absolutely irreducible character $\chi$ for a finite group $G$.
  • A representation $\rho_H : H \to \mathrm{GL}_n(F)$ affording $\chi \downarrow_H$ and such that $\chi \downarrow_H$ is absolutely
    irreducible, where $H$ is a maximal subgroup of $G$ and $F$ is a field with $F(\chi) = F$.
OUTPUT:
  • A representation $\rho : G \to \mathrm{GL}_n(F)$ of $G$ affording $\chi$, such that $\rho \downarrow_H$ equals $\rho_H$.
STEPS:
  1. Set $g, [A_1, \dots, A_l] :=$ EXTENSIONIMAGESETUP($G, \rho_H$).
  2. Set $[A_0, A_1, \dots, A_k] :=$ LINEARTRACEREDUCTION($\chi, \rho_H, g, [A_1, \dots, A_l], \infty$).
     Assert that $k = 0$.
  3. Define $\rho : G \to \mathrm{GL}_n(F)$ via $\rho(h) = \rho_H(h)$ by $h \in H$ and $\rho(g) = A_0$ and return $\rho$.
```

Theorem 4.4.1. *Algorithms* LINEARTRACEREDUCTION *and* IRREDUCIBLEEXTENSION *are correct.*

Proof. By Lem. 4.3.2, EXTENSIONIMAGESETUP is correct and the input to LINEARTRACEREDUCTION is correct. For the correctness of LINEARTRACEREDUCTION, first write $X = A_0 + \sum_{i=1}^{k} x_i \cdot A_i$ for indeterminates $x_1, \dots, x_k$. It it easy to see that the following condition is an invariant of the main loop: for any representation $\rho : G \to \mathrm{GL}_n(F)$ which affords χ with $\rho \downarrow_H = \rho_H$, $\rho(g)$ must equal X for some assignment of the x_i to elements of F. The condition is initially satisfied because of the input condition on the initial value of $[A_1, \dots, A_k]$ and the fact that $A_0 = 0$. Within the loop, each time a linear relation $\sum_{i=1}^{k} c_i \cdot x_i = c_0$ is constructed, it clearly gives a necessary condition on the x_i (cf. Thm 4.2.2). If the relation is non-zero, then it can be written in the form:

$$x_l = \frac{1}{c_l}(c_0 - \sum_{i=1}^{l-1} c_i x_i),$$

so the term $x_l \cdot A_l$ in the sum defining X can be expanded as follows:

$$\begin{aligned} X &= A_0 + \sum_{i=1}^{l-1} x_i \cdot A_i + \frac{1}{c_l}(c_0 - \sum_{j=1}^{l-1} c_j x_j) A_l + \sum_{i=l+1}^{k} x_i \cdot A_i \\ &= A_0 + \frac{c_0}{c_l} \cdot A_l + \sum_{i=1}^{l-1} x_i \cdot (A_i - \frac{c_i}{c_l} \cdot A_l) + \sum_{i=l+1}^{k} x_i \cdot A_i. \end{aligned}$$

Thus after replacing A_0 by $A_0 + \frac{c_0}{c_l} A_l$, A_i by $A_i - \frac{c_i}{c_l} A_l$ for $1 \le i < l$, and then deleting A_l and decreasing k, the newly defined X based on the new A_i clearly preserves the loop invariant. This invariant implies that the matrices returned by LINEARTRACEREDUCTION satisfy the condition on the output. For termination, note that if the bound `MaxTries` is finite, then LINEARTRACEREDUCTION trivially terminates (this situation will be used in the general extension algorithm). Otherwise, we can assume that ρ_H is absolutely irreducible, so by Thm. 4.2.2, we will eventually reduce to the case that $k = 0$ (the initial basis can be considered as equivalent to a full basis of the image of ρ_H with some associated initial linear relations and so the expected number of tries is at most $2n^2$ on average). Thus A_0 will give the unique image of g defining ρ. This proves the correctness of IRREDUCIBLEEXTENSION. □

Remarks 4.4.2. We note the following points on the implementation:

1. The main advantage of this algorithm over previous forms of the linear algebra-based algorithm is that if there are k initial image matrices, then they effectively give $n^2 - k$ initial independent linear relations on the n^2·coordinates in the image matrix of g, so there are only k more independent relations to be found instead of n^2.
2. The algorithm as stated requires that H is a maximal subgroup of G. But if there is an arbitrary proper subgroup H of G for which $\chi \downarrow_H$ is absolutely irreducible, then one can simply apply the algorithm iteratively up a chain of subgroups to H to G for which each subgroup is maximal in the next one; the intermediate representations must all be absolutely irreducible too. In our implementation, we can either compute the maximal subgroups of G very quickly using the MAGMA implementation of the algorithm given in [CH04], or for the very large quasi-simple groups, we can use the words provided in the online ATLAS [WWT+].
3. If H is normal in G, then we may let $L = H$ and g be one of the given generators of G which is outside H. Also, since ρ_H is absolutely irreducible, its endomorphism ring is trivial and the Hom-module must have dimension 1. So the algorithm has very little to do (one trace relation will determine the scalar by which the single basis element must be multiplied to obtain the image of g).
4. In the above simple presentation of the function LINEARTRACEREDUCTION, $\rho_H(h)$ is evaluated for each random $h \in H$. As usual, one can use words in the strong generators of H instead of the original generators of H, but this still means that potentially several products of matrices (which are images of the strong generators) are needed for each evaluation of ρ_H. Thus it is more efficient to use the product replacement algorithm [CLGM+95] in parallel on both the elements of H and their corresponding images in ρ_H. By using the accumulator variant, which needs two products per random element, we can then generate each new random $h \in H$ and the corresponding $\rho_H(h)$ with only two matrix products.
5. Each time the subalgorithm LINEARTRACEREDUCTION computes $\mathrm{Tr}(B \cdot A_i)$, it can use the fast method for computing the trace of a product of two matrices efficiently (see p. 48). This avoids very many matrix multiplications, which yields a huge reduction in time if the number of image matrices is large.
6. In the LINEARTRACEREDUCTION algorithm, as presented, each time a new independent linear relation in the x_i is found, one variable and the corresponding matrix is removed. This means that the subsequent relations only have to be constructed from one less matrix and will be in terms of one less variable, which means the construction of the actual relations speeds up as the algorithm proceeds. However, the reduction step (removing the matrix and corresponding variable) can be expensive. It involves l multiplications of a scalar by a matrix and l matrix additions. Typically, l will equal

k or be close to it, so when k is large, the cost of this reduction is comparable to the cost of computing a new linear relation. We have found that it is best to delay the reductions and wait until $r = \lceil k/2 \rceil$ linear relations have accumulated; then if these relations are echelonized (which takes little cost compared with all the other matrix operations), it is easy to see that the above reduction can be done with r scalar products and matrix additions, which typically halves the time taken for all the reductions.

7. The cost of rewriting the representation so that it is defined on the original generators of G can be non-trivial when the degree is large. For an arbitrary finite group G, we can simply define a representation ρ_1 on the generators $\{h_1, \ldots, h_s\}$ of H and g and then evaluate ρ_1 on the original generators of G (using words in strong generators as usual to make things more efficient). However, this involves computing a BSGS for G which may be very expensive and there is a better method if G is a well-known group with standard generators. Wilson introduced the concept of 'standard generators' for generators of sporadic simple groups [Wil96]; he and others provided black-box algorithms for their construction, given arbitrary generators of the group. E. O'Brien has implemented this within MAGMA as the function `StandardGenerators(G, S)` [O'B06, Sec. 7.6]; the function also works for several classes of classical groups and their covers. We can thus apply this to a definition of G with generators $\{h_1, \ldots, h_s, g\}$ and then evaluate the resulting words defining the standard generators of G at the images $[\rho_H(h_1), \ldots, \rho_H(h_s), A_0 = \rho(g)]$. This is very efficient in general and avoids the construction of a BSGS for G.

8. Under the assumption that the entries of the matrices defining ρ_H are small, then the entries of the matrices defining ρ tend to be rather small too. This can be seen from Minkwitz's formula (Thm. 4.2.1): if we set $D = \frac{|H|}{\chi(1)} \in \mathbb{Z}^{>0}$, then clearly the common denominator introduced into the matrices defining ρ must be a divisor of D. Also, the numerators will increase by at most a factor of the order of $|H| \cdot B$, where B bounds the values of χ, excluding $\chi(1)$ (since $\chi(1)$ cannot occur in the sum for $g \notin H$). So if $|H|$ is moderate, the number of digits in the entries of ρ can never be dramatically more than for those of ρ_H. As will be seen in examples below, the growth in coefficients when moving from ρ_H to ρ is typically small in practice. Usually the denominator introduced is much smaller than D and is sometimes 1. (As an example, if one restricts an integral representation ρ of G to H to obtain absolutely irreducible ρ_H, then the unique extension of ρ_H back to G must equal ρ which is integral.) The very attractive consequence is that we can construct representations of very high degree with small entries via irreducible extension, assuming that the representation ρ_H of H has small entries, and this is often easy to achieve because H is smaller than G.

9. Suppose the given generators of G are $\{g_1, \ldots, g_r\}$. Given any subgroup H_1 for which irreducible extension is applicable for χ, we can first attempt to conjugate H_1 by an element of G to another subgroup H so that one of the g_i is in H. In practice, for several trials (typically up to 1000), we simply choose random $r \in G$ and test whether any g_i^r is in H_1. If so, then we let H be $(H_1)^{r^{-1}}$, and use H instead of the original H_1 for the subgroup. This has the great practical advantage that for the final representation ρ of G affording χ, the image matrix for g_i of G will be $\rho_H(g_i)$, so will be very often sparse or be written over a subfield of F, assuming that the representation ρ_H of H is such. This means that storing the final representation can save a lot of space: since

nearly all groups in the database in Part II have two standard generators, the space taken is often virtually halved (very often, one of the generators is monomial or at least very sparse). As an example, for the degree-126 representation of 3.McL (p. 171), the field F is $\mathbb{Q}(\alpha)$, where the minimal polynomial of α is $x^4 - x^3 - 2x^2 - 3x + 9$, g_1 has order 2 and $\rho(g_1)$ is a monomial matrix with the only non-zero entries being 1 and $\pm\beta$, where $\beta = \frac{1}{6}(-\alpha^3 - 2\alpha^2 + 2\alpha + 3)$ (of order 3 in F). Furthermore, for some of the representations which have been constructed, one of the standard generators of G (say g_1) has order 2 and one can conjugate H so that $g_1 \in H$ and $\rho_H(g_1)$ is diagonal, with only ± 1 on the diagonal; in this case it is very nice to store (and view) the representation in this form! See Ex. 4.5.2 below, for example.

4.5. Examples

Here are a few non-trivial examples which use the irreducible extension algorithm. Several more instances of irreducible extension can be seen in the tables in Part II of the thesis (those entries with 'IE' in the 'Method' field; see Chapter 8 for more information).

Example 4.5.1. Let $G = \mathrm{L}_3(5)$. G has a class of 10 degree-96 conjugate irreducible representations which is missing from the database in [Nic06]. Let χ be one of the corresponding characters, which has entries in $\mathbb{Q}(\zeta_{31})$ and Schur index 1. The minimal-degree character field of χ can be written as $F = \mathbb{Q}(\alpha)$, where α has minimal polynomial

$$x^{10} - 9x^9 + 38x^8 - 116x^7 + 285x^6 - 531x^5 + 747x^4 - 804x^3 + 679x^2 - 390x + 125.$$

We computed a representation $\rho : G \longrightarrow \mathrm{GL}_{96}(F)$ affording χ, as follows (table entry on p. 168). We set H to a maximal subgroup of G of index 31 (there are two such classes but either will do). Now $\chi_H = \chi \downarrow_H$ is absolutely irreducible, so irreducible extension can be used. A representation $\rho_H : H \longrightarrow \mathrm{GL}_{96}(\mathbb{Q})$ was first constructed as the direct induction to H of a degree-4 rational representation of an index-24 subgroup of H (in 0.14s). Then IrreducibleExtension was applied to χ and ρ_H. The largest possible normalized subgroup L had order 400 and for the associated $g \in G \setminus H$ with $L^g = L$ there were 24 initial image matrices (3.8s). Then it took 50 random elements of H to find 24 independent linear relations to obtain the unique image of g (4.4s). Finally, the rewriting of the representation on the standard generators g_1, g_2 of G took 3.4s and yielded $\rho : G \longrightarrow \mathrm{GL}_{96}(F)$. The total time taken was 12.1s.

It was easy to conjugate H at the beginning so that $g_1 \in H$; consequently $\rho(g_1)$ is very sparse (at most two non-zero entries per row, all of which are ± 1), while $\rho(g_2)$ has density 84.8% and its entries have denominator LCM $209375 = 5^5 \cdot 67$ and numerator coefficients of up to 6 digits. Note that the larger entries cannot be avoided if we write the representation over the minimal field F (F has reduced discriminant $5^6 \cdot 31 \cdot 67^2$). But if we rewrite this representation over the cyclotomic field $\mathbb{Q}(\zeta_{31})$ (by simply mapping the entries from F into that field), then the image of g_2 has denominator LCM 25 and the numerator coefficients are all 0, ± 1 or ± 2.

Example 4.5.2. Let $G = \mathrm{U}_5(4)$, of order 53443952640000. A minimal-degree faithful representation of G has degree 204. Let χ be the corresponding character, which has character field $\mathbb{Q}$ and Schur index 2. We computed a representation ρ affording χ as follows (table entry on p. 174). G has a maximal subgroup H of index 66625 with shape

$2^{8+8}.3.\mathrm{L}_2(16)$, such that $\chi_H = \chi \downarrow_H$ is absolutely irreducible, so irreducible extension can be used. A subgroup H_2 of H of index 51 was then found such that it had an irreducible character χ_{H_2} of degree 4 with $\chi_{H_2} \uparrow^H = \chi_H$ (the search for the suitable subgroup took 47s). It then took AbsolutelyIrreducibleRepresentation only 0.6s to construct a representation $\rho_{H_2} : H_2 \longrightarrow \mathrm{GL}_4(F)$ affording χ_{H_2}, where $F = \mathbb{Q}(i)$. This could then be immediately induced to H to obtain $\rho_H : H \longrightarrow \mathrm{GL}_{204}(F)$, affording χ_H. Finally, IrreducibleExtension was applied to χ and ρ_H. The largest normalized subgroup L of H had order 12240, yielding 16 corresponding image matrices and then the desired representation $\rho : G \longrightarrow \mathrm{GL}_{204}(F)$ affording χ was constructed (8.7s). The field F is clearly a minimal field for χ.

Let g_1, g_2 be the standard generators of G. It was easy to conjugate H at the beginning so that $g_1 \in H$; in fact, $\rho(g_1)$ is diagonal with only ±1 on the diagonal, while $\rho(g_2)$ has density 71.4% with denominator LCM 8 and absolute maximum numerator 2, and only 19 distinct entries, such as $\frac{1}{8}(i+2)$. The whole computation took about 57s.

Example 4.5.3. Let $G = \mathrm{Co}_1$, which has order 4157776806543360000. G has an absolutely irreducible rational representation of degree 8855, which we constructed via irreducible extension (table entry on p. 187). Let χ be the corresponding character. By choosing H to be the third largest maximal subgroup of G, equal to $2^{11}{:}\mathrm{M}_{24}$ (index 8292375), we have that $\chi_H = \chi \downarrow_H$ is absolutely irreducible. A representation $\rho_H : H \longrightarrow \mathrm{GL}_{8855}(\mathbb{Q})$ was constructed as the direct induction to H of a degree-5 representation of an index-1771 subgroup of H (18s to find the subgroup of H for induction, and 17s to construct the degree-5 representation by IrreducibleRationalRepresentations). The largest normalized subgroup L of H had order 141557760 and this yielded 10 initial image matrices (9633s; mostly dominated by the modular Meataxe when computing the homomorphisms by the modular algorithm from p. 23). Then 42 random elements of H yielded enough linear relations to determine the unique image of the normalizing element g (327s; the multiplication of images of ρ_H was very fast since the matrices were very sparse). Finally, rewriting the representation on the standard generators of G took 2229s (5 and 12 products respectively for each generator, in terms of the matrices defining the images of the two generators of H and g). The total time taken was about 3.5 hours. The matrices defining the resulting representation have density 41.7% and 34.1% respectively, with entry denominator LCM 16 and all numerators in the range -7 to 7.

CHAPTER 5

General Extension

5.1. Introduction

Let χ be an absolutely irreducible character of a finite group G. The major limitation of the irreducible extension algorithm is that it is very often the case that there is no subgroup H of G such that $\chi \downarrow_H$ is absolutely irreducible, so the algorithm simply cannot be used. The algorithm presented in this chapter removes this limitation completely: it can extend a representation ρ_H affording $\chi \downarrow_H$ to a representation of G which affords χ, where there are no conditions on ρ_H. We call this **general extension** from χ_H to G.

Schulz described an algorithm for general extension, based on a generalization of Minkwitz's formula [Min96] when the multiplicity of each constituent is 1 [Sch02, 2.2]; since this algorithm involves looping over H, it is again obviously limited to the case that H is rather small.

The algorithm presented here involves setting up and solving a system of polynomial equations. The basic situation is as follows. Suppose that χ is an absolutely irreducible character of a finite group G, H is a subgroup of G and $g \in G$ with $G = \langle H, g\rangle$, and we also have a representation $\rho_H : H \to \mathrm{GL}_n(F)$ which affords $\chi_H = \chi \downarrow_H$. We wish to compute a representation $\rho : G \to \mathrm{GL}_n(F)$ affording χ, with $\rho \downarrow_H = \rho_H$ (and we assume that χ can be realized over F). Just as in the previous irreducible extension algorithm, suppose that we know matrices $[A_0, A_1, \ldots, A_k]$ such that the matrix $\rho(g)$ must equal $X = A_0 + \sum_{i=1}^{k} x_i \cdot A_i$ for some assignment of the x_i to elements of F. We can construct relations in G involving g and generators $\{h_1, \ldots, h_r\}$ of H and evaluate these at the matrices X and $\{\rho_H(h_1), \ldots, \rho_H(h_r)\}$ respectively, yielding polynomial relations on the x_i which give necessary conditions for the possible solutions. For example, if $g^2 = h$ for some $h \in H$, then we can form the corresponding matrix equation $X^2 - \rho_H(h) = 0$, yielding one polynomial equation for each entry of the matrix on the LHS of the equation. Some of the practical difficulties with this approach are:

1. As the degree n of the representation grows, the required operations on $n \times n$ matrices with polynomial entries becomes very expensive.
2. There may be a large number of variables $x_1, \ldots, x_k$ and the maximal degree d of a relation in the x_i variables (which will equal the degree of g in the corresponding group relation involving g) may grow large. There are $\binom{k+d-1}{d}$ monomials of degree d in k variables, and this number grows very quickly as d increases.
3. After collecting several polynomial relations on the x_i, we need to know whether there are enough relations so that a solution to the polynomial system yields a valid image matrix for $\rho(g)$.

4. Solving the polynomial system itself can be very difficult when there are several variables.

Previous presentations of this kind of algorithm have been restricted to limited situations, particularly for characteristic zero. Wilson sketched some basic techniques and gave some simple manual examples in [Wil99]. Plesken & Souvignier [PS97] mentioned a similar method which was suitable only for representations of small degree; they gave a few basic improvements but they were mainly interesting in proving finitely-presented groups infinite, so did not pursue the method in detail.

Despite the above challenges, we will describe a heuristic algorithm which is very effective for constructing representations of very large degree. Since the algorithm involves solving non-linear polynomial equations, we need some non-trivial concepts from Algebraic Geometry and Commutative Algebra, and we use Gröbner bases in practice. The key feature which we develop is an effective termination criterion so that one can generate a relatively small number of low-degree polynomial relations efficiently and know when there are enough relations to determine a correct result.

5.2. Theory

Let F be a field and $F[x_1, \ldots, x_n]$ be the ring of multivariate polynomials over F. We first note some basic concepts from Algebraic Geometry and Commutative Algebra which will be needed. To save space, we refer the reader to standard texts such as [CLO96, BW93], and assume that the following objects and associated facts are familiar:

1. An (affine) **variety** V, the variety $\mathbf{V}_F(I)$ of an **ideal** I of $F[x_1, \ldots, x_n]$ and the ideal $\mathbf{I}_F(V)$ of a variety V, and the fact that $I \subseteq \mathbf{I}_F(\mathbf{V}_F(I))$ for an ideal I, but equality need not occur. [CLO96, Ch. 1, §4, §5]
2. A **Gröbner basis** of an ideal I of $F[x_1, \ldots, x_n]$, w.r.t. the *grevlex* (graded-reverse-lexicographical) or *lex* (lexicographical) monomial order for R. [CLO96, Ch. 2]
3. The **Zariski closure** of a subset of affine space, **irreducible** varieties, and **prime** ideals and the fact that a variety V is irreducible over F if and only if $\mathbf{I}_F(V)$ is a prime ideal. [CLO96, Ch. 4, §4, §5]
4. A **rational map** between two irreducible affine varieties and a **birational map** from one variety to another (a rational map with a rational inverse map; this has to be understood in the extended sense that the composition, in either order, need only be defined on a non-empty Zariski open subset). [CLO96, Ch. 5, §5]
5. **Projective space** $\mathbb{P}^k$ and projective varieties. [CLO96, Ch. 8]
6. The **dimension** of a variety V (equivalent to the transcendence degree of the function field of V) and the fact that the dimension of V equals the dimension of the ideal $I = \mathbf{I}_F(V)$ (which also equals the degree of the Hilbert polynomial of I, or the Krull dimension of the affine ring $F[x_1, \ldots, x_n]/I$). [CLO96, Ch. 9, §5]
7. **Isomorphic** varieties and the fact that they have the same dimension [CLO96, Ch. 9, §5]
8. A **maximally independent set** modulo an ideal I of $F[x_1, \ldots, x_n]$ (a subset S of $\{x_1, \ldots, x_n\}$ such that $I \cap \langle S \rangle = \emptyset$ and the cardinality of S is maximal) and the fact that the dimension of I equals the cardinality of a maximally independent set

modulo I. (Intuitively, such a S is a set of 'free variables' for the system of polynomial equations corresponding to I.) [BW93, 9.3], [CLO96, Ch. 9, §5, Cor. 4]

Theorem 5.2.1. *(***The Projective Extension Theorem***)* [CLO96, Ch. 8, §5, Def. 4, Thm. 6] *Let $I = \langle f_1, \ldots, f_l \rangle$ be an ideal of $F[t_1, \ldots, t_D, x_1, \ldots, x_m]$, where F is an algebraically closed field and the f_i are $(t_1, \ldots, t_D)$-homogeneous polynomials (homogeneous in the t_i variables). Set*

$$V := \mathbf{V}_F(I) \subset {\mathbb{P}_F}^{D-1} \times {\mathbb{A}_F}^m$$

and set

$$\tilde{I} := \{f \in F[x_1, \ldots, x_m] : \textit{for } 1 \leq i \leq D, \exists e_i \geq 0 \textit{ with } t_i^{e_i} f \in I\},$$

called the projective elimination ideal *of I. If*

$$\pi : {\mathbb{P}_F}^{D-1} \times {\mathbb{A}_F}^m \longrightarrow {\mathbb{A}_F}^m$$

is the projection onto the last m coordinates, then

$$\pi(V) = \mathbf{V}_F(\tilde{I}).$$

(The point of the theorem is that we have equality in the last statement, so $\pi(V)$ is itself an algebraic variety, and not just that $\pi(V) \subseteq \mathbf{V}_F(\tilde{I})$ as sets.)

We can now present our main theorem which characterizes the set of possible image matrices in an extension from a representation of a subgroup H to G.

Theorem 5.2.2. *Suppose that G is a finite group, $H < G$ and $g \in G$ with $G = \langle H, g \rangle$, $\chi \in \mathrm{Irr}(G)$ and F is a field such that χ can be realized over F and $\rho_H : H \longrightarrow \mathrm{GL}_n(F)$ affords $\chi_H = \chi \downarrow_H$. Let V be the set of all possible $A \in \mathcal{M}_n(F)$ such that $\rho(g) = A$ for any extension $\rho : G \longrightarrow \mathrm{GL}_n(F)$ of ρ_H to G which affords χ. Then V can be characterized as follows:*

1. *Let ρ_1 be any fixed F-representation which affords χ, with $\rho_1 \downarrow_H = \rho_H$ and set $A_1 := \rho_1(g)$. Then*
$$V = \{TA_1T^{-1} : T \in C_{\mathrm{GL}_n(F)}(\rho_H(H))\}.$$
2. *Let $D = \mathrm{Dim}_F(\mathrm{End}_{FH}(\rho_H))$ (which equals the norm of χ_H w.r.t. H, by Prop. 4.3.1). Then V is an irreducible affine variety over F of dimension $D - 1$.*

Proof. For the first point, first note that such a ρ_1 exists, since if ρ_0 is any representation over F which affords χ, then $\rho_0 \downarrow_H$ is equivalent to ρ_H so one may conjugate ρ_0 to some ρ_1 so that $\rho_1 \downarrow_H = \rho_H$. For the chosen fixed ρ_1, $A_1 = \rho_1(g)$ is a fixed constant matrix which is in V. Write $C = C_{\mathrm{GL}_n(F)}(\rho_H(H))$ (the centralizer of the matrix group image of ρ_H). If ρ is any other F-representation of G which affords χ, with $\rho \downarrow_H = \rho_H$, then clearly $\rho = (\rho_1)^T$, where $T \in C$, so $\rho(g) = TA_1T^{-1} \in V$. Conversely, for any $T \in C$, defining $\rho(g)$ to be TA_1T^{-1} clearly gives an extension of ρ_H. This proves the first point.

The second point is much more difficult to prove. By the first point, V can be defined by a rational parametrization (involving **rational functions**), but we need to prove that it is identical to an affine variety, which is the set of solutions to a set of **polynomial** equations. The non-trivial thing to prove is that V itself is an affine variety; the irreducibility and dimension conditions then follow fairly easily.

Keep the same fixed A_1 from above. Let $E = \mathrm{End}_{FH}(\rho_H)$ and $D = \mathrm{Dim}_F(E)$ and let $[e_1, \ldots, e_D]$ be an F-basis for E. Then the centralizer $C = C_{\mathrm{GL}_n(F)}(\rho_H(H))$ of ρ_H equals the unit group of E and a general element of C can be written as

$$T(s_1, \ldots, s_D) = \sum_{i=1}^{D} s_i e_i, \quad s_i \in F,$$

where $T(s_1, \ldots, s_D)$ is invertible. Now since conjugation by a non-zero scalar matrix has no effect, conjugation by $T(s_1, \ldots, s_D)$ can be considered to be a projective operation, so the tuple of s_i values can be viewed as lying in the projective space ${\mathbb{P}_F}^{D-1}$ and there is also a corresponding symbolic matrix $T(t_1, \ldots, t_n)$ which is homogeneous in the t_i indeterminates. Similarly, we can let $X(x_{1,1}, \ldots, x_{n,n})$ be the $n \times n$ matrix with the (i,j)-th entry equal to $x_{i,j}$, where the $x_{i,j}$ for $1 \le i, j \le n$ are n^2 extra indeterminates. Since we desire $X(x_{1,1}, \ldots, x_{n,n})$ to correspond to a generic element of V, consider the system of n^2 polynomial equations given by the matrix equation:

$$X(x_{1,1}, \ldots, x_{n,n}) \cdot T(t_1, \ldots, t_D) = T(t_1, \ldots, t_D) \cdot A_1, \qquad \text{[E1]}$$

where each polynomial is in the multivariate polynomial ring $F[t_1, \ldots t_D, x_{1,1}, \ldots, x_{n,n}]$ (recall that A_1 is a constant matrix). Each solution to this system of equations is a tuple of the form:

$$(s_1, \ldots, s_D, c_{1,1}, \ldots, c_{n,n}) \in {\mathbb{P}_F}^{D-1} \times {\mathbb{A}_F}^{n \times n}.$$

The potential problem is that there could conceivably be a solution to the polynomial system [E1] in which the T matrix would be not invertible, and such a solution would not correspond to an element of V. But this situation does not arise for the following reason. Suppose that $(s_1, \ldots, s_D, c_{1,1}, \ldots, c_{n,n})$ is a solution of [E1]. Let $C = X(c_{1,1}, \ldots, c_{n,n})$ and let $S = T(s_1, \ldots, s_D)$. Since the s_i coordinates are in the projective space ${\mathbb{P}_F}^{D-1}$, S is non-zero. Now if S were not invertible, then its rows would generate a non-zero proper subspace of F^n which is invariant under right multiplication by both A_1 (since $CS = SA_1$ by [E1]) and by $\rho_H(h)$ for all $h \in H$ (since S is an endomorphism of ρ_H by construction) and thus also by $\rho_1(x)$, for all $x \in G$. But this contradicts the irreducibility of χ, which ρ_1 affords. We thus have that if $(s_1, \ldots, s_D, c_{1,1}, \ldots, c_{n,n})$ is a solution of [E1], then $T(s_1, \ldots, s_D)$ is invertible and hence

$$X(c_{1,1}, \ldots, c_{n,n}) = T(s_1, \ldots, s_D) \cdot A_1 \cdot T(s_1, \ldots, s_D)^{-1}. \quad \text{[E2]}$$

Thus the point set of the variety $V_{[\mathrm{E1}]} \subset {\mathbb{P}_F}^{D-1} \times {\mathbb{A}_F}^{n \times n}$ of the ideal $I_{[\mathrm{E1}]}$ generated by the polynomials given by [E1] equals the set of solutions of [E2]: for each non-zero tuple $(s_1, \ldots, s_D) \in (F^*)^D$, there exists an element in $V_{[\mathrm{E1}]}$ and vice versa.

We next show that if we remove the s_i coordinates from the elements of $V_{[\mathrm{E1}]}$, then we still have an algebraic variety. Define the projection

$$\pi : {\mathbb{P}_F}^{D-1} \times {\mathbb{A}_F}^{n \times n} \longrightarrow {\mathbb{A}_F}^{n \times n}$$

by

$$(s_1, \ldots, s_D, c_{i,j}) \mapsto (c_{i,j}).$$

and let V_π denote the image of $V_{[\mathrm{E1}]}$ under π. We claim that V_π is a variety over F and equals $\mathbf{V}_F(I_\pi)$. First let $\bar{F}$ be an algebraic closure of F. Then π naturally extends to a map $\bar{\pi} : {\mathbb{P}_{\bar{F}}}^{D-1} \times {\mathbb{A}_{\bar{F}}}^{n \times n} \longrightarrow {\mathbb{A}_{\bar{F}}}^{n \times n}$. Let $\bar{I}_{[\mathrm{E1}]}$ be the ideal of $\bar{F}[t_1, \ldots t_D, x_{i,j}]$ generated by the polynomials given by [E1], let $\bar{V}_{[\mathrm{E1}]}$ be the variety over $\bar{F}$ of $\bar{I}_{[\mathrm{E1}]}$, and let $\bar{V}_\pi$ denote the image of $\bar{V}_{[\mathrm{E1}]}$ under $\bar{\pi}$. Then by the Projective Extension Theorem (Thm. 5.2.1), $\bar{V}_\pi$ equals the variety over $\bar{F}$ of the projective elimination ideal of $\bar{I}_{[\mathrm{E1}]}$, so $\bar{V}_\pi$ itself is a variety (i.e., no new points arise in the Zariski closure of $\bar{V}_\pi$). It is not difficult to move the result back to F. Write $I_\pi = I(V_\pi)$ over F. Suppose there is a point p in $\mathbf{V}_F(I_\pi)$ but not in V_π. By the previous paragraph, a point in $\mathbf{V}_F(I_\pi)$ must also be $\bar{V}_\pi$, so there must be a preimage $(s_1, \dots s_D, c_{i,j})$ of p under $\bar{\pi}$ with $s_1, \dots, s_D \in \bar{F}$ and some $s_k \notin F$, but with all $c_{i,j} \in F$. So the corresponding $C = C(c_{i,j})$ has entries in F and is similar to A_1 over $\bar{F}$. Moving to the rational forms of C and A_1 over F, there must be invertible $U \in \mathcal{M}_n(F)$ with $C = UAU^{-1}$. But U must then be in the centralizer of ρ_H over F, with corresponding $s_1, \dots, s_D$ values all in F. Contradiction. Thus V_π is a variety over F and equals $\mathbf{V}_F(I_\pi)$.

We now show that V_π can be defined by a rational parametrization. Define the partial map

$$f_0 : {\mathbb{P}_F}^{D-1} \dashrightarrow {\mathbb{P}_F}^{D-1} \times {\mathbb{A}_F}^{n\times n}$$

by

$$(s_1, \dots, s_D) \mapsto (s_1, \dots, s_D, c_{i,j})$$

where $c_{i,j} = (T(s_D) \cdot A_1 \cdot T(s_D)^{-1})_{[i,j]}$. Then f_0 is defined on a Zariski open subset and is trivially injective. Thus $V_{[\mathrm{E1}]}$ is the partial image of f_0 and combining the 2 injective maps f_0 and π gives an injective map $f = f_0 \circ \pi$ from ${\mathbb{P}_F}^{D-1}$ to V_π. We can also take the affine part ${\mathbb{A}_F}^{D-1}$ of ${\mathbb{P}_F}^{D-1}$ with first coordinate equal to 1. The natural embedding ι of ${\mathbb{A}_F}^{D-1}$ in ${\mathbb{P}_F}^{D-1}$ is birational. Combining this with f yields an injective map $g = \iota \circ f$ from ${\mathbb{A}_F}^{D-1}$ onto the variety V_π. This map g thus presents the variety V_π as a rational parametrization over the infinite field F, so by [CLO96, Ch. 4, §5, Prop. 6], V_π is irreducible.

Finally, we see that the map $g = \iota \circ f = \iota \circ f_0 \circ \pi$ is birational. First, ι and f_0 are easily seen to be birational. If we restrict π to $V_{[\mathrm{E1}]}$, then π becomes injective, since distinct elements in the domain having the same image under π would yield matrices T_1 and T_2 with $T_1T_2^{-1}$ non-scalar and centralizing both ρ_H and A_1, again contradicting the irreducibility of χ. Thus there is a unique inverse under π for any element of V_π and so π restricted to $V_{[\mathrm{E1}]}$ is birational. Thus g is birational and since two irreducible varieties which are birationally equivalent have the same dimension [CLO96, Ch. 9, §5, Cor. 7] we have that ${\mathbb{A}_F}^{D-1}$ and V_π have the same dimension, which is $D-1$, thus proving the second point of the theorem. □

Corollary 5.2.3. *Let $G, H, g, \chi, F, \rho_H, V, D$ be as in the previous theorem. Suppose that I is a prime ideal of $F[x_{1,1}, \dots, x_{n,n}]$ of dimension $D-1$ with $\mathbf{V}_F(I) \supseteq V$. Then $\mathbf{V}_F(I) = V$ and so if A is any matrix in $\mathbf{V}_F(I)$, then defining $\rho : G \to \mathrm{GL}_n(F)$ by $\rho(h) = \rho_H(h)$ for $h \in H$ and $\rho(g) = A$ yields a valid representation ρ of G affording χ with $\rho \downarrow_H = \rho_H$.*

Proof. Since $\mathbf{V}_F(I)$ and V are both irreducible algebraic varieties over F of equal dimension and $I \subseteq \mathbf{I}_F(V)$ (since $\mathbf{V}_F(I) \supseteq V$) then by [BW93, 7.57][1] we must have that the ideals are equal and thus the corresponding varieties over F are also equal. The second statement follows by the actual definition of V in the Theorem. □

[1] There is a misprint in the statement of that Lemma: the first 'dim(J)' should be 'dim(I)'.

5.3. The Heuristic Algorithm

We can now present our heuristic algorithm for general extension. This is broken into three parts, as follows:

1. The first subalgorithm ElementOfVariety attempts to find an element of the variety of a given ideal I over a characteristic zero field F. The basic idea is to set some variables to constants until there is a unique solution over F. Since the ideal will be positive-dimensional in general, finding a solution point with entries in F (i.e., without extending the field) is a hard problem in Arithmetic Geometry in general, but this simple method works effectively for the applications we encounter.
2. The heart of the general extension algorithm generates polynomial relations in the variables occurring in the symbolic matrix X which represents the image of a fixed $g \in G$, where $G = \langle H, g \rangle$. We will call a relation in G of the form $(gh)^e = 1$, for some $h \in H$ and $e > 1$ a **group order relation**, since it involves finding elements of small order defined by products of g and elements of H. The main algorithm successively generates such group order relations for increasing e (starting with 2) and collects the corresponding polynomial relations. In this way, the degree of g in each word stays as low as possible early on, so the degrees of the corresponding polynomials start low also and simplifications of the polynomial system as the algorithm proceeds may make higher-degree relations feasible later (this phenomenon is discussed in detail below). The second subalgorithm ExtendRelations finds a group order relation for the given order e if possible and extends the polynomial relations accordingly. A primitive version is first presented here; a much more efficient version will be given in Subsec. 5.4.6 below.
3. Finally, the main algorithm GeneralExtension uses the above subalgorithms in a simple way. The algorithm first computes initial image matrices via a normalized subgroup and uses linear reduction with the character to reduce the number of image matrices, just as in the irreducible extension algorithm. The only difference is that the linear reduction stops when no more reduction is possible (since the linear reduction will not reduce to a unique solution if χ_H is not absolutely irreducible). Then the algorithm calls ExtendRelations to generate polynomial relations on the symbolic matrix defined by the remaining image matrices until there are enough relations and then it calls ElementOfVariety to find a solution of the polynomial system which yields a valid image for g, from which the representation affording χ can be constructed.

Subalgorithm ElementOfVariety(I)
Input:

- An ideal I of $F[x_1, \ldots x_k]$, where F is $\mathbb{Q}$ or a number field.

Output:

- An element $(a_1, \ldots, a_k) \in F^k$ of $\mathbf{V}_F(I)$ or `'Fail'` if none is found.

STEPS:

1. Let d be the dimension of I and let $S = \{x_{i_1}, \dots x_{i_d}\}$ be a maximally independent set modulo I (using, for example, the algorithm in [BW93, Table 9.6]).
2. Choose non-zero constants $c_1, \dots, c_d \in F$ so that the ideal $J := \langle I, x_{i_1} - c_1, \dots, x_{i_d} - c_d \rangle$ has dimension 0. If a Gröbner basis of J (with any monomial order) consists of linear polynomials only, then return the unique element of $\mathbf{V}_F(J)$.
3. Compute the lexicographical Gröbner basis G of I. Select an $f(x_i, x_j)$ in G such that $f(x_i, x_j)$ involves variables x_i, x_j only and has total degree 2, and $x_i \in S, x_j \notin S$. If no such $f(x_i, x_j)$ exists, then return `'Fail'`. Otherwise, determine whether the conic C defined by $f(x_i, x_j) = 0$ has a rational point $(c_1, c_2) \in F^2$. If there is no such point then return `'Fail'`. Otherwise set
$$J := \langle I, x_i - c_1, x_j - c_2 \rangle,$$
and return ELEMENTOFVARIETY(J).

Subalgorithm EXTENDRELATIONS$(g, \rho_H, [A_0, A_1, \dots, A_k], B, e)$
INPUT:

- An element $g \in G$ for a finite group G and a representation $\rho_H : H \to \mathrm{GL}_n(F)$ of H, a subgroup of G.
- Matrices $[A_0, A_1, \dots, A_k] \in \mathcal{M}_n(F)$ and a set $B \subset F[x_1, \dots, x_k]$ of relation polynomials such that for any extension ρ of ρ_H to G, $\rho(g)$ must equal $A_0 + \sum_{i=1}^{k} c_i \cdot A_i$ for some $(c_1, \dots, c_k) \in \mathbf{V}_F(I)$, where $I = \langle B \rangle$.
- An integer $e > 1$.

OUTPUT:

- A new set of relation polynomials B' such that $I' = \langle B' \rangle \supseteq I$ and for any extension ρ of ρ_H to G, $\rho(g)$ must equal $A_0 + \sum_{i=1}^{k} c_i \cdot A_i$ for some $(c_1, \dots, c_k) \in \mathbf{V}_F(I')$.

STEPS:

1. Set T to some default value (typically 1000). For T tries, choose a random element $h \in H$ until $t = (h \cdot g)^e \in H$. If no such h is found, return B.
2. Set $X := A_0 + \sum_{i=1}^{k} x_i \cdot A_i \in \mathcal{M}_n(F)[x_1, \dots, x_k]$.
3. Set $A := (\rho_H(h) \cdot X)^e - \rho_H(t)$ and set S to the set of all entries of A.
4. Set B' to the interreduction of $(B \cup S)$ and return B'. *[The interreduction of a set of polynomials is computed by repeatedly reducing each polynomial to normal form w.r.t. the other polynomials until no more reductions are possible.]*

Algorithm GENERALEXTENSION(χ, ρ_H)
INPUT:

- An absolutely irreducible character χ for a finite group G.

- A representation $\rho_H : H \rightarrow \mathrm{GL}_n(F)$ affording $\chi\downarrow_H$, where H is a maximal subgroup of G and F is a field with $F(\chi) = F$.

OUTPUT:

- A representation $\rho : G \rightarrow \mathrm{GL}_n(F)$ affording χ, such that $\rho\downarrow_H = \rho_H$. Or possibly `'Fail'` is returned, if not enough relations found.

STEPS:

1. Set `MaxLinearTries` to some default value (typically 100).
   Set `MaxOrder` to some default value (typically 100).
   Set `StableCount` to some default value (typically 3).
2. Set $g, [A_1, \ldots, A_l] :=$ EXTENSIONIMAGESETUP(G, ρ_H). [See p. 83.]
   Set $[A_0, A_1, \ldots, A_k] :=$ LINEARTRACEREDUCTION$(\chi, \rho_H, g, [A_1, \ldots, A_l],$ `MaxLinearTries`$)$.
3. Set $D := \langle \chi_H, \chi_H \rangle_H$, where $\chi_H = \chi\downarrow_H$. In the following, let `Finished`(I) for an ideal I denote the condition that I is prime and the dimension of I equals $D - 1$.

   Set $B := \{\}$, $c := 0$ and e to the smallest divisor of $|G|$ with $e > 1$.

   ```
   Loop forever:
   {
       Set $B_{\mathrm{new}} :=$ EXTENDRELATIONS$(g, \rho_H, [A_0, A_1, \ldots, A_k], B, e)$.
       If $B_{\mathrm{new}} \neq B$ then set $B := B_{\mathrm{new}}$, set $c := 0$ and go to the top
           of the loop (use the same $e$ while something new).
       Set $c := c + 1$ and if $c <$ StableCount then go to the top of the loop.
       Set $I := \langle B \rangle$. If Finished$(I)$ then break out of the loop.
       Set $c := 0$ and set $e$ to the smallest integer greater than $e$ which divides $|G|$.
       If $e >$ MaxOrder then break out of the loop.
   }
   ```

4. If not `Finished`(I) then compute a presentation of G on the generators $\{g, h_1, \ldots, h_r\}$ (where $h_1, \ldots, h_r$ are generators of H), and successively evaluate each relation on $(X, \rho_H(h_1), \ldots, \rho_H(h_r))$ (where $X = A_0 + \sum_{i=1}^{k} x_i \cdot A_i \in \mathcal{M}_n(F)[x_1, \ldots, x_k]$) and include the corresponding relation polynomial in the ideal I (one can stop if `Finished`(I) becomes true at any point).
5. If not `Finished`(I) then for each conjugacy class representation c of G which is not in H, compute a word w such that $c = w(g, h_1, \ldots, h_r)$ and include the relation polynomial $w(X, \rho_H(h_1), \ldots, \rho_H(h_r)) - \chi(c)$ (where X is as above) in the ideal I (again, one can stop if `Finished`(I) becomes true at any point).
6. Set $(c_1, \ldots, c_k) :=$ ELEMENTOFVARIETY(I). If `'Fail'` is returned, then return `'Fail'`.
7. Set $A := A_0 + \sum_{i=1}^{k} c_i A_i$, define $\rho : G \rightarrow \mathrm{GL}_n(F)$ by $\rho(h) = \rho_H(h)$ for $h \in H$ and $\rho(g) = A$ and return ρ.

Theorem 5.3.1. *Algorithm* GENERALEXTENSION *is correct (i.e., if it does not return* '`Fail`'*, then the returned ρ is valid extension of ρ_H to G affording χ and is written over F).*

Proof. Let V be the set of all possible images of g over F under an extension of ρ_H to G affording χ. By Thm. 5.2.2, V is an irreducible variety over F of dimension $D-1$. We need only show that the algorithm terminates and that if '`Fail`' is not returned in Step 8, then the matrix A assigned in Step 9 must lie in V.

Step 2 does the same setup of the image matrices as IRREDUCIBLEEXTENSION, except that LINEARTRACEREDUCTION will return without reducing to a unique image matrix A_0 if ρ_H is not absolutely irreducible. Now for the matrices $[A_0, A_1, \ldots, A_k]$ assigned at the end of Step 3, define $\phi : F^k \to F^{n \times n}$ by

$$(c_1, \ldots, c_k) \mapsto A_0 + \sum_{i=1}^{k} c_i A_i.$$

Then ϕ is a morphism (polynomial map) from the variety F^k to the variety $F^{n\times n}$, and since $[A_1, \ldots, A_k]$ are linearly independent over F, ϕ is an embedding. Then by Lem. 4.3.2, $V \subset F^{n\times n}$ is a subvariety of $\phi(F^k)$ and each call to Subalgorithm EXTENDRELATIONS in Step 3 clearly adds only relation polynomials from $F[x_1, \ldots, x_k]$ to B which match group relations in G so that it always holds that any $f \in B$ vanishes on $\phi^{-1}(v)$ for all $v \in V$, so we always have that $\phi(\mathbf{V}_F(\langle B\rangle)) \supseteq V$.

The loop in Step 3 clearly terminates. For each possible order e, if there are `StableCount` calls of EXTENDRELATIONS for e with no change to B, then e is increased. Now B cannot change indefinitely, since that would imply an infinite sequence of strictly increasing ideals, which contradicts the ascending chain condition on ideals of multivariate polynomial rings over a field [CLO96, Ch. 2, §5, Thm. 7]. So either the ideal I generated by B eventually satisfies the primality and dimension condition and the loop is exited, or there is termination of the loop when e exceeds the bound `MaxOrder`.

Assume first that the '`Finished`' condition on the ideal $I = \langle B\rangle$ of $F[x_1, \ldots, x_k]$ is satisfied at the end of Step 3 (so Steps 4 and 5 are skipped). Then at Step 6, I is prime and has dimension $D-1$ so if we let $W = \mathbf{V}_F(I)$, then $\phi(W)$ is a subvariety of $\phi(F^k)$ which contains V. As W is irreducible and has dimension $D-1$, we have that $\phi(W) = V$ by Cor. 5.2.3.

If either of the bodies of Steps 4 and 5 is entered, then again relation polynomials are inserted into I which give necessary conditions for a solution, based on the presentation of G or the character values. All the relations inserted in Step 4 force an element of $\phi(W)$ to be a valid image of g under some extension to G of ρ_H and Step 5 forces such a representation to afford the character χ also. So trivially $\phi(W) = V$ in this situation also.

Subalgorithm ELEMENTOFVARIETY clearly finds an element of W if it does not fail: in Step 2 of that subalgorithm, the extension ideal of I w.r.t. S (obtained by moving the variables of S into a rational function field) must be zero-dimensional [BW93, 1.122, 7.47] and since F is infinite, it is elementary to find constants $c_1, \ldots, c_d$ such that the corresponding denominators do not vanish and so that the ideal J is zero-dimensional; in such a case, the variety of J over F is finite and has cardinality one if and only if all the Gröbner basis elements are linear. If the conic method is used, then forcing the relevant coordinates to match the solution to the conic clearly reduces the dimension of the ideal by 2, so the recursive call will terminate.

Thus the matrix A constructed in Step 7 must lie in $\phi(W) = V$. This proves that the returned ρ is a valid representation of G which affords χ. □

Remarks 5.3.2. The whole of the next section will be devoted to a detailed description of several major improvements to the basic algorithm which make it much more practical. But we first give some simple remarks on the algorithm and a small example to illustrate its basic working.

1. The input representation ρ_H may be any representation of H affording $\chi \downarrow_H$, but in practice one should of course pass in a block diagonal form of ρ_H with irreducible blocks so that ExtensionImageSetup and LinearTraceReduction can exploit the block structure. It is often the case that the latter two subalgorithms dominate the time (even for very large examples), so it is worth improving things here as much as possible. The irreducible components over F can be computed by the algorithm IrreducibleRepresentationsOverField.
2. The parameter `MaxLinearTries` determines when LinearTraceReduction should give up trying further random elements of H; 100 seems a reasonable default but it can be varied, depending on the expense of a single try. In IrreducibleExtension, the linear reduction is guaranteed to reduce to a unique solution (with no variables left), but this will not happen here if ρ_H is not absolutely irreducible. The initial linear reduction is usually very much worth doing, since it reduces the number of variables, and this can make a critical difference when constructing the polynomial relations later, as will be seen below.
3. Rewriting the final representation on the original generators of G is done in exactly the same way as for IrreducibleExtension (see p. 88). As pointed out before, this can be non-trivial but for the large representations which we computed, we were able to use the method involving words in the standard generators which is very efficient (examples of this will be seen for the large sporadic group representations below).
4. Steps 4 and 5 are included to guarantee that there are enough polynomial relations to give a correct solution, but these steps are practically never needed in our implementation. We have found that for every time we have used the algorithm, it is easy to find enough group order relations of very small order to generate the same ideal as that given by a full presentation in Step 4. Also, Step 5 ensures that all character values of χ are covered by polynomial relations, but what happens practically always is that either ρ_H can extend only to a unique representation affording χ (and not a distinct conjugate of χ) or the initial call to LinearTraceReduction hits enough elements of the form gh with $h \in H$ such that the corresponding character values produce enough conditions so that any solution to the polynomial system can only give a representation which affords χ itself.
5. Suppose that the irreducible constituents of ρ_H over F are all absolutely irreducible and occur with multiplicity 1 and ρ_H has the corresponding block diagonal form (this situation happens very often in practice). Then clearly every element of the centralizer of ρ_H is a block-diagonal sum of non-zero scalar matrices. It is then easy to see that for any image matrix $A \in \mathcal{M}_n(F)$ of the corresponding variety V over F from Thm. 5.2.2,

the (i,j)-th entry of A has the form $\frac{t_{l_i}}{t_{l_j}} c_{i,j}$ for some constants $c_{i,j} \in F$, $1 \le l_i, l_j \le D$, and any $(t_1, \ldots, t_D) \in (F^*)^D$, and the corresponding ideal $I = \mathbf{I}_F(V)$ is generated by linear polynomials and polynomials of the form $x_A \cdot x_B - d_{A,B}$ for non-zero constants $d_{A,B} \in F^*$ (so these polynomials have recursive degree 1 in each variable, even though they are not necessarily linear). Thus in the subalgorithm ElementOfVariety, the Gröbner basis will consist of such polynomials only and it is always trivial to find an element of the variety with values lying in F (practically any non-zero evaluation choice for the maximally independent variables will give a maximal ideal). The whole algorithm GeneralExtension thus **always** succeeds in this case, and so returns a representation written over a minimal field F, if F is such for χ. A very simple example of this occurs in Ex. 5.3.3 below, but in nearly all the larger examples we present, ρ_H satisfies the above condition too, so the corresponding relation ideal has the simple form too (with recursive degree 1 in all variables). An example of this situation with several variables occurs when constructing the degree-3588 representation of Fi_{23} (see p. 118).

When this simple situation does not occur, it is still common that for each irreducible component σ of ρ_H, either of these conditions hold:

- The dimension of the endomorphism ring of σ is 2 and the multiplicity of σ is 1;
- σ is absolutely irreducible and occurs with multiplicity 2.

In such a case, we have found that the conic method always succeeds for all the examples which we have encountered. An example where the conic method is needed is given in Ex. 5.5.3 below.

6. It is in fact easy to extend the subalgorithm ElementOfVariety so that it always succeeds and returns an element of the variety, but potentially with coordinates in some proper extension field of F. If Step 3 of that subalgorithm fails, then the subalgorithm can simply put the zero-dimensional ideal J from Step 2 into normal position [BW93, 8.81] and let E be the appropriate extension field and so after lifting to E, the variety of J will be non-empty and an element of this with values in E can be returned (and this would yield a valid representation affording χ, written over E). But since we focus on computing representations over minimal fields in this thesis, the algorithm as stated avoids extending the input field F.

7. Just as in the irreducible extension algorithm, we first conjugate H if possible so that one of the given generators of G is in H, so that the corresponding image matrix is usually sparse or has entries in a subfield, etc., so the final representation is more compact (see p. 88). Examples of this will be seen below.

Example 5.3.3. Let $G = \langle g_1, g_2 \rangle \cong \mathrm{A}_6$, where $g_1 = (1,2)(3,4,5,6), g_2 = (1,2,3)$ and let χ be one of the irreducible rational characters of G of degree 5. Let $H = \langle h_1, h_2 \rangle$, where $h_1 = (1,3,5)(2,4,6), h_2 = (1,6,4,3)(2,5)$; H is a subgroup of G of order 24 (shape $2.2^2.3$) and $\chi_H = \chi \downarrow_H$ splits as $1+1+3$. Corresponding irreducible representations $\sigma_1, \sigma_2, \sigma_3$ of H are easily constructed and are defined by:

$$\sigma_1(h_1) = \begin{pmatrix} 1 \end{pmatrix}, \quad \sigma_1(h_2) = \begin{pmatrix} 1 \end{pmatrix}$$

$$\sigma_2(h_1) = \begin{pmatrix} 1 \end{pmatrix}, \quad \sigma_2(h_2) = \begin{pmatrix} -1 \end{pmatrix}$$

$$\sigma_3(h_1) = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \sigma_3(h_2) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$$

Let $\rho_H = \sigma_1 \oplus \sigma_2 \oplus \sigma_3$ (the block diagonal sum). We can extend ρ_H to a representation ρ affording χ using GENERALEXTENSION, as follows.

- In Step 2, a subgroup $L = \langle (2,3)(5,6), (2,5)(3,6) \rangle$ of H and $g = (1,4)(2,6,3,5) \in G$ are immediately found with $g \in G \setminus H$ and $L^g = L$; L has order 4 and g has order 4, with $g^2 = h_s \in H$, and there are 7 initial image matrices $[A_1, \ldots, A_7]$. Then 3 linear relations are found in LINEARTRACEREDUCTION, so there are 4 new image matrices $[A_1, A_2, A_3, A_4]$ with a constant matrix A_0 such that the image $\rho(g)$ must equal $X = A_0 + \sum_{i=1}^{4} x_i A_i$ for some assignment of the x_i variables. Writing this out, we get:

$$X = \begin{pmatrix} -\frac{1}{2} & 0 & 0 & 0 & x_1 \\ 0 & 0 & x_2 & -x_2 & 0 \\ 0 & x_3 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & -x_3 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ x_4 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}.$$

- In Step 3, first D is set to $1 + 1 + 1 = 3$ (the dimension of the endomorphism ring of ρ_H). The loop starts with $e = 2$, and the group relation $g^2 = h_s \in H$ in G yields the corresponding polynomial relation $X^2 - \rho_H(h_s) = 0$. The ideal generated by the entries of the LHS of this equation is

$$I = \langle x_1 x_4 - \frac{3}{4}, \quad x_2 x_3 + \frac{1}{2} \rangle.$$

  Clearly I is prime and has dimension 2 ($\{x_3, x_4\}$ is a maximally independent set for I, for example), so the loop can be exited immediately and the algorithm skips to Step 6.

- In Step 6, by including the 2 polynomials $x_3 - 1, x_4 - 1$ in the ideal, we obtain the solution vector $(c_1, c_2, c_3, c_4) = (\frac{3}{4}, -\frac{1}{2}, 1, 1) \in \mathbb{Q}^4$.

- Finally, in Step 7 we set $A = A_0 + \sum_{i=1}^{4} c_i A_i$, so:

$$\rho(g) = A = \begin{pmatrix} -\frac{1}{2} & 0 & 0 & 0 & \frac{3}{4} \\ 0 & 0 & -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & -1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 1 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}$$

- Applying the resulting ρ on the original generators g_1, g_2, we obtain:

$$\rho(g_1) = \frac{1}{8} \begin{pmatrix} 2 & -6 & -3 & 3 & 3 \\ -4 & 4 & -2 & 2 & 2 \\ 4 & 4 & 2 & 6 & -2 \\ -4 & -4 & 6 & 2 & 2 \\ 4 & 4 & 2 & -2 & 6 \end{pmatrix}, \quad \rho(g_2) = \frac{1}{8} \begin{pmatrix} 2 & -6 & -3 & -3 & -3 \\ 4 & -4 & 2 & 2 & 2 \\ -4 & -4 & -2 & 6 & -2 \\ 4 & 4 & 2 & 2 & -6 \\ 4 & 4 & -6 & 2 & 2 \end{pmatrix}$$

Notice the structure of the symbolic matrix X at the end of Step 2 and the corresponding ideal I of relations after Step 3. The square block diagonal submatrices with dimensions 1, 1, 3 respectively are constant, so these portions of $\rho(g)$ are unique. For any valid image matrix A, the only possible operations to modify it to another valid image matrix are:

- Multiply row 1 by a non-zero scalar s_1 and divide column 1 by s_1,
- Multiply row 2 by a non-zero scalar s_2 and divide column 2 by s_2,
- Multiply rows 3 to 5 by a non-zero scalar s_3 and divide columns 3 to 5 by s_3.

These operations correspond to the components of the endomorphism ring of ρ_H and do not modify the blocks on the diagonal.

5.4. Major Improvements to the Basic Algorithm

We now outline several major improvements to the basic GENERALEXTENSION algorithm; most of these involve the subalgorithm EXTENDRELATIONS. Every single improvement described here was absolutely necessary for the construction via general extension of several of the representations of very high degree of the sporadic groups.

5.4.1. The Polynomial and Ideal Operations. If B is the list of relation polynomials at any point in the algorithm, then whenever any new polynomials are created at any point, they should be reduced to normal form modulo B. This should be done not only when new polynomials are added to B, but especially after every intermediate product when a group order relation is being evaluated at the symbolic matrix $X = A_0 + \sum_{i=1}^{k} x_i \cdot A_i$ and the appropriate images of ρ_H.

Reducing every polynomial modulo B can cut down on the number of monomials enormously. The greatest reduction would occur if one could work in the residue class ring $R = F[x_1, \ldots x_k]/\langle B \rangle$, but to compute with elements of that ring would require computing a full Gröbner basis for the ideal generated by B each time it changes, which should be avoided until the set of relation polynomials becomes stable. Thus our implementation only uses the current basis B of the ideal to reduce by, instead of a full Gröbner basis, but this still can give a very significant reduction. When evaluating a group relation of degree d in the x_i variables, then there are potentially $\binom{k+d-1}{d}$ monomials in each relation polynomial, as noted above. But if these polynomials are reduced modulo B, then the number of monomials in each polynomial may be reduced to a number of the order of $H_I(d)$, the d-th coefficient in the Hilbert series of $I = \langle B \rangle$, and this number will often be much smaller.

As an example, when constructing the degree-2480 representation of the Lyons group below (p. 116), after order-3 group relations had been used, there were 6 variables and the leading monomials of the polynomials in the current set B were $\{x_1^2, x_1x_2, x_1x_3^2, x_2^4, x_5x_6\}$. The next group relation which would reduce the polynomial system further had to be an order-7 group relation. Now an inhomogeneous polynomial of degree 7 in 6 variables can have up to 1716 monomials, but the new relation polynomials were constructed modulo B and there were only 62 distinct monomials occurring in all the polynomials. So even this relation of rather high degree could be managed quite easily.

One can also generate more relation polynomials for B without evaluating group relations by successively computing a partial Gröbner basis as the algorithm progresses, as follows. Let DEGREEGROEBNERBASIS(B, M) denote the well-known simple variant of Buchberger's algorithm which:

- Takes as input a set B of polynomials in $F[x_1, \ldots, x_r]$ and a positive integer M;
- Computes a partial Gröbner basis of the ideal I generated by B by following Buchberger's algorithm, except that all S-polynomial pairs of degree greater than M are ignored;
- Interreduces and minimizes the resulting set of polynomials B' and returns B' (which generates I, even if it is not a full Gröbner basis for I).

The output of this algorithm does not necessarily equal the set of polynomials from a complete Gröbner basis of I which have degree up to M (it would if the input polynomials were all homogeneous, but this is never the case in the context of the general extension algorithm). Now whenever B is extended in EXTENDRELATIONS, by letting m be the maximum degree of the elements of B and then calling DEGREEGROEBNERBASIS with B and $M = m + 1$, the output B' will generate the same ideal as B and will not be too hard to compute because of the degree bound, but also:

- B' may contain polynomials which have smaller degree than those in B (because of non-trivial collapsing arising from the partial Gröbner basis computations) and in this case: (1) there may be some linear polynomials (so the number of image matrices and variables can immediately be reduced; see below) or (2) at least the normal forms of subsequent polynomials reduced modulo B' may have far less monomials.
- B' will typically contain many polynomials of degree $m + 1$, so when group order relations of higher degree are used, the normal forms of the new generated polynomials will have less monomials than otherwise, since they will be reduced by these extra polynomials.

It is thus much better in practice to use this algorithm instead of just interreducing the new set of polynomials whenever it is extended: even a partial Gröbner basis of B contains more information than the original set B. We have implemented an efficient implementation of Faugère's F_4 algorithm [Fau99] in MAGMA, and the truncated degree-M variant is easily implemented with some simple modifications.

To compute the dimension of an ideal I, our implementation uses the recursive search algorithm given in [BW93, Table 9.6]; this algorithm returns the dimension d and a maximally-independent set S of variables of cardinality d. As it stands, this algorithm has exponential complexity in the number of variables, and so can be hopeless if the dimension of the current ideal I is much larger than the target value $D - 1$ (this may occur near the start when very few polynomial relations have been gathered). However, it is easy to modify this algorithm so that one can give a lower bound L so that the algorithm will return as soon as it finds an independent set of variables of cardinality $d \geq L$. We have implemented this and use $L = D - 1$ (since the dimension of the relation ideal I at any point must be at least $D - 1$). Consequently, if the dimension of the current ideal is too large, then that is typically discovered immediately. Note also that an alternative method to compute the dimension is to compute the degree of the Hilbert polynomial via the algorithm in [BS92].

To test whether an ideal I is prime, we use the approach described in [GTZ88], [EHV92] or [BW93, 8.7] with some heuristic optimizations: the basic technique is that if I has dimension d and S is a maximally-independent set of cardinality d, then by moving the variables of S into a rational function field F we can reduce the problem to testing whether the corresponding zero-dimensional ideal over F is prime (which can done efficiently by an evaluation technique) and recursing on a suitable saturation of the ideal; the dimension must eventually decrease, ensuring termination [BW93, 8.8]. The prime testing is thus not a major issue for the ideals which arise in the algorithm.

5.4.2. Removing Linear Relations Progressively. Suppose that at any point, the set B of polynomial relations contains a polynomial of total degree 1. Then one matrix and symbolic variable can be removed (just as in the algorithm LINEARTRACEREDUCTION), as follows. Suppose the linear polynomial has the form:

$$x_l = c_0 + \sum_{j=1}^{l-1} c_j x_j,$$

where $c_j \in F$. Since the symbolic matrix X is written as $A_0 + \sum_{i=1}^{k} x_i \cdot A_i$, one can simply replace A_0 with $A_0 + c_0 \cdot A_l$ and replace A_i with $A_i + c_i \cdot A_l$ for $1 \leq i < l$, then remove A_l from $[A_1, \ldots, A_k]$ and decrease k and redefine X (see the proof of Thm. 4.4.1 for the details in a similar situation). At the same time, x_i should be replaced by $c_0 + \sum_{j=1}^{i-1} c_j x_j$ in each polynomial of B (equivalently, each polynomial can be reduced to normal form modulo this polynomial, assuming that x_i is greater than the other variables w.r.t. the monomial order). This reduction should be done successively for each linear polynomial in B.

We have found that this reduction always helps greatly and should be done immediately when possible: as noted above, when we generate polynomial relations of degree d from the k image matrices, there are up to $\binom{k+d-1}{d}$ monomials in the polynomials, so reducing k can reduce this number dramatically. Note that this situation is in contrast to the irreducible extension algorithm: recall that for that algorithm it is not necessarily advantageous to reduce the system as soon as each new linear relation is found, since a single reduction of the system can be expensive compared to the collection of more linear relations (see p. 87).

5.4.3. Representing the Symbolic Matrix. Suppose X represents the symbolic matrix

$$(A_0 + \sum_{i=1}^{k} x_i \cdot A_i) \in \mathcal{M}_n(F[x_1, \ldots, x_k]),$$

corresponding to the image of g, as the algorithm progresses. In our first implementation, we did actually represent X by an element of $\mathcal{M}_n(F[x_1, \ldots, x_k])$, i.e., by a matrix whose entries lie in the multivariate polynomial ring $F[x_1, \ldots, x_k]$. This made the implementation simple, since MAGMA easily supports the required matrix operations over multivariate polynomial rings. However, multiplication of multivariate polynomials can be very expensive, particularly when there are large number of variables, let alone large matrices over such polynomials!

It is better to represent the symbolic matrix as an element of $(\mathcal{M}_n(F))[x_1, \ldots, x_k]$. In our implementation, we represent such a matrix by a list of pairs of the form $\langle m_i, A_i \rangle$ where

m_i is a monomial in the $x_1, \ldots, x_k$ variables and $A_i \in \mathcal{M}_n(F)$. Multiplying two such matrices involves multiplying all pairs and then collecting the pairs with the same monomials and adding the corresponding matrices, etc. One should reduce all the product monomials modulo the current relations (as in Subsec. 5.4.1 above) before collecting them. This has the effect that all the matrix multiplications only involve matrices over F, so multivariate polynomial arithmetic is avoided and a fast modular matrix multiplication algorithm over F can be used. One can also use parallelism in multiplying all the pairs. So this representation of the symbolic matrix X leads to a great speedup in our implementation.

Note also that the matrices are often very sparse initially (arising from the Hom-module basis for the restriction to the subgroup L) but typically become denser after removal of linear relations which arise from both the call to LINEARTRACEREDUCTION and from subsequent order relations. Our implementation uses both sparse and dense representations for the image matrices, switching appropriately between these representations according to the density of each matrix.

5.4.4. Using the Action on a Smaller Matrix. Suppose again that X represents the symbolic matrix $A_0 + \sum_{i=1}^{r} x_i \cdot A_i$, corresponding to the image of g. The simple subalgorithm EXTENDRELATIONS finds $h, t \in H$ with $(hg)^e = t$ for $e > 1$ and then computes the n^2 polynomial relations coming from the matrix equation:

$$Y = (\rho_H(h) \cdot X)^e - \rho_H(t) = 0.$$

As n grows larger (in the hundreds, let alone thousands), this obviously becomes impractical to manage. Also, there tends to be a lot of redundancy: the number of distinct entries of Y after normalization (multiplying each polynomial by a scalar to make it monic) tends to be much less than n^2.

The following idea avoids this problem. Choose a positive weight $w < n$ (take $w = 10$ by default if $n > 10$) and then choose a $w \times n$ matrix W with small random entries in F (typically, random values from $\{-1, 0, 1\}$). Then the relations can be based on the multiplicative action of the symbolic matrices on W instead of full products of the symbolic matrices. That is, we can compute the wn polynomial relations coming from the matrix equation:

$$W \cdot (\rho_H(h) \cdot X)^e - W \cdot \rho_H(t) = 0.$$

Each term of the LHS of this equation should of course be computed by successively multiplying each intermediate $w \times n$ matrix by each new matrix on the right. This procedure thus avoids computing any full matrix product of two $n \times n$ matrices.

Clearly, the polynomial relations coming from the above matrix equation involving the action on W are just F-linear combinations of all the possible polynomial relations coming from the entries of Y. In practice, this seems sufficient to yield essentially the same relations. But the time and memory improvement is typically of the order of n/w, which is very significant when n is very large. For cases such as constructing the minimal-degree faithful representations of the Baby Monster group ($n = 4371$) and Fischer F'_{24} group ($n = 8671$) via general extension, where we used $w = 10$, the improvement was critical (see p. 119 and p. 121 respectively).

Recall that in the function LINEARTRACEREDUCTION, the product-replacement random algorithm was used on the images of elements of H to avoid recomputing $\rho_H(h)$ from scratch for each h (see p. 87). In contrast, in ExtendRelations it is better to recompute $\rho_H(h)$ for each h using the standard method of words in the strong generators of H, since it may be necessary to generate many random elements of H until an h is found with $(gh)^e \in H$ for the given e, and there are other elements of H at which ρ_H must be evaluated. Thus, unlike the situation in LinearTraceReduction, it is better not to compute all the corresponding image matrices under ρ_H in parallel while generating random elements of H. We have also added a variant to the kernel code in Magma for computing $W \cdot \rho_H(h)$, as follows. If `MH` is the FH-module corresponding to ρ_H, then the existing Magma function `Representation(MH)` returns a map f so that `f(h)` gives $\rho_H(h)$ for $h \in H$ (using the standard method of words in the strong generators). The new variant is called by `f(W, h)`, where W is a $w \times n$ matrix W, and returns $W \cdot \rho_H(h)$: again, instead of multiplying the full matrices out first, it evaluates the appropriate action on the $w \times n$ matrices by each successive matrix determined by the relevant word in the strong generators and their inverses, and thus avoids any multiplication of $n \times n$ matrices after the initial setup of ρ_H.

Finally, for huge H, it may be too hard even to compute a BSGS for H and write an arbitrary element of H as a word in the strong generators of H. So the algorithm can just try elements of H such h_1, h_2, h_1h_2, etc. until the product by g has reasonably small order so such elements of H can be used for group order relations. This technique was used for constructing the degree-8671 representation of Fi_{24}' where we only needed to use the group relations $g^2 = 1$ and $(gh_2)^8 = 1$, where h_2 was the second standard generator of H (see p. 121).

5.4.5. Using Inverses in Relations. Suppose that for the normalizing element $g \in G$, we have $g^2 = s \in H$. Then we after we initially include the polynomial relations coming from the relation $g^2 = s$ in B, we can reduce the degree of subsequent relations by splitting a relation into a LHS and RHS and using a symbolic image for g^{-1} which does not need inverses of the A_i. First note that

$$g^{-1} = gs^{-1} = s^{-1}g.$$

Then suppose we have $t = (h \cdot g)^e \in H$. Let e_r be $\lfloor \frac{e}{2} \rfloor$ and e_l be $e - e_r$, so $e = e_l + e_r$. Then

$$\begin{aligned}
(h \cdot g)^{e_l} &= (h \cdot g)^{-e_r} \cdot t \\
&= (g^{-1} \cdot h^{-1})^{e_r} \cdot t \\
&= (g \cdot s^{-1} \cdot h^{-1})^{e_r} \cdot t \\
&= (g \cdot u)^{e_r} \cdot t \quad [\text{where } u = (h \cdot s)^{-1}] \\
&= (g \cdot u)^{e_r - 1} \cdot g \cdot (ut).
\end{aligned}$$

Thus if X is the symbolic matrix representing the image of g, then we can use the polynomial relations coming from the matrix equation:

$$(\rho_H(h) \cdot X)^{e_l} = (X \cdot \rho_H(u))^{e_r - 1} \cdot X \cdot \rho_H(ut).$$

We have separated out the final gu in the RHS so that (ut) can be placed together (both $u, t \in H$) so one can multiply by the single matrix $\rho_H(ut)$ over F.

We thus have an equivalent relation but the degree in the variables is $e_l = \lceil \frac{e}{2} \rceil$ instead of e, which makes a huge difference in practice in the number of monomials occurring in the polynomials.

Combining this idea with the action on a smaller matrix W, our implementation always computes the relations via the matrix equation:

$$W \cdot (\rho_H(h) \cdot X)^{e_l} = W \cdot (X \cdot \rho_H(u))^{e_r - 1} \cdot X \cdot \rho_H(ut).$$

This idea can be extended to the case that $g^3 \in H$ (so g^{-1} can be written in terms of g^2 and elements of H) and so on.

5.4.6. The Advanced ExtendRelations **subalgorithm.** Combining all of the ideas in the 5 previous subsections, we can now present an advanced version of the subalgorithm ExtendRelations which is a lot more efficient than the original simple formulation (and matches our implementation fairly closely). In the last step of this new version, the algorithm performs the linear reduction as described in Subsec. 5.4.2 and then returns not only B' but the new $[A_0, A_1, \ldots, A_k]$ as well. So the original GeneralExtension just has to be modified so that in Step 3, $[A_0, A_1, \ldots, A_k]$ and k are updated to the value returned by the new ExtendRelations.

Subalgorithm ExtendRelations$(g, \rho_H, [A_0, A_1, \ldots, A_k], B, e)$ **[ADVANCED]**
Input and Output as for original ExtendRelations (p. 97) except that the new polynomial relation set B' **and** new $[A_0, A_1, \ldots, A_k]$ are returned.
Steps:

```
1. For $T$ tries, choose a random element $h \in H$ until $t = (h \cdot g)^e \in H$. If unsuccessful,
   return $B$, $[A_0, A_1, \ldots, A_k]$.
2. Set $X := A_0 + \sum_{i=1}^{k} x_i \cdot A_i \in \mathcal{M}_n(F)[x_1, \ldots, x_k]$.
3. Choose a positive weight $w \leq n$ (default 10) then a random $w \times n$ matrix $W$ with
   small random entries in $F$. In the following, compute $U \cdot \rho_H(h)$, etc. for any $h \in H$ by
   the above method with successive action on $w \times n$ matrices (see p. 106), thus avoiding
   computing $\rho_H(h)$ explicitly.
4. If $e > 2$ and $g^2 \in H$ then:
   {
      Set $s := g^2$ and $u := (h \cdot s)^{-1}$.
      Set $e_r := \lfloor \frac{e}{2} \rfloor, e_l := e - e_r$ [so $(hg)^{e_l} = (gu)^{e_r - 1} \cdot g \cdot (ut)$].
      Set $U_1 := W$.
      For i := 1 to $e_l$ do:
         Set $U_1 := ((U_1 \cdot \rho_H(h)) \cdot X) \bmod B$.
      Set $U_2 := W$.
      For i := 1 to $e_r - 1$ do:
         Set $U_2 := (((U_2 \cdot X) \bmod B) \cdot \rho_H(u)$.
      Set $U_2 := ((U_2 \cdot X) \bmod B) \cdot \rho_H(ut)$.
      Set $A := U_1 - U_2$.
   }
   Else:
   {
      Set $U := W$.
      For i := 1 to $e$ do:
            Set $U := ((U \cdot X) \bmod B) \cdot \rho_H(h)$.
        Set $A := U - W \cdot \rho_H(t)$.
    }
5.  Set $S$ to the set of all entries of $A$ and set $T := B \cup S$.
    Set $d$ to the maximum of the total degrees of the elements of $T$.
    Set $B' :=$ DegreeGroebnerBasis$(T, d+1)$.
6.  While $B'$ contains a linear polynomial $f_l$ do:
    {
        Write the normalized $f_l$ as $x_l - (c_0 + \sum_{j=1}^{l-1} c_j x_j)$.
        Set $A_0 := A_0 + c_0 \cdot A_l$.
        Set $A_i := A_i + c_i \cdot A_l$ for $1 \le i < l$.
        Remove $A_l$ from $[A_1, \ldots, A_k]$ and remove $f_l$ from $B'$.
        Replace $x_l$ by $c_0 + \sum_{j=1}^{i-1} c_j x_j$ in $f$ for all $f \in B'$.
        Replace $x_j$ by $x_{j-1}$ for $l < j \le k$ in $f$ for all $f \in B'$.
        Set $k := k - 1$.
    }
7.  Return $B'$ and $[A_0, A_1, \ldots, A_k]$.
```

5.4.7. The Quality of the Final Representation. The algorithm as stated does not consider the quality of the output (the size of the entries in the matrices), in that the final representation will depend on the choice of the solution point from the variety V.

In our implementation we have added another step before Step 6 which first reduces the basis given by the final matrices $[A_1, \ldots, A_k]$ (as an F-vector space) and applies the corresponding transformation to the relation polynomials. The reduction method is very similar to that used in the algorithm ReducedBasisForAction (p. 73): expand the F-basis over $\mathbb{Q}$, saturate it and apply LLL and then select a reduced F-basis corresponding to a suitable subset of this expanded $\mathbb{Q}$-basis. After this reduction, the default choice of ± 1 for the constants which the independent variables are set to in ElementOfVariety tends to yield a representation with very small entries in practice.

For some of the very high-degree representations described below, we have made a particular choice of constants for the solution point (after the above reduction), so as to keep the final entry numerators and denominators as small as possible. The next chapter gives an alternative way of reducing the result, which makes the particular choice of the point in the variety unimportant.

5.4.8. Finding a Normalized Subgroup in Large Matrix Groups. When the group G is so large that it has to be defined by a high-degree matrix group representation over a finite field in practice (such as some of the sporadic simple groups), it can be very difficult to compute a suitable subgroup L of H and $g \in G \setminus H$ with $L^g = L$, since the computation of normalizers is very difficult for such matrix groups.

We outline here a method which we have used to handle this situation, assuming that G is defined by an irreducible modular matrix representation.

1. First select a proper subgroup S of H (typically a maximal subgroup) and then search for a subgroup E of G which includes S but is not contained in H. This can be done (avoiding the computation of a BSGS) by repeatedly choosing a random element $t \in G \setminus H$ of very small order and setting $E = \langle S, t\rangle$ and testing whether E is reducible (via the modular Meataxe); if so, then E must be a proper subgroup of G, since G is irreducible.
2. Let π be some homomorphism from E to a smaller-degree representation. Usually one can use the representation given by some element of the composition series of the natural E-module (small enough so that one can compute effectively with this representation, but large enough to avoid too much collapsing).
3. Finally, let $E_\pi = \pi(E), H_\pi = \pi(H)$, find a subgroup L_π of H_π and $g_\pi \in E_\pi$, with $(L_\pi)^{g_\pi} = L_\pi, g_\pi \notin H_\pi$ (either by recursion or by using a simple loop over the subgroups of H_π) and then map all of these back via π^{-1} to L and g respectively in the original matrix representation of G. Since the kernel of π is a normal subgroup, it is clear that $L^g = L$ and $g \notin H$.

We have used this method when computing these irreducible representations via extension:

- The degree-248 and -4123 representations of the Thompson group (p. 115).
- The degree-1333 representation of the Janko Group J_4 (p. 116).
- The degree-1938 representation of ${}^2\mathrm{E}_6(2)$ (p. 184). (Here L had order 174182400 and there were 27 initial image matrices.)
- The degree-2480 representation of the Lyons group (p. 116).
- The degree-4371 representation of the Baby Monster group B (p. 119).
- The degree-64 and degree-128 representations of $2.\mathrm{A}_n$ for $n = 13, 14, 15, 16, 17$ (p. 171, etc.). Non-trivial modular matrix representations[2] are used to define these groups since permutation representations are too large. Now for each group $G = 2.\mathrm{A}_n$, instead of searching for the subgroup E as above, one can of course just let π be the (non-faithful!) degree-n permutation representation of G with image equal to A_n and then proceed as in Step 3 above.

5.5. Examples

This section contains some basic examples of general extension. Later sections in this chapter describe in detail how general extension was used to construct representations of the very large sporadic groups.

Example 5.5.1. Let $G = 6.\mathrm{M}_{22}$ and let χ be one of the minimal-degree faithful characters of G; χ has degree 66, Schur index 1 and character field $F = \mathbb{Q}(\alpha)$, where α has minimal polynomial $x^4 - 5x^3 + 8x^2 - 7x + 7$. A typical call to AbsolutelyIrreducibleRepresentation on χ constructs an F-representation affording χ in about 1370s, with entries having 32-digit numerators and common denominator 1, so this is an example where it is hard to construct an absolutely irreducible representation with small entries.

As a better alternative, we computed a representation ρ affording χ using general extension, as follows (table entry on p. 166). Let H be the maximal subgroup of G with shape $2.2^4.3.\mathrm{A}_6$ (order 34560, index 77). Then $\chi_H = \chi \downarrow_H$ splits over F as 30+36. Note that the

[2] Provided by D.F. Holt for $n = 15, 16, 17$.

degree-36 representation is not absolutely irreducible, but both of these representations can be realized (minimally) over $\mathbb{Q}(\zeta_3)$, which is a subfield of F. Representations over F affording these characters were constructed by IrreducibleRepresentationsOverField in only 3.4s. Then GeneralExtension was called with χ and these representations of H. A normalized subgroup $L \leq H$ of order 2160 and $g \in G \setminus H$ with $L^g = L$ and $g^2 \in H$ was instantly found, with 18 initial image matrices. Linear reduction reduced this to 12 matrices and then the single group relation $g^2 = h_1 \in H$ yielded 6 linear polynomial relations and an ideal in 6 variables of dimension 2 which was the required dimension, since the norm of χ_H equals 3 (1.4s). Then a solution matrix was instantly constructed and the rewriting of the representation to be defined on the standard generators g_1, g_2 of G took 1.4s, so the whole of GeneralExtension took only 2.8s total. Since we could initially conjugate H so that $g_1 \in H$, $\rho(g_1)$ is very sparse (at most 3 non-zero entries per row), while $\rho(g_1)$ has density 67.2% and absolute maximum numerator 17 and common denominator 32; typical entries are $\frac{1}{32}3(\alpha^3 - 12\alpha^2 + 16\alpha - 7)$, $\frac{1}{8}(-\alpha^3 + 3\alpha^2 - 2\alpha + 2)$.

Example 5.5.2. Let G be the sporadic simple Suzuki group Suz and let χ be the minimal-degree faithful character of G; χ has degree 143 and is rational with Schur index 1. We computed a representation ρ affording χ using general extension, as follows (table entry on p. 171). Let H be the largest maximal subgroup of G, which equals $\mathrm{G}_2(4)$ (index 1782); $\chi_H = \chi \downarrow_H$ splits as $65 + 78$ and rational representations affording these characters were found in 10s (via IrreducibleRationalRepresentations). Then GeneralExtension was called with χ and these representations of H. The subgroup $L \leq H$ of order 604800 was instantly found with 9 corresponding image matrices. Linear reduction reduced this to 2 image matrices, then the initial square group relation reduced the system to the single relation $x_1x_2 = \frac{3}{8}$, from which an image of g was easily constructed (0.4s). Finally, the representation was rewritten to be defined on the standard generators of G in 1.5s to obtain $\rho : G \rightarrow \mathrm{GL}_{143}(\mathbb{Q})$, which has absolute maximum 2-digit numerators and denominator LCM 4. So the general extension algorithm took only 2.4s after the representations of H were set up. We also conjugated ρ to an integral representation in 0.8s; the result has 2-digit entries.

Example 5.5.3. In this example, the conic method is needed in the subalgorithm ElementOfVariety. Let $G = 3.\mathrm{O'N}{:}2$, which is the automorphism group of $H = 3.\mathrm{O'N}$. A minimal-degree faithful representation of G has degree 684 and is realized over the quadratic field $F = \mathbb{Q}(\sqrt{-6})$. Let χ be one of the corresponding characters. We computed such a representation by general extension, as follows (table entry on p. 201). We had already computed the absolutely irreducible representation $\sigma_{342} : H \rightarrow \mathrm{GL}_{342}(F_2)$ where $F_2 = \mathbb{Q}(\beta)$ with defining polynomial $x^4 + 2x^2 + 4$ (table entry on p. 178). Since F is a subfield of F_2, we could immediately compute the restriction to scalars representation $\sigma_{684} : H \rightarrow \mathrm{GL}_{684}(F)$ of σ_{342} from F_2 to F (via Prop. 1.6.2); σ_{684} is irreducible over F, but not absolutely irreducible. When we applied GeneralExtension to χ and σ_{684}, there were 2 initial image matrices, no linear trace reduction, and the initial square group relation yielded one quadratic polynomial relation (354s; time totally dominated by ExtensionImageSetup). The single relation was:

$$x_1^2 - 2x_2^2 + \alpha x_1 x_2 + \frac{1}{2401}(\alpha + 5) = 0 \quad [\alpha = \sqrt{-6}].$$

A rational point $(\frac{1}{49}(-2\alpha+1), \frac{3}{49}) \in F^2$ on the corresponding conic was then computed in 0.2s and this yielded a suitable image matrix over F. The total time was 359s (starting with the precomputed σ_{324}).

5.6. General Extension Without Explicit Use Of The Character

We now describe a practical variant of GENERALEXTENSION where the character χ does not need to be used explicitly by the algorithm. This variant is useful when G is so large that is not practical to compute the conjugacy classes of G or work with characters of G explicitly. All that is needed to be known about χ explicitly on the computer is the decomposition of $\chi_H = \chi \downarrow_H$ into irreducible characters for some subgroup H of G so that a suitable ρ_H affording χ_H can be set up first. Very often, basic theory or a manual inspection of the ATLAS [CCN+85] reveals how χ decomposes w.r.t. a given maximal subgroup H; we have done exactly this for several of the representations of the very large sporadic groups.

The first simple modifications to the original algorithm (p. 97) are the following: in Step 2, simply set k to l and initialize A_0 to zero instead of calling LINEARTRACEREDUCTION, and omit Step 5. There is thus never any use of χ explicitly in this variant of the algorithm.

Now suppose that ρ_H is the fixed input representation which affords χ_H. Let C be the number of representations of G of degree $n = \chi(1)$ which are extensions of ρ_H (it is generally easy to determine C in practice by examining the Galois-conjugacy class of χ, inspecting the character table of G, H, etc.). If $C > 1$, then algebraic relations from words involving elements of G and H may be insufficient to determine a unique representation (up to equivalence) which affords χ and extends ρ_H. But this can be easily handled by loosening the condition that the relation ideal I must be prime over the target field F, while keeping the same dimension condition. When the algorithm reaches the correct dimension and includes linear relations for the maximally-independent variables in ELEMENTOFVARIETY, so that the corresponding ideal J is zero-dimensional, there will be a finite number s of points in $\mathbf{V}_F(J)$ (instead of the usual single element when I is prime and J is maximal). Now $s \geq C$ always, so while $s > C$, there are not enough relations yet to determine all degree-n extensions of ρ_H, so the algorithm must proceed further to gather more relations.

When $s = C$, any solution to the polynomial system must yield a valid degree-n extension of ρ_H to G. So if the characters of all extensions of ρ_H to G are only the conjugates of χ, then we can just use any solution and then find the desired conjugate of the resulting representation. But there may also be degree-n extensions of ρ_H to G whose characters are not conjugate to χ. In each case, by evaluating traces of images of the generators and small-length products of these, we can determine enough of the character of any computed representation to identify it, and so we can select the solution which gives a representation affording the particular character we desire (and there are at most C possible solutions which must be considered).

There is one other important optimization for this variant of the algorithm. Let F be the minimal field over which χ is to be realized. Now the minimal field S over which ρ_H is written may be a proper subfield of F (e.g., very often $S = \mathbb{Q}$ while $F \neq \mathbb{Q}$). In this case, since χ is not explicitly used in the algorithm, all computations up to Step 3 can be done over S instead of F, and the main loop of Step 3 can be exited when the relation ideal I becomes prime over S (but not necessarily prime over F) and has the correct dimension.

Then $\mathbf{V}_S(I)$ will be empty (since otherwise it would imply a representation affording χ, realizable over S), but $\mathbf{V}_F(I)$ must be non-empty, so a solution over F can be found, yielding the desired representation over F. An illustration of this is given in Ex. 5.6.2 below, and a very large example which also benefits from this situation is the construction of the degree-2480 irreducible representation of the sporadic Lyons group for which a minimal field is $F = \mathbb{Q}(\sqrt{-11})$: here the relevant representation of H is irreducible and written over $\mathbb{Q}$ and the final relation ideal I is prime over $\mathbb{Q}$ but has two prime components over F; see p. 116 for details.

Several of the high-degree representations of the sporadic simple groups were constructed by this variant algorithm, as will be seen in the subsequent sections of this chapter. But it can also be useful for any degree size when ρ_H is absolutely irreducible, since it often runs faster than the irreducible extension algorithm of the previous chapter. Instead of gathering k independent linear relations for the k initial image matrices, the variant algorithm may compute the unique image matrix for g more quickly by constructing and solving a suitable polynomial system (via one or two group relations). The first of the following examples demonstrates this situation.

Example 5.6.1. Let $G = 3.\mathrm{U}_9(2)$, of order 976419878163325334323200 ($\sim 9 \times 10^{23}$). G has two conjugate minimal-degree irreducible representations of degree 171, which can be realized over $F = \mathbb{Q}(\zeta_3)$; let χ be one of the corresponding characters.

Let H be the maximal subgroup of G of order 150698880 and index 6479277604208640 which is equal to $3.\mathrm{J}_3$. Now $\chi_H = \chi \downarrow_H$ is absolutely irreducible and we had already computed a representation ρ_H affording χ_H (p. 173), so we could compute a representation affording χ by applying IrreducibleExtension to χ and ρ_H; we originally did exactly this before we had developed the general extension algorithm. But the initial computation of the character table of G in Magma took 9.6 days! After that, IrreducibleExtension on χ and ρ_H took only 155s, using a normalized subgroup L of order 3456, with 32 initial image matrices.

Alternatively, we were able to construct a representation affording χ more quickly by using the above variant of GeneralExtension without explicit use of the character χ (table entry on p. 173), thus avoiding the computation of the character table for G completely. This time we only used the given ρ_H and the knowledge that it extends to an irreducible representation of G over F. With the same L as above, the general extension algorithm again started with 32 image matrices and then used group order relations with orders 4, 4, 6, 10 respectively. Using the order 4, 4, 6 relations only determined a zero-dimensional ideal in 2 variables whose lexicographical Gröbner basis is:

$$\{\ x_1 - \frac{6}{5}x_2^3 + \frac{1}{10}(-\zeta_3 - 1)x_2, \quad x_2^4 + \frac{1}{12}(\zeta_3 + 1)x_2^2 + \frac{1}{144}\zeta_3\ \}.$$

The variety of this ideal over F has cardinality 4:

$$\{\pm(\frac{1}{60}(\zeta_3 - 1), \frac{1}{6}(\zeta_3 - 1)), \quad \pm(\frac{1}{60}(\zeta_3 + 2), \frac{1}{6}(-2\zeta_3 - 1))\}.$$

But there should only be 2 solutions, since the Galois orbit of χ has cardinality 2, so 2 of these solutions must not give a valid image for g. But after an order-10 relation was included, the relation ideal collapsed to being generated by a single polynomial in 1 variable: $x_1^2 + \frac{1}{10}\zeta_3$, with variety $\{\pm\frac{1}{60}(\zeta_3 - 1)\}$ over F. Each solution gives a valid extension of ρ_H, which affords χ or its conjugate. This time it took only 56.0s to compute the solution image and 13.4s to rewrite the representation on the original generators of G, for a total of 69.4s, so using this method was in fact faster than IRREDUCIBLEEXTENSION, even after the character table had been computed.

The degree-170 irreducible representation of $G = \mathrm{U}_9(2)$ can be handled similarly (rational character with Schur index 2; table entry on p. 172). Here we used $H = \mathrm{J}_3$ (index 6479277604208640 again) and GENERALEXTENSION with G and the direct sum of the two conjugate degree-85 representations of H over $F = \mathbb{Q}(\zeta_3)$, again without using the character explicitly (or having to compute the character table of G). The normalized subgroup L had order 1152, there were 48 initial image matrices, and group order relations with orders $4, 4, 4, 6, 10, 17$ produced the dimension-1 ideal generated by:

$$\{x_1 x_2 + \frac{1}{35020800}\}$$

The total time taken for the general extension was 85s. Note that before the order-17 relation was used, the ideal contained the above polynomial and also one quadratic relation $x_3^2 + \frac{1}{70041600}$. So the algorithm had to go all the way to an order-17 relation to produce a linear polynomial which was the correct factor of the above polynomial. Despite the very high order, this was easily handled since the polynomials were always reduced modulo the current relations.

The degree-121 and -122 representations of $\mathrm{S}_{10}(3)$ and $2.\mathrm{S}_{10}(3)$ respectively (p. 170) were computed in a similar way. The order of the latter group is about 10^{26} and it is currently hopeless to compute its conjugacy classes or character table in MAGMA. Our general extension algorithm was in fact first developed to compute these particular representations!

Example 5.6.2. The degree-783 representation of $3.\mathrm{Fi}_{24}'$ (realized over the character field $F = \mathbb{Q}(\zeta_3)$) was computed by general extension without explicit use of the character (table entry on p. 181). Here the subgroup H was equal to Fi_{23} (index 920808) and the restriction to H splits as $1 + 782$ over $\mathbb{Q}$. The normalized subgroup $L \leq H$ was equal to $2.\mathrm{Fi}_{22}$ and there were 6 initial image matrices; then group order relations of orders $8, 8, 9$ reduced this to a system with 3 variables and the corresponding ideal I of $\mathbb{Q}[x_1, x_2, x_3]$ generated by:

$$x_1 x_2 - \frac{25}{7452}, \quad x_3^2 + \frac{1063}{13248} x_3 + \frac{681073}{175509504}.$$

I has dimension 1 and is prime over $\mathbb{Q}$, but over $F = \mathbb{Q}(\zeta_3)$ the second polynomial has the two roots:

$$\frac{1}{13248}(729\zeta_3 - 167), \quad \frac{1}{13248}(-729\zeta_3 - 896),$$

which yield the two conjugate representations of G.

In the rest of this chapter, we give detailed descriptions of how we constructed several of the high-degree ordinary representations of the larger sporadic groups by the general extension algorithm (most of them with the variant method without explicit use of the character). The order followed is roughly the order of difficulty (and the order in which they were constructed). Most of the representations above degree 1000 had never before been explicitly constructed.

5.7. The degree-248 and -4123 representations of the Thompson Group

Let G be the sporadic simple Thompson group, of order

$$90745943887872000 = 2^{15}.3^{10}.5^3.7^2.13.19.31.$$

We computed the degree-248 and -4123 irreducible rational representations of G, which are the first two faithful representations of G by degree (table entries on p. 176 and p. 186 respectively). The degree-248 modular matrix representation over $\mathbb{F}_2$ was used to define G. It is too difficult to compute the classes or character table of G in practice in MAGMA, so we used the variant of the general extension algorithm without explicit use of the character (Sec. 5.6) in both cases.

Let H be the second largest maximal subgroup of G, which equals $2^5.\mathrm{L}_5(2)$ (the so-called 'Dempwolff group', of index 283599225). A suitable subgroup L of H and $g \in G \setminus H$ with $L^g = L$ were constructed by the advanced method of Subsec. 5.4.8 (p. 109), as follows. Let S be the largest maximal subgroup of H (shape $2.2^{4+4}.\mathrm{A}_8$, index 31). Random search yielded an $r \in G$ of order 2 so that the subgroup $E = \langle S, r\rangle$ of G was a reducible matrix group (2424 tries; 71s). Now E was equal to $2.2^8.\mathrm{A}_9$ (order 92897280; another maximal subgroup of G), so it could be mapped via a homomorphism π onto a permutation representation E_π of A_9. Then instantly a subgroup of $\pi(S)$ normalized by an element of E_π was found and then these were mapped back, thus yielding $L < H$ and $g \in G \setminus H$ with $L^g = L$; L had order 1290240 and g had order 8, with $g^2 \in H$.

To find the relevant representations of H, we could use the permutation representation of H degree 7440 (with matching standard generators). The degree-248 representation σ_{248} of H was first computed in 105s by inducing a degree-8 representation of an index 31 subgroup; the degree-8 representation was constructed using IRREDUCIBLERATIONALREPRESENTATIONS.

The degree-248 representation of G was then computed by using general extension without use of the character, applied to G and σ_{248}, using the above L and g. There were only 3 image matrices, then the initial square relation and order relations for order 3 and 13 produced a maximal relation ideal in only 4.0s, yielding a unique solution for the image of g. Then it took only 0.6s to rewrite the representation so that it was defined on the standard generators.

Now let χ be the degree-4123 irreducible rational character of G; $\chi_H = \chi \downarrow_H$ splits over $\mathbb{Q}$ as $155 + 248 + 3720$. The degree-155 and -3720 representations of H were constructed by exact induction of linear representations in 53s. General extension was then applied without explicit use of the character to G and the direct sum of the 3 representations of H, using the same L and g. There were 33 initial image matrices (2892s; the restriction to L and L^g was trivial the generators of H had been extended to include those of L and L^g before constructing the representations of H). The initial square group relation yielded 28 degree-2 polynomial relations (426s), then an order-3 group relation gave 19 linear relations, reducing to 14 variables (132s). Next, an order-7 group relation gave 10 more linear relations, reducing to 4 variables and 2 degree-2 relations (490s); the ideal was then equal to $\langle x_1x_4 + \frac{1}{512}, x_2x_3 + \frac{9}{16384}\rangle$, which is prime with dimension 2, so the relation gathering could stop. Using the point $(\frac{1}{32}, \frac{3}{128}, -\frac{3}{128}, -\frac{1}{16})$ from the variety, the image matrix for g was constructed in 2.5s. Let the standard generators of G be g_1, g_2. Since g_1 was in H, $\rho(g_1)$ was trivial to construct and is sparse. The construction of $\rho(g_2)$ took 48s; the density is 92.6%, the absolute maximum numerator is 91 (average 4.1) and the denominator LCM is 512. The total time taken was 4643s (1.3h).

5.8. The degree-1333 representation of the Janko Group J_4

Let G be the sporadic simple Janko group J_4, of order

$$86775571046077562880 = 2^{21}.3^3.5.7.11^3.23.29.31.37.43.$$

A minimal-degree faithful ordinary representation of G has degree 1333. Let χ be one of the corresponding characters; the character field is $F = \mathbb{Q}(\sqrt{-7})$. We constructed a representation $\rho : G \longrightarrow \mathrm{GL}_{1333}(F)$ affording χ by general extension (table entry on p. 183). A degree-112 representation over $\mathbb{F}_2$ was used to define G; since it is too difficult to compute with characters explicitly, we again used the general extension algorithm with no explicit character for G.

Let H be the largest maximal subgroup of G, which equals 2^{11}:M_{24}; $\chi_H = \chi\downarrow_H$ splits over F as 45+1288. The degree-45 representation was constructed by irreducible extension of the degree-45 representation of M_{24} to H (3s), and the degree-1288 representation was constructed by direct induction of a linear representation of a subgroup of index 1288 (2s).

For the general extension, the normalized subgroup $L \leq H$ and element $g \in G \backslash H$ with $L^g = L$ was again constructed by the advanced method of Subsec. 5.4.8; the resulting L had order 33030144 (10s), which yielded 8 initial image matrices (996s). The square group relation gave 6 degree-2 polynomial relations (4s), then an order-3 group relation gave 6 linear relations and one degree-2 relation (84s), reducing the number of variables to 2. The ideal now had the required dimension 1 and was generated by the single polynomial:

$$x_1x_2 - \frac{1}{256}.$$

Setting $x_1 = x_2 = \frac{1}{16}$ gave a valid image matrix for g and then the rewriting of the representation to be defined on the standard generators g_1, g_2 of G took 223s, yielding $\rho : G \longrightarrow \mathrm{GL}_{1333}(F)$. The first image matrix $\rho(g_1)$ is sparse, while the second image matrix $\rho(g_2)$ has density 85.6%, with denominator LCM 128 and absolute maximum numerator 14 (average 0.8). The total time taken was 1354 seconds.

5.9. The degree-2480 representation of the Lyons Group

Let G be the sporadic simple Lyons group, of order

$$51765179004000000 = 2^8.3^7.5^6.7.11.31.37.67.$$

A minimal-degree faithful ordinary representation of G has degree 2480. Let χ be one of the corresponding characters; the character field $F = \mathbb{Q}(\chi)$ equals $\mathbb{Q}(\sqrt{-11})$. We constructed a representation $\rho : G \longrightarrow \mathrm{GL}_{2480}(F)$ affording χ, as follows.

A degree-111 representation over $\mathbb{F}_5$ was used to define G. Let H be the maximal subgroup of G equal to the non-split extension $5^3.L_3(5)$ (order 46500000, index 1113229656), which can be constructed using words from the online ATLAS. Then $\chi_H = \chi\downarrow_H$ is also absolutely irreducible, so a representation for χ can be computed via irreducible extension from a representation affording χ_H.

We first computed a representation $\rho_H : H \to \mathrm{GL}_{2480}(\mathbb{Q})$ affording χ_H as follows. Let H_2 be the largest maximal subgroup of H (order 1500000, index 31 in H, shape $2^5.5^5.A_5$, with a faithful degree-150 permutation representation). Now χ_H restricted to H_2 splits as:

$$80 + 240 + 240 + 480 + 480 + 480 + 480$$

(all absolutely irreducible over $\mathbb{Q}$ with Schur index 1). Corresponding representations were computed in 549s using the IrreducibleRationalRepresentations algorithm; they were all integral with 1-digit entries. Then the default general extension algorithm was applied to χ_H and the block-diagonal sum of these representations of H_2. A subgroup L_2 of H_2 of order 50000 normalized by an element h of $H \setminus H_2$ was found in a few seconds; this yielded 136 initial image matrices for h. Linear reduction via χ_H reduced this to 112 image matrices. One square group relation and two order-3 group relations reduced it to 42 image matrices and a corresponding prime relation ideal of dimension 6 which was non-trivial but manageable (the lexicographical Gröbner basis consisted of 441 degree-2 polynomials!). Since this was the required dimension for the ideal (since there were 7 absolutely irreducible representations of H_2), a particular solution for the image of the normalizing element h could then immediately be constructed, and then the representation was rewritten on the generators of H_2. The resulting representation $\rho_H : H \to \mathrm{GL}_{2480}(\mathbb{Q})$ has entries with a maximum of 2 digits and denominator LCM 5^4. The total time for the construction of ρ_H was 4519s.

Since G is too large to compute with characters explicitly, the irreducible extension algorithm could not be used to construct the representation of G, so we used the general extension algorithm with no explicit character. To compute suitable $L \leq H$ and normalizing element g, we again used the advanced method of Subsec. 5.4.8. Let S be the largest maximal subgroup of H (order 1500000, index 31). Random search yielded an $r \in G$ of order 2 so that the subgroup $E = \langle S, r\rangle$ of G was a reducible matrix group (787 tries; 12s). Here E was equal to $\mathrm{G}_2(5)$ (order 5859000000; another maximal subgroup of G). This could first be mapped to a degree-7 matrix representation over $\mathbb{F}_5$, and then to the (faithful) permutation representation E_π of degree 3906. It was then easy to find a suitable subgroup L_π of the image S_π of S and normalizing element $g_\pi \in E_\pi$ and map these back to $L \leq H$ (order 50000) and $g \in G \setminus H$ in the degree-111 representation over $\mathbb{F}_5$ (23s).

The extension from ρ_H to G with this L and g could finally be done, as follows. The restriction of ρ_H to L and L^g took 10872s and the relevant Hom-module computation yielded 136 initial matrices (18204s). The square group relation gave 121 degree-2 polynomial relations (4712s), then an order-3 group relation gave 23 linear relations and 634 degree-2 relations (4272s), reducing the number of variables to 88. Another order-3 group relation gave 82 linear relations and 3 degree-2 relations (272s), reducing the number of variables to 6. At this point, there were now 6 variables, but the relation ideal had dimension 3 (and dimension 0 was needed because ρ_H was absolutely irreducible). Group order relations for orders 4, 5 and 6 did not change the ideal of relations (610s). Finally, an order-7 group relation gave 5 linear relations and 1 degree-2 relation (493s) and so there was now only one variable and the corresponding ideal had dimension 0 and was prime over $\mathbb{Q}$, generated by the single polynomial:

$$x_1^2 - \frac{2}{25}x_1 + \frac{69}{15625}.$$

Setting $x_1 = \frac{1}{125}(-2\alpha + 5)$ in $F = \mathbb{Q}(\alpha)$ (where $\alpha = \sqrt{-11}$) yielded a valid image matrix for g. Finally, computing the corresponding images of the standard generators g_1, g_2 of G took 41s (via the sparse ρ_H) and 1274s respectively to yield the final representation $\rho : G \to \mathrm{GL}_{2480}(F)$ affording χ. (As usual, we had first conjugated H so that the first standard generator g_1 was in H.)

The density of $\rho(g_1)$ is only 0.15% (about 4 non-zero entries per row), with all non-zero entries equal to ±1. The density of $\rho(g_2)$ is 99.7%, with denominator LCM $5^6 = 15625$ and the numerators have at most 4 digits (average 18.1). The total time for the general extension of ρ_H to G was 69656s (19.3h).

Since 2 did not divide the denominators in the final representation ρ, we could directly reduce ρ modulo 2 to obtain a degree-2480 representation of G over $\mathbb{F}_4$. It was then easy to verify in about a minute (by the standard modular Meataxe tools) that this modular representation is equivalent to the corresponding irreducible representation in the online ATLAS which was computed by Wilson in [Wil98b].

5.10. The degree-782, -3588 and -5083 representations of the Fischer Group Fi_{23}

Let G be the sporadic simple Fischer group Fi_{23}, of order

$$4089470473293004800 = 2^{18}.3^{13}.5^2.7.11.13.17.23.$$

and can be defined by a degree-31671 permutation representation. The first three faithful representations of G have degrees 782, 3588 and 5083 respectively and can all be written over the rational field. We first computed the degree-782 representation in 596s by IrreducibleRationalRepresentations (see p. 64).

The two larger degree-3588 and degree-5083 representations (needed for the computation of representations of the Baby Monster [p. 119] and Fischer F_{24}' [p. 121] respectively) were computed by general extension via the maximal subgroup H equal to $2^{11}.\mathrm{M}_{23}$ (index 195747435 in G). First a sufficiently large normalized subgroup $L < H$ of order 41287680 (shape $2^4.2^4.2^6.\mathrm{A}_7$) was found, with a normalizing element $g \in G \setminus H$ (12s); these were used in both extensions.

Degree 3588: Let χ be the degree-3588 irreducible character of G; $\chi_H = \chi \downarrow_H$ splits over $\mathbb{Q}$ as $1 + 22 + 253 + 506 + 1288 + 1518$. Corresponding representations were computed easily: the degree-22 representation was trivially derived from a permutation representation of degree 23 of H, while the other representations were computed by direct induction of representations of degree 1 or 6 for suitable subgroups. Then general extension was applied to χ and the direct sum of these representations of H with the above L and g. There were 43 initial image matrices; linear reduction via the character took this down to 27 variables, and then group order relations with orders 2, 4, 6 reduced this to 22 variables and a corresponding prime ideal of required dimension 5 (6871s). Since this is a rather non-trivial ideal, we give the lexicographical Gröbner basis of the ideal out of interest:

$$\{\ x_1 + 2x_2x_{11},\ \ x_2x_{10} - \frac{63}{92},\ \ x_3 - \frac{28}{3}x_6x_{20},\ \ x_4 - \frac{28}{3}x_6x_{10}x_{19},$$

$$x_5 + \frac{14}{3}x_6x_{19},\ \ x_6x_{11} + \frac{14}{69}x_9x_{14},\ \ x_6x_{15} - \frac{1}{6}x_{14},\ \ x_6x_{22} - \frac{3}{28},$$

$$x_7 - \frac{112}{3}x_9x_{14}x_{21},\quad x_8 + 8x_9x_{14},\quad x_9x_{15} + \frac{23}{28}x_{11},\quad x_9x_{18} - \frac{3}{7}x_{17}x_{20},$$
$$x_9x_{19} + \frac{3}{28}x_{20},\quad x_9x_{22} + \frac{92}{7}x_{11}x_{20},\quad x_{11}x_{18} + \frac{3}{92}x_{17}x_{22},$$
$$x_{11}x_{19} - \frac{3}{368}x_{22},\quad x_{12} - \frac{28}{3}x_{14}x_{21},\quad x_{13} - \frac{56}{9}x_{14}x_{19},\quad x_{14}x_{20} - \frac{9}{224},$$
$$x_{14}x_{22} - \frac{9}{14}x_{15},\quad x_{15}x_{20} - \frac{1}{16}x_{22},\quad x_{16} - 4x_{17}x_{20},\quad x_{17}x_{19} + \frac{1}{4}x_{18},$$
$$x_{17}x_{21} + \frac{1}{32},\quad x_{18}x_{21} - \frac{1}{8}x_{19}\ \}.$$

Note that even though there are several variables, this basis has the structure discussed in point 5 on p. 100, since the representations of H are all absolutely irreducible, with multiplicity 1. The point of the variety of the ideal was:

$$\left(\frac{1323}{184}, \frac{63}{92}, \frac{1}{644}, -\frac{1}{644}, \frac{1}{1288}, \frac{3}{644}, -\frac{9}{448}, -\frac{27}{28}, \frac{3}{28}, 1,\right.$$
$$\left. -\frac{21}{4}, -\frac{3}{64}, -\frac{1}{4}, \frac{9}{8}, \frac{161}{4}, 1, 7, 1, -\frac{1}{28}, \frac{1}{28}, -\frac{1}{224}, 23\right).$$

Computing the corresponding images of the standard generators of G took 51s (via the diagonal block representation of H) and 9086s respectively. The denominator LCM for the defining matrices is $2^{11}.7.23$ and the absolute maximum numerator is 12285. The total time to compute this representation was 4.4 hours.

Degree 5083: Let χ be the degree-5083 irreducible rational character of G. Then $\chi_H = \chi \downarrow_H$ splits over $\mathbb{Q}$ as 253+1288+3542. The first two representations were computed above, while the degree-3542 representation was computed as the direct induction to H of a degree-7 representation of a subgroup of H of index 506, with shape $2^{11}.\mathrm{A}_8$. Again, general extension could then applied with the same L and g as above. There were 20 initial image matrices; linear reduction via χ reduced this to 8 variables, then group relations for orders 2 and 4 reduced this to 6 variables with a corresponding prime relation ideal of dimension 2, as required. Computing the corresponding images of the standard generators g_1, g_2 of G took 204s (via the diagonal block representation of H) and 27303s respectively. The image of the first generator is sparse and its non-zero entries are only ± 1. Interestingly, despite being dense, the image of the second generator of G has only 71 different entries, with LCM denominator 256 and absolute maximum numerator 39.

5.11. The degree-4371 representation of the Baby Monster Group

Let G be the Baby Monster sporadic simple group, of order

$$4154781481226426191177580544000000 = 2^{41}.3^{13}.5^6.7^2.11.13.17.19.23.31.47.$$

A minimal-degree faithful ordinary representation of G has degree 4371 and can be realized over $\mathbb{Q}$. We constructed such a representation explicitly over $\mathbb{Q}$ by general extension without explicit use of the character (table entry on p. 186). The degree-4370 modular representation over $\mathbb{F}_2$ was used to define G. Considering the huge size of G (by far the largest group for which we computed an ordinary representation) and the matrices by which G is defined, the effectiveness of our GeneralExtension algorithm can be seen in that the only computations involving G itself were elementary group arithmetic operations for: (1) finding the subgroup L (with some use of the modular Meataxe), (2) finding group order relations within the subalgorithm ExtendRelations and (3) the rewriting of the final representation on the standard generators. All of this was quite feasible: using the above modular matrix representation of G, Magma can multiply two elements of G in about 0.2s and invert an element in about 0.6s (we have implemented fast algorithms for these operations, similar to those described in [ABH10]).

Let χ be the degree-4371 irreducible character of G and let H be the third largest maximal subgroup of G, which equals Fi_{23} and has index 1015970529280000 in G. The restricted character $\chi_H = \chi \downarrow_H$ splits as $1 + 782 + 3588$. The appropriate representations of H had already been computed (see p. 118).

To compute suitable $L \leq H$ and normalizing element $g \in G \setminus H$, we used the advanced method of Subsec. 5.4.8. Let S be the largest maximal subgroup of H which equals $2.F_{22}$ (order 129123503308800). Random elements of G of order 2 were generated until the extension of S by such an element was a reducible matrix group (869 tries, 6844s; for each try, it took about 0.4s to generate a random element, then 2–3s to compute its order and power up to obtain an element of order 2, then 4–5s to test irreducibility of the extended matrix group). This yielded a subgroup $E = \langle S, r\rangle$ of G equal to ${}^2\mathrm{E}_6(2)$ (order 76532479683774853939200). Using the modular Meataxe, a projection $\pi : E \longrightarrow E_\pi$ was then constructed, where E_π was a degree-78 matrix representation over $\mathbb{F}_2$, and S_π was then set to $\pi(S)$. Since the computation of normalizers was still too hard within E_π, we instead successively generated a random order-2 element g_π of E_π and computed the intersection of S_π and $(S_\pi)^{g_\pi}$ until this was reasonably large. After a few random tries (a few seconds per try), this yielded an intersection L_π which had order 454164480 and shape $2^{10}.\mathrm{M}_{22}$. This could then be mapped back under π^{-1} to the original degree-4370 matrix representation over $\mathbb{F}_2$ of G to obtain $L \leq H$ with order 908328960 and shape $2.2^{10}.\mathrm{M}_{22}$ and the corresponding $g \in G \setminus H$ with $L^g = L$ and $g^2 = 1$.

The restriction of the representations of H to L and L^g in ExtensionImageSetup took 480s for degree-788 and 39426s for degree-3588. Constructing the Hom-module basis took 16500s and yielded 54 initial image matrices. Within ExtendRelations, the initial square group relation yielded 54 degree-2 relations (1560s), then an order-3 group relation yielded 42 linear relations and 366 degree-2 relations, reducing the number of variables to 7 (5790s), and finally an order-4 group relation yielded one linear relation and 4 quadratic relations, reducing the number of variables to 6 and giving a dimension-2 ideal (1351s). The final lexicographical Gröbner basis of the ideal was:

$$\{\, x_1 + \frac{3726}{325}x_2x_6, \quad x_2x_5 - \frac{4225}{905418},$$
$$x_3 + \frac{3726}{325}x_4x_5, \quad x_4x_6 - \frac{4225}{83298456} \,\}.$$

The point of the variety was $(\frac{13}{5589}, \frac{130}{5589}, -\frac{13}{972}, \frac{65}{11178}, \frac{65}{324}, \frac{65}{7452})$ and the corresponding image matrix $\rho(g)$ had density 14.7% and denominator LCM $2^8.3^5.7.23$.

Finding words for the standard generators g_1, g_2 of G in terms of the generators of H and the normalizer element g took 3200s; it then took 26924s to compute the images of g_1, g_2 using these words. We were unable to conjugate H so that one of the standard generators of G was in H, so both image matrices are dense. Although $\rho(g_1)$ has density 89.2%, it has only 10950 distinct entries, with absolute maximum numerator 39725 and denominator LCM $2^{11}.3^5.7.23$, while $\rho(g_2)$ has density 89.2%, and only 11251 distinct entries, with absolute maximum numerator 31045 and denominator LCM $2^{11}.3^5.7.23$. The traces of the matrices are -53 and 78 respectively, matching the character table in the ATLAS. A sample of 10 random entries of the matrices is the following:

$$\{\frac{111}{32}, \frac{7}{384}, \frac{581}{24}, \frac{-31}{24}, \frac{-1909}{1344}, \frac{-159}{5888}, \frac{333}{896}, \frac{153}{896}, \frac{437}{24}, \frac{-467}{5184}\}.$$

The total time taken was 35.0 hours, starting from the precomputed representations of H.

We can of course easily construct the mod-p reduction of this representation for any prime p not dividing the denominators (in particular, 5 is of interest). We also used a p-adic conjugation algorithm (outlined on p. 152) to construct corresponding irreducible degree-4371 representations over $\mathbb{F}_p$ for $p = 3, 7, 23$ (6291s, 1560s, 2357s respectively) and verified that the mod-3 representation is equivalent to the one in the online ATLAS [WWT+] by the modular Meataxe.

5.12. The degree-8671 representation of the Fischer Group Fi_{24}'

Let G be the sporadic simple Fischer group Fi_{24}', of order

$$1255205709190661721292800 = 2^{21}.3^{16}.5^2.7^3.11.13.17.23.29.$$

A minimal-degree faithful ordinary representation of G has degree 8671 and can be realized over $\mathbb{Q}$. We constructed such a representation explicitly over $\mathbb{Q}$ by general extension without explicit use of the character (table entry on p. 187).

A degree-306936 permutation representation was used to define G. Let χ be the degree-8671 irreducible character of G and let H be the largest maximal subgroup which is equal to F_{23} (and can be computed as a point stabilizer); $\chi_H = \chi \downarrow_H$ splits as $3588 + 5083$ over $\mathbb{Q}$. Corresponding representations of H had already been computed (see p. 118).

Thankfully, finding a sufficiently large subgroup $L \leq H$ and normalizer $g \in G \setminus H$ required very little computation. We simply took L to be the largest maximal subgroup of H which equals $2.F_{22}$ (index 31671). The normalizer N of L in G was computed in 30s (using the standard backtrack algorithm in permutation groups) and N has order $2 \cdot |L|$ with $N \not\subset H$. Now write $H = \{h_1, h_2\}$, where h_1, h_2 are the standard generators of H. Because computing $\rho_H(h)$ for an arbitrary $h \in H$ would be very expensive, we did a random search for some $g \in N \setminus H$ such that $g^2 \in H$ and $g \cdot h_1$ or $g \cdot h_2$ had a small order. After a minute's search (when the smallest possible order had been stable for quite a while), we had a suitable g of order 2 such that $g \cdot h_2$ had order 8.

Then we applied the general extension algorithm without the explicit use of the character with the above L and g. Computing the restriction of the representations of H to L was very expensive: the restriction of the degree-3588 representation took 23675s, while the restriction of the degree-5083 representation took 47280s. There were 5 initial image matrices, then order relations for orders 2 $[g^2 = 1]$ and 8 $[(gh_2)^8 = 1]$ reduced this to 3 matrices and a corresponding prime ideal of required dimension 1 (32046s).

Finally, computing the corresponding images of the standard generators g_1, g_2 of G took 58072s total. We were unable conjugate H so that one of the standard generators of G was in H, so both image matrices are dense: $\rho(g_1)$ has density 84.9% but only 1407 distinct entries, while the absolute maximum numerator is 2277 and the denominator LCM is $989184 = 2^{11}.3.7.23$. Similarly, $\rho(g_2)$ has density 89.0% but only 2936 distinct entries, while the absolute maximum numerator is 1655 and the denominator LCM is $1978368 = 2^{12}.3.7.23$. A sample of 10 random entries of the matrices is the following:

$$\{\frac{117}{256}, \frac{9}{16}, \frac{819}{368}, \frac{115}{192}, \frac{259}{128}, \frac{3}{5152}, \frac{11}{8}, -\frac{469}{16}, -\frac{21}{2944}, -\frac{651}{1472}\}.$$

The total time taken was 38.6 hours, starting from the precomputed representations of H.

5.13. Representations of the Harada-Norton Group

Let G be the sporadic simple Harada-Norton group, of order

$$273030912000000 = 2^{14}.3^6.5^6.7.11.19.$$

We computed several irreducible representations of G via general extension, as follows.

The degree-1140000 permutation representation was used to define G. Let H be the largest maximal subgroup of G, which equals A_{12} (index 1140000). We used general extension to compute several representations of G via the subgroup H. It was easy first to compute the largest normalized subgroup L of H with $g \in G \setminus H$, such that $L^g = L$; L had order 518400 and g had order 10, with $g^2 \in H$. Also, H was first conjugated so that the first standard generator g_1 of G is in H, so the image of g_1 in each of the following representations is sparse.

The degree-133 representation of G, realized over $\mathbb{Q}(\sqrt{5})$ (table entry on p. 171), was computed via general extension with the above H, L and g. The relevant representations of H had degrees 1 and 132, and these were first constructed in 1.1s. There were 7 initial image matrices, and these were reduced to 6 by linear reduction; then the initial square group relation and one order-4 group relation reduced the system to 2 image matrices and an ideal of the required dimension 1 (15s). Rewriting the representation to be defined on the final generators took 1s, and the total time was 19.8s. This ordinary representation had also been explicitly constructed by Bray & Curtis [BC03].

The degree-760 rational representation of G (table entry on p. 181) was again computed via general extension with the above H, L and g. The relevant representations of H had degrees $1, 132, 165, 462$ and these were computed in 77s using IRREDUCIBLERATIONALREPRESENTATIONS. There were 43 initial image matrices, and these were reduced to 24 by linear reduction; then the initial square group relation and group relations of order 4 and 6 reduced the system to 18 image matrices and an ideal of the required dimension 3 (234s). Rewriting the representation to be defined on the final generators took 48s, and the total time was 359s.

The degree-3344 rational representation of G (table entry on p. 185) was again computed via general extension with the above H, L and g. The relevant representations of H had the following degrees with multiplicities: $1, 54, 132 \times 2, 462 \times 2, 616, 1485$; these were all quickly constructed by IRREDUCIBLERATIONALREPRESENTATIONS (207s), except for the degree-1485 case (2126s: see p. 72 for details). Note the non-trivial multiplicities; this rarely arises when H is the largest maximal subgroup of G, but is still handled successfully by the general extension algorithm. Because of the multiplicities, the norm of $\chi \downarrow_H$ was

12, so an ideal of dimension 11 was required. Initially there were 146 initial image matrices, and these were reduced to 123 by linear reduction. The initial square group relation yielded 130 quadratic relations alone. Then an order-4 group relation gave 673 quadratic relations and 33 linear relations, reducing to 90 variables. Another order-4 group relation then gave 6 linear relations, reducing to 84 variables with 584 quadratic relations. The ideal now had dimension 11, as required (11327s for the complete relation collection). Including linear relations for 11 maximally independent variables reduced the ideal to have linear relations only, so it was then easy to write down a rational image matrix for g. Finally, rewriting the representation to be defined on the standard generators took 228s. The total time taken was 11762s (3.2h).

The degree-8778 and degree-8910 representations of G were extracted from the tensor square of the degree-266 irreducible rational representation by the hybrid algorithm of the next chapter, while the degree-9405 representation was constructed by general extension (see p. 187 for details).

5.14. Conclusion

We summarize the main features of the extension approach. Some of the key advantages are the following:

1. This approach can handle the situation where G has no proper subgroups of moderate index so the splitting approach is not applicable when reasonable tensor products are not available (since one cannot construct permutation or induced representations of reasonable degree), and does not require any specific conditions for G or χ. In fact, if the number of variables in the symbolic matrix can be kept to be small (say under 50), then the general extension algorithm is very efficient; the size of G and the indices of its maximal subgroups become rather irrelevant and the degree of the representation is not a major factor either (the heart of the algorithm involves only matrix multiplication and computations with partial Gröbner bases which can be managed when the number of variables is reasonable and the dimension of the ideal is not too large).
2. Under the inductive assumption that ρ_H has small entries, this approach typically yields a result with very small entries also. This is true even when the final representation is written over an irrational field (in contrast to the splitting approach).
3. A simple variant of the general extension algorithm avoids the explicit use of the character χ, so even when it is not feasible to compute with the classes or character table of G, then one can often still compute a representation affording χ (the other algorithms need the character explicitly). Several of the huge sporadic simple groups can be handled this way, for example.
4. Once a suitable normalized subgroup L is found and the linear reduction is done (which may be skipped in the variant with no use of χ), the only computations needed with the group G are elementary operations on elements to gather suitable relations. Thus non-trivial properties of the structure of G are irrelevant; in particular, the algorithm avoids a search in subgroups (which the splitting approach requires to find a suitable virtual representation). As we have seen, the algorithm is very effective even when the basic group arithmetic is expensive (e.g., for the representation of

the Baby Monster in Sec. 5.11) or it is impossible to compute a base and strong generating set for G. Also, the potentially expensive multiple evaluations of the character can be avoided (as noted in Ex. 5.6.1, it is sometimes better to avoid using the character even when it has already been computed because non-linear relations can reduce the system more quickly).

Some of the limitations are the following:

1. In both irreducible and general extension, a sufficiently large normalized subgroup L may be hard to find. Even when the largest possible normalized subgroup L can be found easily, it may be such that the number of associated image matrices is very large and in the general extension algorithm in particular, the number of variables and the number of terms in the polynomials may grow so large as to make the computation impossible.
2. If $\chi \downarrow_H$ splits into many irreducibles over the field F, it may be expensive to set up a suitable block-diagonal representation ρ_H, and so this may take longer than using the splitting approach directly to compute the representation affording χ. But this is rarely a major problem.

CHAPTER 6

Entry Reduction and the Hybrid Algorithm

6.1. Introduction

Let χ be an absolutely irreducible character for a finite group G. The previous two chapters presented algorithms which start with a representation ρ_H affording $\chi_H = \chi \downarrow_H$ for a subgroup H of G, and extend ρ_H to a representation ρ of G affording χ, such that $\rho \downarrow_H = \rho_H$. Also, if χ_H is absolutely irreducible, then ρ is unique, and we have seen both from the simple bounds implied by Minkwitz's formula (p. 82) and in practical examples that if the entries of the image matrices defining ρ_H are reasonably small, then those defining ρ are also reasonably small. This fact led to the idea that one could reverse the process to reduce the entries of an existing representation. Suppose first that we already have an arbitrary representation $\rho_1 : G \to \mathrm{GL}_n(F)$ affording χ. Now if we construct some representation $\rho_H : H \to \mathrm{GL}_n(F)$ affording $\chi_H = \chi \downarrow_H$, such that ρ_H has small entries, then we can compute a transformation matrix T such that $(\rho_1 \downarrow_H)^T = \rho_H$, and then $\rho = (\rho_1)^T$ affords χ, and is such that $\rho \downarrow_H = \rho_H$. Using this idea, we first present a heuristic LLL-based algorithm to choose a suitable transformation matrix T; this seems to work very well in practice to yield a reduced representation most of the time.

We then introduce a 'hybrid' algorithm which combines the splitting and extension approaches. It first sets up information determining an absolutely irreducible representation ρ_1 using the splitting approach (via condensation of a potentially large-degree virtual representation). Now ρ_1 is not constructed explicitly: often it will have very large entries and would take a very long time to construct. But the algorithm can use the above reduction algorithm with modular techniques to conjugate ρ_1 directly to a reduced representation ρ written over a minimal field and with reduced entries. This algorithm is extremely efficient and routinely allows the construction of representations of very high degree over non-trivial number fields, typically with very small entries. It avoids the need to find a normalized subgroup L in the extension algorithms, so is particularly suitable for the case that ρ_H splits into many irreducible components. Also, H does not need to be maximal for the method to work well.

6.2. The Entry Reduction Algorithm

For the algorithm to reduce the entries of a given representation, we first present a subalgorithm to select a suitable reduced partial basis of a Hom-module. Then our main algorithm to construct a reduced representation is very simple, based on this.

Algorithm REDUCEDHOMBASIS(B, r)

INPUT:

- A basis $B = [b_1, \dots, b_k]$ of a subspace S of $\mathcal{M}_{m \times n}(F)$, where F is a number field.
- An integer r with $1 \leq r \leq k$, such that the sum of the rowspaces of the b_i has rank at least rm over F.

OUTPUT:

- A reduced basis $C = [c_1, \dots, c_r]$ of a subspace of S such that the sum of the rowspaces of the c_i has rank rm over F.

STEPS:

1. Write $F = \mathbb{Q}(\alpha)$, and let $d = \mathrm{Deg}_{\mathbb{Q}}(F)$.
   Let $\phi : \mathcal{M}_{m \times n}(F) \to \mathbb{Q}^{mnd}$ be the natural $\mathbb{Q}$-vector space isomorphism, viewing $\mathcal{M}_{m \times n}(F)$ as a vector space over $\mathbb{Q}$.
2. Set $S_{\mathbb{Q}}$ to the (kd)-dimensional subspace of $\mathbb{Q}^{mnd}$ generated by
   $$\{\phi(b_i \cdot \alpha^j) : 0 \leq j \leq d-1, 1 \leq i \leq k\}.$$
   Set $L := (l_1, \dots, l_{kd})$ to a LLL-reduced basis of the saturation of $S_{\mathbb{Q}}$, sorted with the shortest vectors first.
   Set $W = (w_1, \dots, w_{kd}) = (\phi^{-1}(l_1), \dots, \phi^{-1}(l_{kd}))$.
3. Construct an F-basis $C = [c_1, \dots, c_r]$ from W such that the sum of the rowspaces of the c_i has rank rm over F, as follows:
   (a) First try each subset of W of cardinality k (in lexicographical order).
   (b) Next try k distinct sums of pairs from W.
   (c) Finally, enumerate k linear combinations of elements of R with increasing integral coefficients.

   Return C.

Algorithm ENTRYREDUCTIONBYSUBGROUP(ρ_1, H)

INPUT:

- A representation $\rho_1 : G \to \mathrm{GL}_n(F)$ for a finite group G and a field F which is normal over $\mathbb{Q}$ (where ρ_1 is not necessarily irreducible over F).
- A proper subgroup H of G (not necessarily maximal).

OUTPUT:

- A representation $\rho : G \to \mathrm{GL}_n(F)$ of G which is equivalent to ρ_1, and such that $\rho \downarrow_H$ is equivalent to a block representation of $\chi \downarrow_H$.

STEPS:

1. Set χ to the character of ρ_1 and χ_H to $\chi \downarrow_H$ and decompose χ_H uniquely as
   $$\chi_H = \sum_{i=1}^{k} m_i \cdot \psi_i,$$

```
   where $\psi_i \in \mathrm{Irr}_F(H)$ and $m_i \geq 1$ for $1 \leq i \leq k$.
2. Set $[\sigma_1, \ldots, \sigma_k]$ := IrreducibleRepresentationsOverField($[\psi_1, \ldots, \psi_k], F$).
3. Set $\rho_H := \rho_1 \downarrow_H$.
   For $i := 1$ to $k$ do:
   {
      Set $B_i$ to an $F$-basis of $\mathrm{Hom}_{FH}(\sigma_i, \rho_H)$.
      Set $[C_{i,1}, \ldots, C_{i,m_i}]$ := ReducedHomBasis($B_i, m_i$).
   }
4. Set $T$ to the vertical concatenation of $[C_{1,1}, \ldots, C_{1,m_1}, \ldots, C_{k,1}, \ldots, C_{k,m_k}]$.
   Set $\rho := (\rho_1)^T$ and return $\rho$.
```

Proposition 6.2.1. *Algorithms* ReducedHomBasis *and* EntryReductionBySubgroup *are correct.*

Proof. Clearly the σ_i representations are set up so that $\rho_H = \oplus_{i=1}^{k} \oplus_{j=1}^{m_i} \sigma_i$ affords χ_H. Subalgorithm ReducedHomBasis is very similar to ReducedBasisForAction (p. 73); the only difference is that the former selects a final basis of matrices so that their images add up to a subspace of the right rank; since the input matrices form a basis for the Hom-module, the search in Step 3 of that subalgorithm must find such a basis. Now as the ψ_i are inequivalent and the i-th call to ReducedHomBasis from EntryReductionBySubgroup returns a basis of homomorphisms whose images are pairwise independent, T must be invertible and clearly conjugating ρ_1 by T yields ρ such that $\rho \downarrow_H = \rho_H$. □

Remarks 6.2.2. We note the following points on the algorithm and its implementation:

1. The point of using LLL-reduction in ReducedHomBasis is that of all the transformation matrices T which can be chosen so that the output representation ρ is such that $\rho \downarrow_H = \rho_H$, using one derived from the LLL-reduction of the Hom-bases seems to give a highly reduced result most of the time, particularly when the field F has small degree. The key feature of this algorithm which makes it so effective is that the dimension of the lattice that the LLL algorithm acts on in the subalgorithm ReducedHomBasis is typically very much smaller than the degree n of the representation. If the field F has degree f, and the dimension of the endomorphism ring of σ_i is d, then the dimension of the lattice will be df. Note that for each i, the length of B_i (the dimension of the Hom-module for the i-th representation of H) may be greater than m_i. But it is very important to give the whole basis of the Hom-module to ReducedHomBasis so that the LLL algorithm has a larger lattice to act upon and can thus produce a more reduced basis.
2. The quality of the output can vary considerably by varying the subgroup H (even if the representations of H have very small entries themselves). Generally speaking, assuming the representations of H have small entries, then the larger the subgroup H is, the more likely it is that the entries of the final representation are smaller. Also, when non-trivial multiplicities are present, this increases the dimension of the lattice which the LLL algorithm has to act on in ReducedHomBasis. So it is usually best

to let H be one of the largest maximal subgroups of G for H, so that there are less constituents of $\chi \downarrow_H$, and usually these only occur multiplicity 1. But this not always the best choice; several examples below will demonstrate this phenomenon.

3. One does not always have to call IrreducibleRepresentationsOverField to construct the initial representations of H: one can instead use any other method to compute these, as long as they are realized over F. This is done in some of the examples below. But IrreducibleRepresentationsOverField does automatically give appropriate irreducible F-representations even when a non-trivial Schur index is present (either for χ or one of the characters of H). Also, the normality condition on F has only been imposed so that IrreducibleRepresentationsOverField can be called for a complete automatic algorithm to construct the representations of H.
4. It is easy to adapt the algorithm RewriteOverMinimalField (p. 80) to use the above algorithm instead of SplitByEigenspace to rewrite a representation over any field to be over a minimal field, and also with reduced entries.

Example 6.2.3. Let $G = 6.\mathrm{A}_7$. Consider the following absolutely irreducible representation $\rho_1 : G \to \mathrm{GL}_6(F)$, where $F = \mathbb{Q}(\alpha)$, $\alpha = \zeta_3$ (a primitive cube root of unity).

$$\rho_1(g_1) = \frac{1}{798}\begin{pmatrix} -1262\alpha-1546 & 787\alpha-145 & -1038\alpha+246 & -1227\alpha+1926 & 1161\alpha+396 & -101\alpha-1948 \\ 172\alpha+236 & 379\alpha-445 & 942\alpha+810 & -381\alpha+600 & -537\alpha-876 & -365\alpha-640 \\ 1828\alpha+1970 & -593\alpha-127 & 918\alpha+276 & 15\alpha-2160 & -1179\alpha-198 & 649\alpha+1772 \\ 382\alpha+320 & 568\alpha+269 & 60\alpha+138 & -591\alpha+117 & 9\alpha-99 & 391\alpha+221 \\ 1200\alpha+366 & 111\alpha-423 & 318\alpha-306 & -1257\alpha-936 & -471\alpha+792 & 729\alpha+360 \\ 1534\alpha+1214 & -299\alpha+629 & 78\alpha-858 & -489\alpha-2202 & -507\alpha+390 & 1027\alpha+2402 \end{pmatrix},$$

$$\rho_1(g_2) = \frac{1}{798}\begin{pmatrix} -1414\alpha-539 & -75\alpha+2022 & 993\alpha+249 & 1604\alpha-3014 & -123\alpha-2637 & -1357\alpha+829 \\ -322\alpha-581 & 45\alpha+702 & -117\alpha-1107 & -58\alpha-692 & -405\alpha-333 & -37\alpha+673 \\ 1106\alpha-329 & -561\alpha-2208 & -297\alpha+75 & -802\alpha+3502 & 1059\alpha+2715 & 479\alpha-1811 \\ 287\alpha-98 & -81\alpha+93 & 450\alpha+237 & 530\alpha+820 & 330\alpha-39 & -226\alpha-41 \\ -168\alpha-945 & -549\alpha-744 & -9\alpha+99 & 1080\alpha+1686 & 951\alpha+711 & -453\alpha-1401 \\ 812\alpha-287 & -1023\alpha-1914 & -213\alpha+747 & 290\alpha+3460 & 1227\alpha+1665 & 185\alpha-1769 \end{pmatrix}.$$

We can reduce the entries of ρ_1 by algorithm EntryReductionBySubgroup as follows. Let $H = \langle h_1, h_2 \rangle$ be one of the maximal subgroups of G of order 180 (index 6). Then $\chi \downarrow_H$ splits as 1+5 over F, where χ is the character of ρ_1. We can instantly construct corresponding representations σ_1, σ_2 as follows:

$$\sigma_1(h_1) = \begin{pmatrix} 1 \end{pmatrix}, \quad \sigma_1(h_2) = \begin{pmatrix} -\alpha - 1 \end{pmatrix},$$

$$\sigma_2(h_1) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad \sigma_2(h_2) = \begin{pmatrix} 0 & -\alpha-1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The echelonized basis of $\mathrm{Hom}_{FH}(\sigma_1, \rho_1 \downarrow_H)$ contains the single 1×6 matrix:

$$\begin{pmatrix} 1 & 0 & 0 & -2\alpha-1 & 1 & -\alpha \end{pmatrix}$$

and ReducedHomBasis applied to this returns the basis containing the single 1×6 matrix:

$$C_{1,1} = \begin{pmatrix} \alpha+1 & 0 & 0 & -\alpha+1 & \alpha+1 & 1 \end{pmatrix}$$

Similarly, the echelonized basis of $\mathrm{Hom}_{FH}(\sigma_2, \rho_1 \downarrow_H)$ contains the single 5×6 matrix:

$$\frac{1}{13}\begin{pmatrix} 13 & 12\alpha+9 & -4\alpha+10 & -10\alpha-14 & -2\alpha-8 & 8\alpha+6 \\ 8\alpha+6 & 8\alpha-7 & 6\alpha-2 & -6\alpha+2 & -17\alpha-3 & 7\alpha+2 \\ -4\alpha-3 & 0 & 6\alpha-2 & 4\alpha+16 & 3\alpha-1 & -15\alpha-8 \\ 4\alpha+3 & -9\alpha+3 & 6\alpha-2 & -2\alpha-8 & -6\alpha-11 & -\alpha+9 \\ -\alpha-4 & 3\alpha-1 & -2\alpha-8 & -2\alpha-8 & 3\alpha-1 & \alpha+4 \end{pmatrix}$$

and ReducedHomBasis applied to this returns the basis containing the single 5×6 matrix:

$$C_{2,1} = \begin{pmatrix} \alpha-3 & -3\alpha-3 & 2\alpha-2 & 2\alpha+4 & 2 & -2\alpha-2 \\ -2\alpha-2 & -3\alpha+1 & -2\alpha & 2\alpha & 5\alpha+2 & -2\alpha-1 \\ \alpha+1 & 0 & -2\alpha & -4 & -\alpha & 4\alpha+3 \\ -\alpha-1 & 3\alpha & -2\alpha & 2 & \alpha+3 & \alpha-2 \\ 1 & -\alpha & 2 & 2 & -\alpha & -1 \end{pmatrix}.$$

After setting T to the vertical concatenation of $C_{1,1}$ and $C_{2,1}$ and then setting ρ to $(\rho_1)^T$, we obtain the following reduced representation:

$$\rho(g_1) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha \\ 0 & 0 & 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\alpha-1 & 0 & 0 & 0 \\ 0 & -\alpha-1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\rho(g_2) = \begin{pmatrix} 0 & -\frac{1}{4}\alpha & \frac{1}{4}(\alpha+1) & -\frac{1}{4} & 0 & -\frac{1}{4}\alpha \\ \alpha+1 & \frac{1}{2}(-\alpha-1) & \frac{1}{2}(-\alpha-1) & \frac{1}{2}(-\alpha-1) & 0 & 0 \\ -\alpha & \frac{1}{2} & \frac{1}{2}(-\alpha-1) & 0 & 0 & \frac{1}{2}(-\alpha-1) \\ -1 & 0 & \frac{1}{2}(-\alpha-1) & \frac{1}{2}\alpha & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \alpha+1 & \frac{1}{2} & 0 & \frac{1}{2}\alpha & 0 & \frac{1}{2}\alpha \end{pmatrix}$$

Note that since H had first been conjugated so that $g_1 \in H$, the first image matrix $\rho(g_1)$ is monomial (a block diagonal sum of images of σ_1 and σ_2).

For this particular example, if the subgroup H is varied, then the algorithm still returns a similar representation with very small entries, even when H is a much smaller subgroup (so that the corresponding representations of H may have non-trivial multiplicities). Also, if the original ρ_1 is first conjugated to have much larger entries, then applying the algorithm to such a ρ_1 produces a result which is essentially the same as the above reduced ρ.

Despite this being a small example, it scales very well as the degree of the representation increases, since the quality of the output tends to depend on the number of blocks in the representation of H, and not on the degree.

Example 6.2.4. Let G be the sporadic simple group J_3. A minimal-degree faithful representation of G has degree 85 and can be realized over the quadratic field $F = \mathbb{Q}(\sqrt{-19})$. Let χ be one of the corresponding characters. In Ex. 3.9.7 (p. 77) we constructed a representation $\rho_1 : G \longrightarrow \mathrm{GL}_{85}(F)$ affording χ in 285s, but the representation was poor, with entry numerators of about 73 digits and common denominator 1. However, we could reduce this representation by algorithm EntryReductionBySubgroup, as follows. Let H be the largest maximal subgroup of G, which has index 6156 (shape $L_2(16).2$). Then $\chi_H = \chi \downarrow_H$

is rational and splits over F as $17 + 68$; corresponding irreducible rational representations $[\sigma_1, \sigma_2]$ were constructed by algorithm IRREDUCIBLEREPRESENTATIONSOVERFIELD in only 0.7s. Then the computation of the Hom-modules and the final conjugation took 22.3s to obtain an equivalent reduced representation $\rho : G \to \mathrm{GL}_{85}(F)$.

Let g_1, g_2 be the standard generators of G. Since $g_1 \in H$, $\rho(g_1)$ is rational with the block form given by σ_1, σ_2 (integral with maximum entry 3), while $\rho(g_2)$ is dense and has entries in F with at most 3-digit numerators and denominator LCM 120. Random sample entries of $\rho(g_2)$ are: $\frac{1}{20}(\alpha+10)$, $\frac{1}{20}(4\alpha+5)$, $\frac{1}{120}(11\alpha-4)$. This is possibly the first time that an absolutely irreducible representation affording χ has been constructed over a minimal field, with very small entries. The algorithm of the next section will allow the same reduced representation to be computed much more quickly (see Ex. 6.4.1).

6.3. The Hybrid Black-box/Entry Reduction Algorithm

6.3.1. Introduction. In this section we present the hybrid algorithm for computing an irreducible representation which combines the splitting and extension approaches. The implementation of this algorithm is very efficient and routinely allows the construction of representations of very high degree over number fields, typically with very small entries. It is preferable over the general extension algorithm when there is a large number of image matrices needed by the latter algorithm.

The basic tool used by the algorithm is a 'black-box representation' which encapsulates a fixed uniquely-determined underlying representation $\rho_F : G \to \mathrm{GL}_n(F)$, where F is $\mathbb{Q}$ or a number field. The key idea is that the explicit construction of ρ_F itself is avoided: often it will have very large entries and would take a very long time to construct. Yet one can efficiently compute the modular projection of ρ_F under any given modular projection function $\phi : F \dashrightarrow \mathbb{F}_p$. This feature will be combined with the entry reduction algorithm (via modular techniques) to conjugate ρ_F directly to a reduced representation ρ.

Definition 6.3.1. *Call $\mathscr{B} = (G, \chi, F, \pi(\phi))$, a* **black-box representation** *for χ, if:*

- *G is a finite group and χ is a character of G (not necessarily irreducible).*
- *F is $\mathbb{Q}$ or a number field $\mathbb{Q}(\alpha)$ which is normal over $\mathbb{Q}$ and contains a subfield isomorphic to $\mathbb{Q}(\chi)$.*
- *There is a fixed underlying representation $\rho_F : G \to \mathrm{GL}_n(F)$ (usually not explicitly constructed) which affords χ, under a suitable embedding of $\mathbb{Q}(\chi)$ into F.*
- *π is a 'modular projection' function which takes a coefficient modular reduction function $\phi : F \dashrightarrow \mathbb{F}_p$ and returns the representation $\rho_\phi : G \to \mathrm{GL}_n(\mathbb{F}_p)$ given by the reduction of ρ_F under ϕ. (The function can return some error flag if the modular reduction cannot be performed on all entries defining ρ_F.)*

The way we will use black-box representations practice is the following:

- A 'black-box setup' function will take G and χ and automatically construct a black-box representation $\mathscr{B} = (G, \chi, F, \pi(\phi))$.
- Once we have such a $\mathscr{B}$, we can extract F and then successively call the modular function $\mathscr{B}.\pi$ with suitable modular reduction functions $\phi : F \dashrightarrow \mathbb{F}_p$ and use the usual modular combination techniques to construct an ordinary representation which is equivalent to the underlying ρ_F.

Note that we require the field F to be normal over $\mathbb{Q}$ since it allows IrreducibleRepresentationsOverField to be used to set up suitable representations of the subgroup H and it will enable a suitable underlying representation to be set up easily when we use condensation (see below). So the following algorithms will return a flag `'Fail'` if a normal field F cannot be found; this has not been a serious restriction in practice for all our applications, since we have always found a normal field easily (it is trivial to find if the Schur index is 1).

6.3.2. Using an Irreducible Rational Representation. One method to set up a black-box representation simply uses the eigenspace of a suitable endomorphism, just as in algorithm AbsolutelyIrreducibleRepresentation.

```
Algorithm BBRationalModuleSetup($\chi, M_{\mathbb{Q}}$)
Input:
  • An absolutely irreducible character $\chi$ for a finite group $G$.
  • An irreducible $\mathbb{Q}G$ module $M_{\mathbb{Q}}$ whose character contains $\chi$.
Output:
  • A black-box representation $\mathscr{B} = (G, \chi, F, \pi(\phi))$ for $\chi$, where $F$ is a minimal field for
    $\chi$.
Steps:
  1. Set $E := \mathrm{End}_{\mathbb{Q}G}(M_{\mathbb{Q}})$.
     Search for a generator $e$ of a maximal subfield of $E$ such that the subfield is normal
     over $\mathbb{Q}$ (for 1000 tries of small random elements of $E$, say). Return 'Fail' if a normal
     field cannot be found.
     Set $U_F$ to a basis matrix of the $\alpha$-eigenspace of $e$ over $F$.
  2. Set $\pi :=$
     Function($\phi$)
     {
        [$\phi : F \dashrightarrow F_p$ is a given partial homomorphism, naturally extended to vectors,
           matrices, modules over $F$, etc.]
        Set $\bar{U} := \phi(U_F)$, $\bar{M} := \phi(M_{\mathbb{Q}})$.
        Set $\bar{S}$ to the submodule of $\bar{M}$ generated by the rows of $\bar{U}$, computing the
           reduced action on a fully echelonized basis [all done over $F_p$].
        [Return 'Fail' if $\phi$ applied to any element is not in the domain of $\phi$.]
        Return $\bar{S}$.
     }
  3. Set $\mathscr{B} := (G, \chi, F, \pi(\phi))$ and return $\mathscr{B}$.
```

Lemma 6.3.2. *Algorithm* BBRationalModuleSetup *is correct.*

Proof. This algorithm is similar to AbsolutelyIrreducibleRepresentation except that the reduced module acting on the α-eigenspace of e is not computed over F but is dynamically computed mod p each time that π is called; since an echelonized basis of the subspace is used, the modular reduced actions are consistent, thus determining a fixed underlying representation ρ_F over F, and correctness follows by Cor. 1.5.5. □

6.3.3. Using Condensation. We now present an advanced method to set up a black-box representation for an absolutely irreducible character χ, using the condensation-based tools of Chapter 3. The basic idea is to use most of the algorithm IRREDUCIBLERATIONALREPRESENTATIONS to determine an underlying irreducible rational representation containing χ and to set up relevant condensation information, without explicitly constructing the final rational representation.

Recall that the condensation operation simply maps the FG-module M to the $eFGe$-module eMe, where e is the idempotent e_K for some subgroup K (see Sec. 3.5). Now an elementary but useful property of this operation is that it commutes with extension of the base field, since it just involves multiplication by an algebra element. Thus if a simple module $\tilde{S}$ over $\mathbb{Q}$ is the condensation of some simple module S, then we can decompose $\tilde{S}$ over an extension field F, and then each component over F must correspond to a submodule of S over F. The following result shows that a suitable field can also be found via the endomorphism ring of the condensed module.

Proposition 6.3.3. *Suppose M is a simple FG-module and let $e = e_K$ be the condensation idempotent for some subgroup K of G. Let $\tilde{M} = Me$ and assume $\tilde{M} \neq 0$, so $\tilde{M}$ is simple by Lem. 3.5.1. Let $E = \mathrm{End}_{FG}(M)$ and let $\tilde{E} = \mathrm{End}_{eFGe}(\tilde{M})$. Then $E \cong \tilde{E}$ as rings.*

Proof. Let $A = FG$. Then A has a unique simple component A_M such that M is isomorphic to the only simple A_M-module [Jac89, 4.4, 5.4]. Then since $\tilde{M} \neq 0$, A_M and $\tilde{A}_M = eA_Me$ are Morita equivalent, so $E \cong \tilde{E}$ by [Lux97, 3.1.2]. □

To apply these ideas in practice, we first make easy modifications to the algorithms IRREDUCIBLERATIONALREPRESENTATIONS (p. 66) and AUTOMATICCONDENSATION (p. 61): an extra flag is added to each algorithm which indicates 'black-box mode'.

In the black-box mode, IRREDUCIBLERATIONALREPRESENTATIONS first proceeds exactly as before, setting up the queue of possible virtual representations and selecting the first one which contains the desired constituent (and recursing as usual to construct a relevant representation of a subgroup H if induction is to be used). But when the algorithm calls AUTOMATICCONDENSATION to obtain a representation affording χ, it also passes the 'black-box mode' flag; that algorithm also first proceeds as before (to find a suitable subgroup K, etc.), except that after it has extracted the simple submodule $\tilde{S}$ of the condensed module $\tilde{M}$ via the rational Meataxe, it does one modular spin and then the character test to check that the condensed algebra has enough generators; when that test passes, it immediately returns the condensation information $\langle \tilde{M}, \tilde{S}, \mathscr{C} \rangle$, instead of calling INTEGRALSPIN to construct the rational representation. Then IRREDUCIBLERATIONALREPRESENTATIONS also immediately returns the information $\langle \tilde{M}, \tilde{S}, \mathscr{C} \rangle$.

It is then easy to set up a black-box representation based on the information returned by IRREDUCIBLERATIONALREPRESENTATIONS in black-box mode, as the following algorithm does.

```
Algorithm BBCondensationSetup($\chi$)
Input:
  • An absolutely irreducible character $\chi$ for a finite group $G$.
Output:
  • A black-box representation $\mathscr{B} = (G, \chi, F, \pi(\phi))$ for $\chi$ where $F$ is a minimal field for
    $\chi$.
Steps:
  1. Let $\chi_\mathbb{Q} \in \mathrm{Irr}_\mathbb{Q}(G)$ be the irreducible rational character containing $\chi$.
     Call IrreducibleRationalRepresentations($[\chi_\mathbb{Q}]$) in black-box mode to obtain
     the information $\langle \tilde{M}, \tilde{S}, \mathscr{C} \rangle$ for $\chi_\mathbb{Q}$.
     [$\tilde{M}$ is the full condensed module, $\tilde{S}$ is the submodule of $\tilde{M}$ corresponding to $\chi$ and $\mathscr{C}$
     is the associated condensation environment.]
  2. Set $\tilde{E} := \mathrm{End}_{\tilde{A}}(\tilde{S})$, where $\tilde{A}$ is the condensed algebra such that $\tilde{S}$ is an $\tilde{A}$-module.
  3. Set $c := \mathrm{Deg}_\mathbb{Q}(\mathbb{Q}(\chi))$ and set $s := s_\mathbb{Q}(\chi)$.
     Search for a generator $e$ of a maximal subfield of $\tilde{E}$ whose degree over $\mathbb{Q}$ is $d = cs$ and
     is such that the subfield is normal over $\mathbb{Q}$ (for 1000 tries of small random elements of
     $\tilde{E}$, say). Return 'Fail' if a normal field cannot be found.
     Let $f$ be the minimal polynomial of $e$ over $\mathbb{Q}$ and set $F := \mathbb{Q}(\alpha)$ where the minimal
     polynomial of $\alpha$ is $f$.
  4. Set $B_{\tilde{S}}$ to a basis matrix of the $\alpha$-eigenspace of $e$ over $F$.
     Set $B_{\tilde{M}}$ to the matrix corresponding to $B_{\tilde{S}}$ under the embedding of $\tilde{S}$ into $\tilde{M}$.
     Set $U_F := \mathscr{C}$.Uncondense($B_{\tilde{M}}$).
  5. Set $\pi :=$
     Function($\phi$)
     {
         [$\phi : F \dashrightarrow F_p$ is a given partial homomorphism, naturally extended to vectors,
             matrices, modules over $F$, etc.]
         Set $\bar{U} := \phi(U_F)$.
         Set $\bar{S}$ to the submodule of the full virtual module of $\mathscr{C}$
             generated by $\bar{U}$, using $\mathscr{C}$.Action [all done over $F_p$].
         [Return 'Fail' if $\phi$ applied to any element is not in the domain of $\phi$.]
         Return $\bar{S}$.
     }
  6. Set $\mathscr{B} := (G, \chi, F, \pi(\phi))$ and return $\mathscr{B}$.
```

Theorem 6.3.4. *Algorithm* BBCondensationSetup *is correct.*

Proof. In Step 1, after calling IrreducibleRationalRepresentations in black-box mode, we have an $\tilde{A}$-module $\tilde{S}$ which is the condensation of an underlying simple $\mathbb{Q}G$-module S whose character is $\chi_\mathbb{Q}$. Let $E = \mathrm{End}_{\mathbb{Q}G}(S)$ and $\tilde{E} = \mathrm{End}_{e\mathbb{Q}Ge}(\tilde{S})$. By Prop. 6.3.3,

E and $\tilde{E}$ are isomorphic, so we may identify them. In Step 3, a maximal subfield of the desired degree must exist by Thm. 1.5.1 and Thm. 1.4.3. Assume that a maximal subfield of $\tilde{E}$ which is normal over $\mathbb{Q}$ is found (otherwise `'Fail'` will be returned) and let $F = \mathbb{Q}(\alpha)$ be the isomorphic number field, of degree d.

In Step 4, let $\tilde{V}_F$ be the submodule of $\tilde{S}^F$ which is generated by the α-eigenspace of e over F. Then by Lem. 3.5.1, a corresponding submodule V_F of S^F must exist whose condensation is $\tilde{V}_F$ and is just the uncondensation of $\tilde{V}_F$ (working over F).

Since $E \cong \tilde{E}$, F is isomorphic to a maximal subfield of E and so by Cor. 1.5.5 and the fact that F is normal over $\mathbb{Q}$, the $\mathbb{Q}$-representation corresponding to S splits over F into d absolutely irreducible representations, whose characters are the $(F/\mathbb{Q})$-conjugates of χ. By considering the symmetry and Lem. 3.5.1, the corresponding d simple constituents of S^F must all condense to d simple non-zero submodules of $\tilde{S}^F$ with the same dimension. Now by Lem. 1.5.3, the dimension of $\tilde{V}_F$ equals the dimension of $\tilde{S}$ divided by d, so it must be isomorphic to one of these simple submodules of $\tilde{S}^F$. Thus V_F has character χ, under some embedding of $\mathbb{Q}(\chi)$ into F (and ρ_F in the definition of the black-box representation is the representation corresponding to V_F). Otherwise the algorithm is the same as BBRationalModuleSetup above, so the correctness follows in the same way. □

Remarks 6.3.5. We note the following points on our implementation:

1. Algorithm BBCondensationSetup is the more powerful of the setup algorithms since it uses condensation and avoids the explicit construction of an irreducible rational representation, but Algorithm BBRationalModuleSetup is useful when one already has an irreducible rational representation (constructed by whatever means); we have applied it in some situations below.
2. In both algorithms (just as in AbsolutelyIrreducibleRepresentation), if the Schur index s of χ is 1, then the field F is essentially unique (and normal over $\mathbb{Q}$) but if $s > 1$, then F is not unique. As an option, one can specify a particular field F to be used, assuming it can easily be found in $\tilde{E}$.
3. One could also create a black-box setup function for any other algorithm which constructs a representation (e.g., restriction from a known representation with a fixed eigenspace of some endomorphism).

6.3.4. The Hybrid Black-box/Entry Reduction Algorithm. We can now present the hybrid algorithm for constructing a representation affording an absolutely irreducible character. The basic idea is to construct a black-box representation $\mathscr{B}$, and to reduce this by the above EntryReductionBySubgroup algorithm in place, so that the final representation ρ is equivalent to the underlying representation ρ_F of $\mathscr{B}$, but ρ also has reduced entries, and modular techniques are used to avoid the explicit construction of ρ_F at any point.

Algorithm BBReductionRepresentation(χ, H)
Input:

- An absolutely irreducible character χ for a finite group G.

- A proper subgroup H of G (not necessarily maximal).

OUTPUT:

- An absolutely irreducible representation $\rho : G \to \mathrm{GL}_n(F)$ affording χ, where F is a minimal field for χ.

STEPS:

1. Set $\mathscr{B} := \text{BBCONDENSATIONSETUP}(\chi)$; if `'Fail'` is returned, then return `'Fail'`. Write $\mathscr{B} = (G, \chi, F, \pi(\phi))$ and then write $F = \mathbb{Q}(\alpha)$ with $f \in \mathbb{Q}[x]$ the minimal polynomial of α and set $d := \mathrm{Deg}(f)$.

   [One can set up $\mathscr{B}$ and F by other means; the condensation-based method is given here as the default.]

2. Set $\chi_H := \chi \downarrow_H$ and decompose χ_H uniquely as
$$\chi_H = \sum_{i=1}^{k} m_i \cdot \psi_i,$$
   where $\psi_i \in \mathrm{Irr}_F(H)$ and $m_i \geq 1$ for $1 \leq i \leq k$.

3. Set $[\sigma_1, \ldots, \sigma_k] := \text{IRREDUCIBLEREPRESENTATIONSOVERFIELD}([\psi_1, \ldots, \psi_k], F)$.

4. Construct an echelonized F-basis B_i of $\mathrm{Hom}_{FH}(\sigma_i, \rho_F \downarrow_H)$ for $i = 1, \ldots, k$ (where ρ_F is the underlying representation of $\mathscr{B}$ over F) by the standard CRT-based modular scheme (see p. 23), choosing each prime to be greater than $2\chi(1)$, as follows. For each successive prime p and root $\beta_j \in \mathbb{F}_p$ of f (for $1 \leq j \leq d$; take $\beta_1 = 1$ if $F = \mathbb{Q}$):
   - Set $\phi_{\beta_j} : F \dashrightarrow \mathbb{F}_p$ to the partial homomorphism given by reduction modulo p and by $\alpha \mapsto \beta_j$.
   - Set $\rho_{\beta_j} := \mathscr{B}.\pi(\phi_{\beta_j})$ (the modular projection of the underlying representation ρ_F); skip to a new prime if there is failure in the modular construction.
   - Compute an echelonized basis of $\mathrm{Hom}_{\mathbb{F}_p H}(\phi_{\beta_j}(\sigma_i), \rho_{\beta_j} \downarrow_H)$ for $i = 1, \ldots, k$.

   Combine the modular bases via interpolation and Chinese Remaindering and use rational reconstruction to construct each F-basis B_i when stable.

5. For $1 \leq i \leq k$, set $[C_{i,1}, \ldots, C_{i,m_i}] := \text{REDUCEDHOMBASIS}(B_i, m_i)$.
   Set T to the vertical concatenation of $[C_{1,1}, \ldots, C_{1,m_1}, \ldots, C_{k,1}, \ldots, C_{k,m_k}]$.

6. Construct the images $[X_1, \ldots, X_r] \in \mathcal{M}_n(F)$ of the generators $\{g_1, \ldots, g_r\}$ of G under the conjugated F-representation $(\rho_F)^T$, again by the standard CRT-based modular scheme, as follows. For each successive prime p and root $\beta_j \in \mathbb{F}_p$, compute ϕ_{β_j} and ρ_{β_j} as above, set $T_{\beta_j} := \phi_{\beta_j}(T)$ and compute $(\rho_{\beta_j})^{T_{\beta_j}}$; combine these modular images via interpolation and Chinese Remaindering and use rational reconstruction when stable to construct matrices $[X_1, \ldots, X_r] \in \mathcal{M}_n(F)$.

7. Test whether $[X_1, \ldots, X_r]$ define a valid representation ρ of G, where $\rho(g_i) = X_i$ for $1 \leq i \leq r$, by computing a presentation of G and checking that all the relations on the g_i are satisfied by the X_i also. If the validation fails, return to Step 4, using new primes and ensuring that more primes are used than last time.

8. Embed $\mathbb{Q}(\chi)$ in F via Lem. 1.5.4 so that the character of ρ equals χ, then return ρ.

Theorem 6.3.6. *Algorithm* BBREDUCTIONREPRESENTATION *is correct.*

Proof. First, the correctness of the output follows from the verification in Step 7 that ρ is a valid representation of G and the fact that ρ equals the underlying representation ρ_F of $\mathscr{B}$ modulo some $M > 2\chi(1)$ under an appropriate embedding of $\mathbb{Q}(\chi)$ into F, so Lem. 1.5.4 is applicable in the last step, so the output is correct upon termination. To see termination, it is clear that after setting up the black-box representation $\mathscr{B}$ of χ and setting up representations of H affording the irreducible components of $\chi \downarrow_H$ (via IRREDUCIBLEREPRESENTATIONSOVERFIELD), the heart of the algorithm is essentially the same as the algorithm ENTRYREDUCTIONBYSUBGROUP, except that the construction of the B_i bases of the Hom-modules and the conjugation of ρ_F by T are performed by the standard CRT-based modular method. As with other modular algorithms, there can only be a finite number of bad primes (dividing the denominators of entries in ρ_F or the σ_i or entries in the echelonized basis of the Hom-modules), and these will be detected if the construction of ρ_{β_j} fails at any point or the verification fails in Step 7. Similarly, if a modular construction succeeds but does not use enough primes, then this will be detected in the verification step and more primes will be used next time. Thus there must eventually be enough good primes chosen so that all of the Hom-modules are properly constructed and a transformation matrix T is constructed so that the modular reconstruction of $(\rho_F)^T$ succeeds and is correct. □

Remarks 6.3.7. We will give several examples in the next section which demonstrate how the algorithm performs in different ways. We first note the following points on the algorithm and its implementation:

1. The primes can be chosen as in the modular algorithm for computing a Hom-module (see p. 24) and each prime will be much greater than $2\chi(1)$ in practice. It is also good to add further restrictions on each prime p so that the defining polynomials of the fields corresponding to the absolutely irreducible representations of H underlying the σ_i representations do not split modulo p. Then when computing $\mathrm{Hom}_{F_pH}(\sigma_i, \ldots)$, the representation σ_i remains irreducible modulo p, so the modular Hom-module can be computed by the faster Holt/Rees Hom algorithm for an irreducible module to construct the modular homomorphisms (instead of having to decompose a semisimple module).
2. Computing the restricted representation $\rho_{\beta_j} \downarrow_H$ for each modular representation ρ_{β_j} can be rather expensive when the degree is large (since it typically involves evaluation of words in the strong generators). But there is a simple trick to avoid this which we can often use: when we set up the black-box representation for G, we can extend the generators of G to include the generators of H and so when the modular spin algorithm constructs the modular representation of G each time, the reduced action of the generators of H are also constructed, so the restriction of this representation to H is then trivial to compute. The extra cost of computing the reduced action of the extra generators is typically very small. This simple modification helps enormously in very large degree. Note also that the ρ_{β_j} computed in Step 4 can be stored and reused in Step 6, for any prime which is used in both steps.
3. An obvious optimization of Step 4 is that when the basis B_i of the Hom-module for a particular representation σ_i of H has been constructed, then there is nothing to do for

this basis for any subsequent primes which are needed to construct other bases. This situation arises commonly in practice.

4. Note that χ does not really need to be absolutely irreducible; we have given a version here where $\mathscr{B}$ is first constructed for an absolutely irreducible χ, but a black-box representation for an arbitrary character of G could be set up and used. For example, we have sometimes recursively computed a representation affording a character χ_H of a subgroup H which is not absolutely irreducible: to obtain a reduced representation we have extended Step 3 of IrreducibleRepresentationsOverField (p. 77) so that BBRationalModuleSetup can be called with a homogeneous (reducible) rational representation with the suitable endomorphism e and then the rest of BBReductionRepresentation is used to obtain a reduced representation over a field F which is non-minimal for χ_H.
5. For the verification in Step 7, we use the usual technique of using words in the strong generators of G, instead of the original generators. All powers of the generating matrices (and their inverses) can be stored as they arise, to avoid later recomputation. Assume that all relations are of the form $w_l(g_1, \ldots, g_r) = 1$. Now one can write a word $w(g_1, \ldots, g_r)$ as a list of the form $[i_1^{e_1}, \ldots, i_n^{e_n}]$ where $i_j^{e_j}$ corresponds to $g_{i_j}^{e_j}$ (note that e_j can be negative) and then sort the words in lexicographical order according to these lists (comparing bases then exponents). Then while looping over the words in this order, subproducts can be remembered so that each new product can be computed from the point at which the word differs from the previous word, etc. (so this is akin to a depth-first search because of the lexicographical order). Also, every matrix multiplication within each word can be done using a relevant modular algorithm for matrices over F. Alternatively, if a word involves multiplying the matrices $[A_{i_1}, \ldots A_{i_k}]$, then one can determine a bound for the whole product (after having taken out the denominator LCM) and then compute the whole product modulo enough primes and check that this product equals the identity matrix each time (thus avoiding a CRT step at the end). This method can be improved further: since there is already a modulus M such that the putative representation is already known to be correct modulo M, the verification only needs to check each relation modulo enough extra primes which cover the relevant bound. Combining all the above improvements, the verification is very efficient, and typically takes a small number of seconds even for representations with degrees in the hundreds, since matrix multiplication is very fast in our implementation.
6. All the remarks on EntryReductionBySubgroup hold here. In particular, the choice of H can have a strong effect on the quality of the result (see p. 127). It is generally best to choose H to be one of the largest maximal subgroups of G, so that the multiplicities are more likely to be 1 and so the relevant Hom-modules will have small dimension and the LLL-reduction will be stronger. Occasionally there are still non-trivial multiplicities when H is the largest maximal subgroup; worst-case examples are the degree-1920 and -1938 representations of J_3 (p. 184), where the multiplicities go up to 8 and the algorithm fails to construct representations with very small entries (in the former case, the dimension of the last Hom-module is 24, since F has degree 3). But it is not always bad if the representations of H occur with a high multiplicity. For example, for the degree-1728 and -2048 representations of ${}^2\mathrm{F}_4(2)'$, both computed by BBReductionRepresentation (p. 183 and p. 184 respectively), the representations

of the subgroup H include a degree-64 representation with multiplicity 11 and 13 respectively, and several others of high multiplicity, but the resulting representations of G still have relatively small entries.

7. Just as in the irreducible extension algorithm (see p. 88), we first conjugate H if possible so that one of the generators of G is in H, so the corresponding image matrix is usually sparse or has entries in a subfield, etc., and the final representation is more compact. Some examples (among very many in our database) are the following, where in each case G has two standard generators g_1, g_2 and ρ is the relevant representation:
   - For the degree-65 representation of $G = \mathrm{Sz}(8)$, $\rho(g_1)$ is diagonal with entries ±1 only (32 -1s and 33 1s, thus trace 1). (The contributing representations of H of degree 7 and 28 are monomial over $\mathbb{Z}$.)
   - For the degree-220 representation of $G = \mathrm{U}_5(2)$, g_1 has order 2 and $\rho(g_1)$ is diagonal, with entries ±1 only, while $\rho(g_2)$ has density 66.7% and entries in $\mathbb{Q}(\zeta_3)$.
   - For the degree-126 representation of $G = 3.\mathrm{McL}$, the field F is $\mathbb{Q}(\alpha)$, where the minimal polynomial of α is $x^4 - x^3 - 2x^2 - 3x + 9$. Here g_1 has order 2 and $\rho(g_1)$ is a monomial matrix with the only non-zero entries being 1 and $\pm\beta$, where $\beta = \frac{1}{6}(-\alpha^3 - 2\alpha^2 + 2\alpha + 3)$ (of order 3 in F).

6.4. Examples

We now devote a whole section to presenting examples of the use of the hybrid algorithm BBReductionRepresentation, since it is effective for a wide range of situations and there are several interesting phenomena which arise.

Example 6.4.1. Let G be the sporadic simple group J_3 and let χ be one of the degree-85 characters of G, with character field $F = \mathbb{Q}(\sqrt{-19})$. We noted in Ex. 3.9.7 that AbsolutelyIrreducibleRepresentation applied to χ returns a poor representation $\rho_1 : G \to \mathrm{GL}_{85}(F)$, and in Ex. 6.2.4 we used the entry reduction algorithm to conjugate ρ_1 to a reduced representation ρ (in 22.9s).

By using BBReductionRepresentation we could instead construct the reduced representation ρ directly without having to construct ρ_1 first, as follows (table entry on p. 168). By using the same H as in Ex. 6.2.4 (order 8160), irreducible rational representations of H having degrees 17 and 68 were first set up in 0.7s. Then a black-box representation $\mathscr{B}$ was constructed for χ, using condensation of a degree-14688 permutation representation of G: the condensation subgroup K had order 81, the condensed module $\tilde{M}$ had dimension 186, and the simple constituent $\tilde{S}$ corresponding to χ had dimension 2. The endomorphism ring of $\tilde{S}$ was isomorphic to F, as expected, so the black-box representation $\mathscr{B}$ could be set up with a suitable eigenspace (total black-box setup time 48.0s). Finally, the rest of BBReductionRepresentation used $\mathscr{B}$ and the representations of H to construct ρ affording χ in only 1.2s. The resulting representation is identical to the representation constructed in Ex. 6.2.4, but avoids the initial construction of ρ_1 (and the irreducible rational representation from which that was extracted).

Example 6.4.2. This example involves computing a representation realized over a degree-4 number field, having extracted it as a constituent of a degree-18954 induced representation, but the result still has very small entries. Let $G = 3.\mathrm{G}_2(3)$, and let χ be one of the degree-189 absolutely irreducible characters of G; χ has Schur index 1 and character field $F = \mathbb{Q}(\alpha)$, where α has minimal polynomial $x^4 - x^3 + 4x^2 + 3x + 9$. We computed a representation ρ affording χ by BBReductionRepresentation, as follows (table entry on p. 174).

The black-box representation $\mathscr{B}$ for χ was set up by using induction condensation for a degree-54 irreducible rational representation of a subgroup of G of index 351. The condensation subgroup K chosen by AutomaticCondensation had order 96 and the full condensed module $\tilde{M}$ had dimension 180. The condensed submodule $\tilde{S}$ corresponding to χ had dimension 4 (split out by the rational Meataxe in only 0.7s); it was then trivial to compute an endomorphism

$$e = \begin{pmatrix} 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & -1 \\ 1 & -1 & 1 & 1 \\ -2 & -1 & -1 & 0 \end{pmatrix}$$

of $\tilde{S}$ whose minimal polynomial is f above. The total black-box setup time was 49.2s.

The subgroup H for reduction was chosen to be a soluble subgroup of order $2^2.3^7$ so that $\chi_H = \chi \downarrow_H$ splits as four inequivalent degree-27 irreducible characters which can be realized over $\mathbb{Q}(\zeta_3)$, one with multiplicity 1 and others with multiplicity 2. Corresponding representations were constructed by inducing linear representations for a subgroup of H of index 27 (2.8s total). The remaining steps of BBReductionRepresentation took 44.8s, as follows. Step 4 used 3 primes and 4 roots per prime to construct the Hom-modules; each modular spin in the induced module of degree 18954 took 3.0s to compute the dimension-189 submodule and the reduced action. Only one prime was needed in Step 7 to compute the final representation $\rho : G \to \mathrm{GL}_{189}(F)$ affording χ, and this was verified in 1.4s. Let $\{g_1, g_2\}$ be the standard generators of G. Then $\rho(g_1)$ is monomial (since $g_1 \in H$ and the representations of H are monomial) while $\rho(g_2)$ has density 97.9% and entry denominator LCM $648 = 2^3.3^4$ and absolute maximum numerator 243; a typical entry is $\frac{1}{216}(-\alpha^3 - 8\alpha^2 + 4\alpha - 31)$.

We thus see that the algorithm BBReductionRepresentation is very effective at constructing a representation of non-trivial degree over a non-trivial number field, even when it is extracted from a very large representation (degree 18954 here).

Example 6.4.3. This example shows that it is sometimes better to use black-box representations which are based on tensor condensation instead of permutation or induction condensation. Let $G_1 = 12_1.\mathrm{U}_4(3)$ and $G_2 = 12_2.\mathrm{U}_4(3)$. Both groups have absolutely irreducible representations of degree 216 and Schur index 1, which can realized over the minimal field $\mathbb{Q}(\zeta_{12})$ of degree 4; let χ_1 and χ_2 be corresponding characters. Without using tensor condensation, the smallest-degree virtual representation for χ_1 has degree 6720 (induction of a degree-24 representation of an index-280 subgroup; 1065s to set up) and for χ_2 the degree is 12960 (induction of a degree-12 representation of an index 1080 subgroup; 875s to set up). Instead, we used a black-box representation based on tensor condensation in both cases; for G_1 we used the tensor product of representations of degrees 30 and 40 (thus virtual degree 1200; 81s to set up the black-box representation), while for G_2 we used the tensor product of representations of degrees 40 and 72 (thus virtual degree 2880; 117s to set up the black-box representation). The modular spin operations are also quicker in both cases than for the induction-based situation, since the degrees of the actions are significantly smaller. See p. 175 for more details.

Several representations of very high degree are also computed via black-box representations based on tensor condensation; e.g., the degree-7497 and -7650 representations of the sporadic Held group, both over the quadratic field $\mathbb{Q}(\sqrt{-7})$ (p. 187).

Example 6.4.4. In this example, the initial call to IrreducibleRepresentationsOverField in the hybrid algorithm needs to use Fieker's algorithm to rewrite a representation over a minimal extension field. Let $G = \mathrm{U}_3(4)$ and let χ one of the degree-75 irreducible characters of G; χ has Schur index 1 and values in $\mathbb{Q}(\zeta_{13})$, and the character field $\mathbb{Q}(\chi)$ can be written as $F = \mathbb{Q}(\alpha)$, where α has minimal polynomial $x^4 + x^3 + 2x^2 - 4x + 3$. A typical call to AbsolutelyIrreducibleRepresentation on χ takes about 3400 seconds and yields a representation with entries having numerators of up to 28 digits and denominators with 2 digits. So we used BBReductionRepresentation instead to construct a representation affording χ, as follows (table entry on p. 167).

First a black-box representation $\mathscr{B}$ for χ was constructed via the induction to G of a linear representation of an index-416 subgroup of G which condensed to a dimension-30 reduced module. The simple condensed constituent $\tilde{S}$ corresponding to χ had dimension 20 and an endomorphism of $\tilde{S}$, generating a subfield isomorphic to F, was instantly constructed. (The total time to set up $\mathscr{B}$ was 0.8s.)

Let H be the largest maximal subgroup of G; H has order 960 and is soluble, with shape $2^{2+4}.3.5$. Now $\chi_H = \chi \downarrow_H$ splits over F into irreducible characters of degree 12, 15, 48 respectively and the degree-15 and degree-48 representations can be realized over $\mathbb{Q}$ (and computed in less than a second). But the degree-12 character ψ_1 has Schur index 2. So when IrreducibleRepresentationsOverField was called on the components of χ_H and F, it first obtained an irreducible rational representation σ_1 of degree 24 corresponding to ψ_1. The endomorphism ring of σ_1 has dimension 4 and is non-commutative with trivial centre, as expected. Elements from the LLL-reduced basis of a maximal order of E and small linear combinations thereof have minimal polynomials such as $x^2 + x + 1$ and $x^2 + 2$, none of which have a root over F. In fact, the single quadratic subfield of F equals $\mathbb{Q}(\sqrt{13})$, and ψ_1 cannot be realized over this subfield. So the algorithm instead set F_0 to one of the quadratic subfields of E and then computed the corresponding absolutely irreducible representation ρ_{F_0} which is a constituent of $(\sigma_1)^{F_0}$ and then called Fieker's algorithm on ρ_{F_0} and F which immediately gave a representation $\rho_1 : H \longrightarrow \mathrm{GL}_{12}(F)$ affording ψ_1. The complete time taken in IrreducibleRepresentationsOverField was 3.0s.

Finally, the rest of BBReductionRepresentation took only 1.1s to construct a representation $\rho : G \longrightarrow \mathrm{GL}_{75}(F)$ affording χ. The total time for constructing ρ was thus 4.9 seconds. Writing the standard generators of G as g_1, g_2, we have that $\rho(g_1)$ has density 0.02% with only ± 1 for non-zero entries, while $\rho(g_2)$ has density 96.1% and maximum 3-digit numerators and denominator LCM $960 = 2^6 \cdot 3 \cdot 5$.

Example 6.4.5. This example shows that the hybrid algorithm can efficiently compute absolutely irreducible representations over a minimal field efficiently and with small entries, even when the field has very high degree.

Let $G = \mathrm{L}_2(83)$, which has a class of 20 degree-84 conjugate irreducible representations. Let χ be one of the corresponding characters, which has entries in $\mathbb{Q}(\zeta_{41})$ and Schur index

1. The minimal-degree character field of χ can be written as $F = \mathbb{Q}(\alpha)$, where α has minimal polynomial f, which is equal to

$$x^{20} + x^{19} - 19x^{18} - 18x^{17} + 153x^{16} + 136x^{15} - 680x^{14} - 560x^{13} + 1820x^{12} + 1365x^{11} - 3003x^{10} - 2002x^{9} + 3003x^{8} + 1716x^{7} - 1716x^{6} - 792x^{5} + 495x^{4} + 165x^{3} - 55x^{2} - 10x + 1.$$

We computed a representation $\rho : G \to \mathrm{GL}_{84}(F)$ affording χ, as follows (table entry on p. 195). We set H to the largest maximal subgroup of G with shape 41.83. Now $\chi_H = \chi \downarrow_H$ splits over F as $2 + 82$ (the minimal fields for these representations are $\mathbb{Q}$ and F respectively). Corresponding representations were constructed by IrreducibleRepresentationsOverField in 11.3s. A typical entry of the image of a generator in the degree-2 representation is:

$$\alpha^{16} - 16\alpha^{14} + 104\alpha^{12} - 352\alpha^{10} + 660\alpha^{8} - 672\alpha^{6} + 336\alpha^{4} - 64\alpha^{2} + 1,$$

while the degree-85 rational representation has only two non-zero entries per row, which are all ±1.

Setting up a black-box representation $\mathscr{B}$ for χ took only 3.8s, via the condensation of a degree-3403 permutation representation of G (condensed dimension 63) and the desired condensed constituent $\tilde{S}$ had dimension 20. The generators of the action of $\tilde{S}$ had entries in the range -8 to 8 and the endomorphism ring $\tilde{E}$ of $\tilde{S}$ was isomorphic to F as expected. The first non-scalar matrix e in a LLL-reduced basis of $\tilde{E}$ was a sparse 20×20 integral matrix with very small entries and with minimal polynomial equal to f, so the eigenspace of e over F was used to generate the submodule over F.

Finally, the rest of BBReductionRepresentation used $\mathscr{B}$ and the representations of H to construct ρ affording χ in only 34.8s (total time 49.9s). The image matrices both have density 98.8%, denominator LCM 83 and absolute maximum numerator 120782 (average 1796.6, 1879.6). So the entries are very small, considering the very large degree of F and the degree of χ. A sample random entry is:

$$\frac{1}{83}(-2w^{19} - 2w^{18} + 36w^{17} + 36w^{16} - 268w^{15} - 274w^{14} + 1062w^{13} + 1150w^{12} - 2394w^{11} - 2906w^{10} + 3011w^{9} + 4508w^{8} - 1849w^{7} - 4142w^{6} + 265w^{5} + 2004w^{4} + 178w^{3} - 382w^{2} - 43w + 20).$$

Note that if this representation is rewritten over the cyclotomic field $\mathbb{Q}(\zeta_{41})$ (into which F embeds), then the entries have denominator LCM 83, and the absolute numerator maximum is just 45 (average 2.4).

For comparison, we also computed the underlying representation ρ_F over F of the black-box representation $\mathscr{B}$ by calling the modular setup function $\mathscr{B}.\pi$ with enough primes till the reconstruction of the combination succeeded; this took 120 primes and 226s. The denominator LCM was a 397-digit integer and the absolute maximum numerator was a 421-digit integer. So the hybrid algorithm often does a very major reduction of the entry size of the underlying representation!

Example 6.4.6. This example shows that is sometimes worth applying the hybrid algorithm recursively all the way down a chain of subgroups which are successively maximal.

For the classical groups $\mathrm{L}_2(q)$ and $2.\mathrm{L}_2(q)$, there are absolutely irreducible representations of degree $(q-1)/2$ or $(q+1)/2$ and these can be very hard to compute over a minimal field (which is always quadratic). Let $G = 2.\mathrm{L}_2(71)$ and let χ be one of the faithful degree-36 irreducible characters of G; the character field of χ equals $F = \mathbb{Q}(\sqrt{-71})$ and

χ has Schur index 1. We computed a representation ρ affording χ as follows (table entry on p. 196).

Let H_1 be the largest maximal subgroup of G (the Borel subgroup), of order 4970. Then $\chi \downarrow_{H_1}$ splits over F as $1 + 35$. Let ψ_1 be the degree-35 character, which has character field F and Schur index 1. Computing a representation affording ψ_1 via AbsolutelyIrreducibleRepresentation yields a representation with 8-digit numerators and denominator LCM 1 (in 32 seconds), and computing a representation for χ by BBReductionRepresentation using this representation of H yields a representation whose image matrices have 11-digit numerators and denominator LCM 71.

But we could compute a representation of better quality, as follows. Consider the chain of subgroups:

$$G > H_1 > H_2 > H_3 > H_4 > 1,$$

with respective orders $357840, 4970, 2485, 497, 71, 1$ and such that each H_i is the largest maximal subgroup of the preceding subgroup (with successive indices 72, 2, 5, 7). Let ψ_i be the restriction of ψ_1 to H_i for $i = 2, 3, 4$. Each ψ_i is irreducible over F. So σ_i (affording ψ_i) was computed for $i = 4, 3, 2, 1$ successively, each time using the previous representation σ_{i+1} for $i < 4$, as follows.

- Since H_4 is cyclic, σ_4 affording ψ_4 could be constructed simply by factoring the polynomial $x^{71} - 1$ over F; this has irreducible factors of degrees 1, 35 and 35 (it takes 0.35 seconds to compute the factorization, using Trager's algorithm [Tra76]). The image of a generator of H_4 under σ_4 was defined to be the companion matrix A of one of the degree-35 factors. The entries of the last row of A are:

$$\begin{gathered}[1, \alpha + 1, \alpha - 8, -2\alpha - 14, -5\alpha - 5, -5\alpha + 13, -2\alpha + 28, 2\alpha + 33,\\ 6\alpha + 33, 11\alpha + 22, 13\alpha - 8, 9\alpha - 40, 2\alpha - 52, -3\alpha - 47, -6\alpha - 38,\\ -8\alpha - 26, -8\alpha - 12, -7\alpha - 5, -7\alpha - 2, -8\alpha + 4, -8\alpha + 18, -6\alpha + 32,\\ -3\alpha + 44, 2\alpha + 54, 9\alpha + 49, 13\alpha + 21, 11\alpha - 11, 6\alpha - 27, 2\alpha - 31,\\ -2\alpha - 30, -5\alpha - 18, -5\alpha, -2\alpha + 12, \alpha + 9, \alpha, -1]\end{gathered}$$

- For $i = 3, 2, 1$, σ_i (affording ψ_i) was computed by setting up a black-box representation for ψ_i and reducing this via σ_{i+1}.
- Finally, ρ affording χ was computed by a black-box representation for χ (via a permutation representation of degree 144), and then reducing via $[1_H, \sigma_1]$, where 1_H is the trivial representation of H.

The final representation ρ has 6-digit numerators and denominator LCM 71 and the total time taken to construct ρ was 11.3 seconds. Note that if we omit one of the subgroups in the chain, the quality of the final representation becomes poorer.

6.5. Comparison with General Extension

Here is a brief comparison of the general extension and hybrid algorithms. General extension is obviously necessary when the group G is such that the index of its maximal subgroups are so large that one cannot set up a reasonable black-box representation based on permutation or induction condensation (although tensor condensation may be applicable, such as in some of the very high-degree representations of HN). But the number of initial image matrices and the norm of $\chi \downarrow_H$ (determining the dimension of the relation ideal) have a critical effect on the time taken by the general extension algorithm, so there are many cases where it may require many more expensive matrix operations in generating all the polynomial relations and solving the final polynomial system may be difficult. Also, the final rewriting of the representation to be defined on the given generators of G may be expensive. So the hybrid algorithm is often much faster and is particularly better when the final field F over which the representation is written has high degree. Some examples below illustrate these points.

We noted in Subsec. 5.4.7 that for the general extension algorithm, one can use LLL-reduction on the final set of image matrices and the associated polynomials (before computing the point of the variety) to attempt to reduce the entries of the final representation. An alternative way to improve the quality of the output is simply to apply the algorithm EntryReductionBySubgroup to the resulting representation, reducing by representations of H_2, where H_2 is some subgroup of G which is conjugate to H; in this case, it costs nothing to set up the corresponding representations of H_2 (see point 7 on p. 138). The resulting representation always seems to have as good quality as that of the result of using BBReductionRepresentation instead.

Example 6.5.1. Let G be the sporadic simple Suzuki group Suz. We noted in Ex. 5.5.2 that the degree-143 rational representation of G can be computed by general extension in only 2.4s after degree-65 and -78 rational representations of H (the largest maximal subgroup) are set up. In contrast, if we use BBReductionRepresentation to construct this representation, then we have to condense a permutation representation of G of degree 32760 or a monomial representation of G of degree 22880 (induction of a linear representation of an index 22880 subgroup). If we use the latter, then setting up the black-box representation takes 106s, and then the rest of the algorithm takes 6s, so after the initial setup of the representations of H, BBReductionRepresentation takes 112s, compared with 2.4s for GeneralExtension.

However, some high degree representations of G are more easily handled by IrreducibleRationalRepresentations or BBReductionRepresentation since they occur as constituents of lower-degree representations: the irreducible representations of degrees 780 and 1001 (p. 182) occur in permutation representations of degree 1782. The irreducible representations of degrees 3432 (p. 185), 5005 and 5940 (p. 186) occur in the tensor square of the degree-143 representation, so we computed them using that, to avoid the situation of a large number of image matrices in the general extension algorithm.

Example 6.5.2. Consider the degree-171 representation of $G = 3.\mathrm{J}_3$ over the degree-4 number field $F = \mathbb{Q}(\alpha)$, where the minimal polynomial of α is $x^4 - x^3 + 2x^2 + x + 1$ (table entry on p. 173). Here the black-box representation was based on the induction of a degree-34 rational representation of an index-6156 subgroup; the condensed module $\tilde{M}$ had dimension 876 (condensed subgroup of order 240) and took 331s to set up, then 138s to extract the dimension-4 condensed submodule $\tilde{S}$. Then BBReductionRepresentation with a subgroup H of index 17442 took 384s to do the rest: there were 3 primes, 4 roots per prime and each modular spin took 28s (with a degree-209304 space over each finite field!). The total time taken was 860s. For the resulting representation $\rho : G \longrightarrow \mathrm{GL}_{171}(F)$, $\rho(g_1)$ has denominator LCM 64, absolute maximum numerator 32 and density 98.7%, while $\rho(g_2)$ is monomial (see [Ste11]). So despite having to extract the representation from a virtual induced representation of degree 209304, the hybrid method still yielded a small result in reasonable time.

In contrast, if we were to use general extension on the appropriate representation of the above H (of index 17442), then the largest normalized subgroup L would have order 1728 with 54 initial image matrices and the dimension of the relation ideal would have dimension 12, so this computation would be more much expensive, particularly since it would be over the number field F of degree 4. (Using the larger maximal subgroup H of index 6156 instead, the largest normalized subgroup L would have order 288 with 313 initial image matrices.)

6.6. The degree-10944 representation of the O'Nan Group

Let G be the sporadic simple O'Nan group, of order

$$460815505920 = 2^9.3^4.5.7^3.11.19.31.$$

A minimal-degree faithful representation of G has degree 10944, which can be realized over $\mathbb{Q}$. Let χ be the corresponding character. We succeeded in constructing a rational representation affording χ, but this was by far the most difficult representation in our database to construct, not just because the degree 10944 is very large, but also because the largest maximal subgroups of G are relatively small.

Let H be the largest maximal subgroup of G, which equals $\mathrm{L}_3(7){:}2$ (order 3753792, index 122760), and let $\chi_H = \chi \downarrow_H$. The norm of χ_H is 52, so there are many corresponding irreducible representations of H. For the general extension method applied to χ and a representation affording χ_H, the largest possible normalized subgroup L has order 672, with 179118 associated image matrices; since also the final relation ideal would have dimension 51, it is clearly infeasible to use this method.

Now the desired representation does occur in a degree-122760 permutation module of G, but clearly it is also infeasible to use the direct condensation-based splitting method here, since the matrices given to the Hermite form and LLL algorithms in the integral spin would be far too large. However, we were able to compute a representation affording χ by algorithm BBReductionRepresentation, as follows (table entry on p. 187).

A degree-122760 permutation representation was used to define G, and H was defined as above. The decomposition of χ_H into irreducible rational characters has the following degrees (with multiplicities):

$$1, 57, 112, 152 \times 3, 304, 342 \times 2, 343 \times 3,$$

$$399, 399 \times 2, 456, 456, 684, 684, 1368, 1728 \times 2.$$

Corresponding irreducible rational representations were constructed by either irreducible extension (degree 57, 152, 342, 343, 399, 456) or general extension, via the normal subgroup $H_2 = \mathrm{L}_3(7)$ of H in both cases. The corresponding representations of H_2 were first constructed by either direct induction or IrreducibleRationalRepresentations (160s total), except for the degree-1728 representation, which was constructed by first computing a corresponding degree-288 absolutely irreducible representation of H_2 over a degree-6 minimal field via BBReductionRepresentation, and then using restriction of scalars to $\mathbb{Q}$ (51s).

We set up a black-box representation $\mathscr{B}$ for χ as follows. Let $n = \chi(1) = 10944$. Condensation was used, where the virtual representation σ was the degree-122760 permutation representation of G, and a condensation subgroup K of order 343 was selected, so that the condensed module $\tilde{M}$ had dimension 366 (43s for setup). Then $\tilde{M}$ split as $1 + 42 + 76 + 97 + 150$ and the dimension-42 constituent $\tilde{S}$ corresponding to χ was extracted (371s for the rational Meataxe). For the uncondensation of $\tilde{S}$ inside the degree-122760 permutation module, a modular spin with parallel operations on integral vectors took 11.4 hours. Then using the resulting invariant $n \times 122760$ integral basis matrix, we could construct in a few seconds integral $n \times n$ matrices U, A_1, A_2, B_1 and B_2 (all sparse with entries mostly ± 1) such that

$$g_1 \mapsto A_1 U^{-1}, \quad g_2 \mapsto A_2 U^{-1}, \quad h_1 \mapsto B_1 U^{-1}, \quad h_2 \mapsto B_2 U^{-1}$$

defined an irreducible rational representation ρ_1 of G which afforded χ, where $\{g_1, g_2\}$ and $\{h_1, h_2\}$ are standard generators of G and H respectively. The above image matrices would have impractically large entries in $\mathbb{Q}$, so they were not computed explicitly, but we could define a black-box representation $\mathscr{B}$ for χ via ρ_1.

To apply the remaining steps of BBReductionRepresentation, we needed to construct $\rho_1 \downarrow_H$ for successive primes and compute corresponding Hom-modules for each of the irreducible representations of H. All of the Hom-modules were computed via 10 parallel processors and took about 191.0 hours total sequential time (10 primes at 19.1 hours each); the bulk of the time was in the modular Meataxe to decompose the semisimple modules corresponding to $\rho_1 \downarrow_H$ over the finite fields. The reduction of the bases of all of the rational Hom-modules in ReducedHomBasis then took 1521s total.

Finally, we could construct the rational representation ρ affording χ. Since it was easy to conjugate H initially so that the first generator g_1 of G was in H, the first image matrix $\rho(g_1)$ was constructed via the block sum of images of the representations of H (density 0.094%). This matrix has integral entries in the range -22 to 22 (43 distinct values) and trace 64, equalling $\chi(g_1)$, as expected. Finally, constructing the second image matrix $\rho(g_2)$ via conjugation of $\rho_1(g_2)$ by the transformation matrix took 3.9 hours. This matrix $\rho(g_2)$ has density 99.7% and 4003690 distinct entries (trace 64 also). The denominator LCM is $278110941696 = 2^9.3^5.7^6.19$ (maximum denominator $8941324 = 2^2.7^6.19$) and the absolute maximum numerator is 45532001, while the average numerator has only 3 digits. The larger numbers only occur in the last 5184 rows and the last 5184 columns of the matrix (since the LLL algorithm had to act on a corresponding lattice of dimension 6 in ReducedHomBasis for the degree-1728 block, so it was harder to reduce the corresponding section). A sample of 10 random entries in this portion of the matrix is the following:

$$\{\frac{46385}{460992}, \frac{35179}{460992}, \frac{41815}{319333}, \frac{23671}{91238}, \frac{15285}{76832}, -\frac{31583}{388962}, -\frac{328}{4617}, -\frac{308925}{8941324}, \frac{56407}{117649}, -\frac{43579}{268912}\}.$$

In the rest of the matrix (the top left 5760×5760 submatrix), the numbers are much smaller (average 2-digit numerators and 5-digit denominators). A sample of 10 random entries in this portion is the following:

$$\{-\frac{745}{9604}, \frac{1}{1176}, \frac{55}{5472}, -\frac{1593}{21952}, \frac{5}{196}, \frac{1024}{21609}, \frac{1375}{65664}, -\frac{31}{4802}, \frac{629}{98496}, \frac{1}{1372}\}.$$

We skipped the final verification step, since it would be extremely expensive, but we did perform several checks on the representation to verify its correctness (by computing traces of products of the matrices over $\mathbb{Q}$, and some modular checks). Also, the 10 primes used above in the modular construction of the Hom-modules are 2 more than were needed (the results of the rational reconstruction were actually covered by 8 primes), and we computed the Hom-modules on 10 more primes in parallel and verified that they were consistent with the rational Hom-modules and the conjugated image of g_2, so the result is also verified to be correct modulo an integer of the order of 10^{140}. The total time taken for the whole computation was about 202.4 hours.

6.7. Conclusion

We summarize the main features of the hybrid approach. Some of the key advantages are the following:

1. Since the underlying representation is constructed via the splitting approach, which first determines an underlying irreducible rational representation, the result is guaranteed to be written over a minimal field F.
2. This approach easily handles much higher degrees (both for the result and for the virtual representation σ) than are practical in the direct splitting approach, since the expensive saturation, Hermite form and reduction operations on large integral matrices is completely avoided.
3. This approach generally works just as well over number fields as over $\mathbb{Q}$, so does not face the major challenge of finding a good eigenspace basis in the splitting approach.
4. This approach is often much more efficient than the general extension algorithm when the number of image matrices in that algorithm is large or when the polynomial system is difficult to solve over the minimal field F. Sometimes the hybrid algorithm is even faster than irreducible extension when the number of image matrices is large.

The only real limitation of the hybrid approach is that if G has no proper subgroups of moderate degree and tensor condensation is not applicable, then one cannot set up an appropriate black-box representation so this approach will not be practical, but general extension is usually applicable in such a case.

CHAPTER 7

A General Strategy

7.1. Outline

We outline here a general procedure to compute a representation $\rho : G \to \mathrm{GL}_n(F)$ affording a given character $\chi \in \mathrm{Irr}(G)$ and such that F is a minimal field for χ, as a synthesis of all the algorithms presented in the thesis.

1. If G is too large to compute its character table, then choose a maximal subgroup H of G (e.g., by using known words in the standard generators of G), determine the decomposition of $\chi \downarrow_H$ (by inspection of the Atlas, say), then compute corresponding irreducible representations of H recursively, and then use general extension with G and the sum of these representations, without explicit use of the character (Sec. 5.6).
2. *[Now the character table of G is assumed to be computed.]* Set $d = \chi(1)$, $C = \mathbb{Q}(\chi)$, $c = \mathrm{Deg}_{\mathbb{Q}}(C)$ and $s = s_{\mathbb{Q}}(\chi)$.
3. If there exists a subgroup H of G of index l, with $l > 1$ and $lq = d$, and such that there is a $\psi \in H$ such that $\psi \uparrow^G = \chi$ and $s_{\mathbb{Q}}(\psi) \cdot \mathrm{Deg}_{\mathbb{Q}}(\mathbb{Q}(\psi)) = s \cdot c$ (choose l to be maximal under such conditions), then recursively compute a representation $\rho_H : H \to \mathrm{GL}_n(F)$ affording ψ and then return $(\rho_H) \uparrow^G$.
4. If $cs = 1$ and d is reasonably small (say, up to 1000) and computing subgroups of G is not too hard, then set $[\rho] :=$ IRREDUCIBLERATIONALREPRESENTATIONS($[\chi]$) and return ρ. During the algorithm, if there is no virtual representation of reasonable degree (say, less than 100,000), then abort and go to Step 6.
5. If $cs \neq 1$ and d is reasonably small (say, up to 200) and computing subgroups of G is not too hard, then set $\rho :=$ ABSOLUTELYIRREDUCIBLEREPRESENTATION(χ) and return ρ. If there is no virtual representation of reasonable degree, or if the basis of the eigenspace is not sparse enough after reduction, then abort and go to the next step.
6. If there is a maximal subgroup H of G such that $\chi_H = \chi \downarrow_H$ is absolutely irreducible, then construct ρ_H affording χ_H recursively, then set $\rho :=$ IRREDUCIBLEEXTENSION(χ, ρ_H) and return ρ.
7. Choose a maximal subgroup H of G. Usually, this should one of the largest maximal subgroups, but not necessarily; a smaller H may be such that computing the relevant representations of H are easier to compute recursively.
8. If proper subgroups of G only have very large index (i.e., so permutation or induced representations from subgroups will have very large degree), or one can compute a normalized subgroup $L \leq H$ so that the norm of $\chi \downarrow_L$ is not too large, then use GENERALEXTENSION on χ and H; otherwise use BBREDUCTIONREPRESENTATION

on χ and H. In either case, to construct ρ_H affording $\chi_H = \chi \downarrow_H$, either use IrreducibleRepresentationsOverField (the default algorithm) or recurse on each irreducible component of χ_H.

7.2. Examples

Example 7.2.1. In this example, BBReductionRepresentation is used twice, with an irreducible extension in between. Let $G = \mathrm{U}_3(13)$, and let χ be one of the degree-157 absolutely irreducible characters of G. χ has character field $F = \mathbb{Q}(\zeta_7)$ (degree 6) and Schur index 1. We constructed a representation over F affording χ as follows (table entry on p. 172).

First a black-box representation $\mathscr{B}$ for χ was constructed via a permutation representation of degree 15386 which condensed to a dimension-94 module $\tilde{M}$; the simple condensed constituent $\tilde{S}$ corresponding to χ had dimension 20 (90s).

Now let H be the largest maximal subgroup of G, which has index 2198 in G and shape $2.2^{1+1}.3.7.13^{1+2}$, and let $\chi_H = \chi \downarrow_H$, which splits over F as $1 + 156$. Let ψ be the degree-156 character of H. Computing a representation affording ψ is non-trivial, so instead of using using the simple IrreducibleRepresentationsOverField which effectively maps to AbsolutelyIrreducibleRepresentation, we did the following. The representation can be computed via irreducible extension for a subgroup H_2 of index 7 in H. The character $\psi_2 = \psi \downarrow_{H_2}$ has Schur index 2, so again it is non-trivial to compute a representation affording it. So let H_3 be a subgroup of index 4 in H_2; then $\psi_3 = \psi \downarrow_{H_3}$ is a rational irreducible character with Schur index 1. Computing $\sigma_3 : H_3 \to \mathrm{GL}_{153}(\mathbb{Q})$ affording ψ_3 was easy via IrreducibleRationalRepresentations (2.9s). Then BBReductionRepresentation was applied to ψ_2 and σ_3; the black-box representation $\mathscr{B}_2$ for ψ was constructed from a degree-2198 permutation representation of H_2 and this yielded $\sigma_2 : H_2 \to \mathrm{GL}_{153}(F_2)$, where $F_2 = \mathbb{Q}(\sqrt{-7})$ is a subfield of F (16.2s). Then $\sigma_1 : H \to \mathrm{GL}_{153}(F)$ could be constructed via irreducible extension of σ_2, with only 2 image matrices (10.0s). The total time for constructing the representation of H was 29 seconds.

Finally, $\mathscr{B}$ and the linear representation of H together with σ_1 could be used to construct $\rho : G \to \mathrm{GL}_{154}(F)$ affording χ (the rest of BBReductionRepresentation took 13.5s). The total time for constructing ρ was thus 104s. If the standard generators of G are g_1, g_2, then $\rho(g_1)$ has density 99.9%, absolute maximum numerator 154 and denominator LCM $169 = 13^2$, while $\rho(g_2)$ has density 0.01% with only ± 1 for non-zero entries. A typical entry of $\rho(g_1)$ is the following:

$$\frac{1}{169}(21\alpha^5 + 40\alpha^4 - 14\alpha^3 + 35\alpha^2 + 3\alpha - 19).$$

Example 7.2.2. Let $G = 3.\mathrm{U}_3(17)$ and let χ be one of the irreducible characters of G of minimal degree. χ has degree 273 and character field $F = \mathbb{Q}(\zeta_9)$ (of degree 6) and Schur index 1. We constructed a representation $\rho : G \to \mathrm{GL}_{273}(F)$ affording χ (table entry on p. 178). This involved non-trivial use of practically every algorithm in this thesis!

First let H_1 be the largest maximal subgroup of G, which is a soluble group of order $2^5.3^2.17^3$ (index 4914). Then $\chi_1 = \chi \downarrow_{H_1}$ splits as $1_6 + 272_6$, so GeneralExtension could be applied to χ and a representation of H_1 corresponding to this decomposition.

To set up the representation $\sigma_1 : H_1 \to \mathrm{GL}_{272}(F)$, we first moved down to a subgroup H_2 of H_1 (index 9) such that the restriction to H_2 of the degree-272 character was also irreducible. So a corresponding representation $\sigma_2 : H_2 \to \mathrm{GL}_{272}(\mathbb{Q}(\zeta_3))$ was constructed by BBReductionRepresentation in 350s, as follows. The black-box representation $\mathscr{B}$ was constructed from the condensation of the induction to H_2 of a degree-32 rational representation σ_2' of an index-34 subgroup H_2' of H_2; constructing σ_2' itself was the hardest step and involved splitting a homogeneous (condensed) module of dimension 128 with endomorphism centre dimension 2, Schur index 2 and multiplicity 4 by SplitHomogeneous: the maximal order O took 87s to compute and there was no split element arising from the elements of a LLL-reduced basis of O or products of such, but a sum of such was a split element and this gave a decomposition into simple components immediately. Next, representations of a subgroup H_3 of H_2 (index 289) were used for the reduction of $\mathscr{B}$ to set up σ_2: the two corresponding representations both had degree 16 with multiplicities 8 and 9 respectively (constructed by IrreducibleRepresentationsOverField in 96s). So then the representation $\sigma_1 : H_1 \to \mathrm{GL}_{272}(F)$ could be constructed by using IrreducibleExtension twice (index 3 and normal both times) to extend σ_2 from H_2 to H_1 (20s).

Finally, GeneralExtension was applied to χ and the degree-1 and -272 representations of H_1. The largest possible normalized subgroup L of H_1 had order only 288, yielding 791 initial image matrices! Nevertheless, linear reduction with χ reduced this to only 6 image matrices, and then one order-2 group relation yielded a relation ideal of dimension 1 generated by:

$$x_1x_2 + \frac{1}{9826}(\zeta_9^5 + \zeta_9^4 + \zeta_9)$$

(934s total). Computing the image matrix corresponding to a solution and rewriting the representation on the original generators of G took 29s. The total time taken for the whole computation of $\rho : G \to \mathrm{GL}_{273}(F)$ was 1333s. The LCM of the entry denominators is $578 = 2.17^2$ and the absolute maximum entry numerator is 1171; typical entries of both image matrices of ρ are:

$$\frac{1}{578}(-33\zeta_9^5 - 72\zeta_9^4 + 185\zeta_9^3 - 59\zeta_9^2 + 71\zeta_9 + 14),$$

$$\frac{1}{578}(7\zeta_9^5 + 11\zeta_9^4 - 31\zeta_9^3 - 21\zeta_9^2 + 20\zeta_9 - 68).$$

Note that the minimal-degree representation from which one could construct a black-box representation for χ and use BBReductionRepresentation has degree 2673216 (induction of degree 272, index 4914), so using that algorithm would take much longer.

Example 7.2.3. There is sometimes non-trivial recursion in the use of the general extension algorithm; e.g., the degree-1938 representation of ${}^2\mathrm{E}_6(2)$ depends on the degree-833 and -1105 representations of $\mathrm{F}_4(2)$, which depend themselves on the degree-253, -510 and -595 representations of $\mathrm{S}_8(2)$, etc. See the higher-degree table in Chapter 9.

Part 2

A Database of Irreducible Representations

CHAPTER 8

Information about the Tables

The rest of the thesis presents several tables describing our database of irreducible ordinary representations with detailed information on how each representation was constructed. This chapter contains a guide on how to read the tables.

Each entry in each table describes a faithful representation $\rho : G \to \mathrm{GL}_n(F)$, which is always absolutely irreducible. Let χ be the character afforded by ρ. The fields for the entry are as follows:

- The field in the column labelled **Deg** gives the degree n of the representation.
- The field in the column labelled **Group** describes the group (matching the Atlas notation and that of Hiss/Malle for the quasi-simple representations to degree 250). An asterisk (*) after the group name indicates that the representation is a minimal-degree faithful representation of the group.
- The field C in the column labelled **C** gives the degree over $\mathbb{Q}$ of the character field $\mathbb{Q}(\chi)$, while the field S in the column labelled **S** gives the Schur index $S = s_{\mathbb{Q}}(\chi)$. The number field F over which the representation is realized always has degree C×S over $\mathbb{Q}$ (so $F = \mathbb{Q}$ if and only if $C = S = 1$) and F is thus **always** a field of minimal degree for the constructed representation. Note also that F is always an abelian extension of $\mathbb{Q}$ (we have been able to ensure this fairly easily in all cases).
- The field in the column labelled **N/D** describes the size of the entries of the final representation ρ and is generally of the form N/D, meaning that in all the rational coefficients of the entries of the matrices defining the representation, the absolute value of all numerators is at most N (typically of the form 'xd', meaning x decimal digits, or simply '1', meaning all non-zero numerators are ± 1) and the LCM of all denominators is D. If the representation is realized over $\mathbb{Q}$ ($C = S = 1$) and the representation is also integral (a very common case), then the '/1' is omitted. However, a '/1' is always kept for irrational representations when relevant, just to make it clear that the number field elements do not have a non-trivial denominator (since the algorithms constructing irrational representations do not always yield integral integral representations). An 's' indicates that the representation is also sparse: the matrices defining the representation all have density 10% or less (very commonly, the density will be very much lower, particularly if the representation is monomial).

  The LCM of the denominators is given so that one can see which primes divide the denominators of at least one entry. Note that we can effectively construct a mod-p representation from any of the constructed ordinary representations, for any prime p. If p does not divide the denominators of the entries in the matrices defining the representation, then of course one can reduce the representation modulo p immediately (perhaps writing the result over an extension field of $\mathbb{F}_p$ if F is not $\mathbb{Q}$). For the case that p does divide a denominator, we have a developed a p-adic variant of the algorithm in Sec. 1.10 to conjugate the representation to an integral representation; the p-adic algorithm only needs to compute modulo p^k for suitable k (instead of over $\mathbb{Q}$). We are thus easily able to reduce any of the constructed representations modulo any prime. This algorithm generally takes a small number of seconds for degree up to 1000, but can also handle much higher degrees effectively; see the discussion on modular representations of the Baby Monster at the end of Sec. 5.11, for example. For the high-degree representations of the sporadic groups, we constructed several derived modular representations and checked that they were equivalent to ones in the online ATLAS [WWT+] when such were present.

- The field in the column labelled **Time** describes the time taken to construct the representation in seconds; if a time is at least 10 seconds, then the number of seconds is rounded to the nearest integer. 'Th' indicates T hours when the time is greater than an hour. If the algorithm is naturally split into two main stages, then the time is split accordingly (the details are explained below, depending on the method).

  In some cases, where the main method involves extension of a representation ρ_H of a subgroup H, an entry for ρ_H (or at least its major components when ρ_H is not absolutely irreducible) is already in one of the tables, so the time is given as '$+E$', indicating that the time was E seconds for constructing the representation affording χ, assuming that ρ_H was already constructed (and the time for constructing the nontrivial components of ρ_H can be seen elsewhere).

  Note that apart from the cases for which general extension was used without explicit use of the character, we assume in general that the character table has first been computed for G so the time for this is not included. The main reason for this is that our main algorithms to compute irreducible representations take character(s) as input and so the computation of irreducible characters is not a part of the algorithms proper. The other reason is that for many of the groups, we have computed the character table once and then computed all of the relevant representations of G in one MAGMA session, so the character table construction is shared by the construction of all the representations. As we have noted before, the computation of the character table takes a very small number of seconds for most of the groups covered here anyway, and in many cases where it is very expensive, we have used general extension without explicit use of the character instead (see Ex. 5.6.1).

Since a variety of methods are used, the entry in the column labelled **Method** gives detailed information on which major algorithm was used, as indicated by the following tags:

- **IRR**: Here $C = S = 1$ always, so the representation is rational and the algorithm IRREDUCIBLERATIONALREPRESENTATIONS (p. 66) was used to construct the representation. This tag is followed by one of the following indicators:
  - 'perm D cC': this indicates that permutation condensation was used: for a virtual permutation representation of G of degree D, the desired component of the corresponding permutation module was split and the full condensed module $\tilde{M}$ (in algorithm AUTOMATICCONDENSATION) had dimension C. Note that if the

degree D is small (typically under 100), then often the direct permutation module was split without condensation, so condensation is not needed and 'cC' is omitted.

  - 'ind iI dD cC': this indicates that induction condensation was used: for a subgroup H of index I in G, a rational representation M_H of degree D was constructed by recursively calling IRREDUCIBLERATIONALREPRESENTATIONS, and then the condensation of the induction of M_H up to G was used; the condensed module had dimension C.
  - '$\rho_a \otimes \rho_b$ cC': this indicates that tensor condensation was used: first irreducible rational representations ρ_a, ρ_b of G of degrees a and b respectively were computed (usually by an earlier stage of IRREDUCIBLERATIONALREPRESENTATIONS), and then the condensation of $\rho_a \otimes \rho_b$ was used; the condensed module had dimension C. If ρ_b is identical to ρ_a (so the tensor square is used), then the notation '$(\rho_a)^2$' is used.

  Since the representation returned by this function is **always** integral, the denominator LCM is always 1, so the 'N/D' field omits the '/1'. The time entry simply gives the total time for the call to IRREDUCIBLERATIONALREPRESENTATIONS.

- **AIR**: Here at least one of C and S is not 1, so the representation had to be realized over a proper extension of $\mathbb{Q}$ and the algorithm ABSOLUTELYIRREDUCIBLEREPRESENTATION (p. 74) was used to construct the representation. Recall that this algorithm simply calls IRREDUCIBLERATIONALREPRESENTATIONS and then constructs an absolutely irreducible representation via an eigenspace of an endomorphism over a suitable number field of minimal degree. Thus the tag is followed by the 'perm' 'ind', or '$\otimes$' indicators, exactly as above, showing how the subalgorithm IRREDUCIBLERATIONALREPRESENTATIONS first constructed the rational representation.

  The time entry has the form $T_R + T_C$, meaning T_R seconds for the call to IRREDUCIBLERATIONALREPRESENTATIONS and T_C seconds for the call to SPLITBYEIGENSPACE (p. 74). Typically, T_C is smaller for T_R for the cases covered here, but not always.

- **IE**: This indicates that the representation was constructed by calling algorithm IRREDUCIBLEEXTENSION (p. 86). A maximal subgroup H of G was first selected such that $\chi_H = \chi \downarrow_H$ was absolutely irreducible. Then a representation ρ_H affording χ_H (over a minimal field) was constructed and then IRREDUCIBLEEXTENSION was called on χ and ρ_H. The tag 'IE' is either followed by a description of H if there is a well-known form which is brief; otherwise 'iI' is used, indicating that H has index I in G. Details on how ρ_H was constructed are generally added in parentheses, unless that is trivial or too complicated to outline; most of the time, this involves a call to IRREDUCIBLERATIONALREPRESENTATIONS or ABSOLUTELYIRREDUCIBLEREPRESENTATION, in which case the 'perm' or 'ind' indicators are used, just as above, but with 'RR' and 'AIR' omitted to save space. If the algorithm IRREDUCIBLEEXTENSION is called recursively (so as to extend from a non-maximal subgroup), then 'iI_2' is given, indicating irreducible extension from a subgroup of H of index I_2, etc.

The time entry has the form T_H+T_E, meaning T_H seconds for the time to construct the representation ρ_H (by whatever method) and T_E seconds for the internal steps of IRREDUCIBLEEXTENSION.

- **GE**: This indicates that the representation was constructed by algorithm GENERALEXTENSION (p. 97). If χ was not used explicitly (using the variant algorithm of Sec. 5.6), then '$[\neg\chi]$' (meaning 'no χ') is appended to the initial tag.

  A subgroup H of G was first selected and is described in the same way as for irreducible extension above (again, often H was the largest maximal subgroup of G, but not always). Next, suitable irreducible F-representations of H were constructed (usually by algorithm IRREDUCIBLEREPRESENTATIONSOVERFIELD [p. 77]) to make up the relevant block-diagonal representation ρ_H of H affording $\chi \downarrow_H$. The list of representations of H, corresponding to the decomposition of χ_H into **distinct** characters from $\mathrm{Irr}_F(H)$ (with multiplicities), is described by a list in square brackets with entries of the form d_f^m, where for each representation of H in the decomposition, d is the degree of the representation, f is the degree over $\mathbb{Q}$ of the **minimal** number field over which it can be realized and such that the field embeds into the target field F, and m is the multiplicity of that representation in χ_H. Note that the corresponding representation is thus irreducible over F, but not necessarily absolutely irreducible. We use the multiplicative notation d_f^m instead of $d_f \times m$ simply to save space. If the degree f is 1, then the subscript 1 is omitted, while if the multiplicity m is 1, then the superscript 1 is omitted. We sometimes also use the notation $d_f^{m_1+\ldots+m_k}$, which indicates k **inequivalent** representations, each of degree d and written over a subfield of degree f and occurring with multiplicity $m_1, \ldots, m_k$ respectively ($d_f^{k\times m}$ is the same with $m_1 = \ldots = m_k = m$). Finally, if all the representations are over a field of degree f, then the subscript f is often placed outside the list to save space. Examples of this notation are the following:

  (1) For the degree-273 representation of $3.\mathrm{U}_3(17)$ (p. 178), H is a subgroup of G of index 4914, and the list of corresponding representations of H is described by $[1, 272]_6$, indicating that χ_H splits into representations of degree 1 and 272, such that a minimal field for both representations has degree 6.

  (2) For the degree-3344 representation of HN (p. 185), H is a subgroup of G equal to A_{12}, and the list of corresponding representations of H is described by $[1, 54, 132^2, 462^2, 616, 1485]$, indicating that χ_H splits into irreducible rational representations of these degrees, and the degree-132 and -462 representations occur with multiplicity 2.

  The time entry has the form T_H+T_E, meaning T_H seconds for the time to construct the representations of H, and T_E seconds to compute the general extension algorithm on χ and the representations of H. Again, this makes it clear how much time is spent on constructing the relevant representation(s) for H and how much time is spent on extending this to the representation for G.

  For some representations, the **G[I]E**$[\neg\chi]$ tag is used, indicating that the general extension algorithm without explicit character was used, even though ρ_H was absolutely irreducible; this is used in the case that it is too hard to compute the character χ and use the direct IRREDUCIBLEEXTENSION algorithm.

- **BB**: This indicates that the representation was constructed by calling the hybrid algorithm BBReductionRepresentation (p. 134). A subgroup H of G was first selected (again, this was often the largest maximal subgroup of G, but sometimes H was not even a maximal subgroup). Then BBReductionRepresentation was called on χ and H.

  The tags after 'BB' indicate how the black-box representation $\mathscr{B}$ for χ was constructed; this is similar to the IRR and AIR cases above but is abbreviated slightly to save space: 'pD' indicates that a permutation representation of degree D was condensed, 'iI dD' indicates that an induced irreducible rational representation σ was used (where I is the index of the subgroup and D is the degree of σ), while '$d_a \otimes d_b$' or '$(d_a)^2$' indicate tensor products as for the IRR case above. The indicator 'cC' again indicates the dimension C of the condensed module $\tilde{M}$ in all cases. The field F over which ρ is written is derived from $\mathscr{B}$ and is always minimal. Note that the simple submodule $\tilde{S}$ of $\tilde{M}$ which is used in $\mathscr{B}$ will generally have smaller dimension than C, of course; space considerations force this dimension to be omitted, but it is very often the degree of the field F, or only a small multiple of that.

  After this, the tag 'RiI' indicates the index of the subgroup H by which the final representation was reduced (a description of H is given instead of 'iI' if it is brief). Corresponding representations of H were usually first constructed by calling IrreducibleRepresentationsOverField (p. 77) on $\chi_H = \chi \downarrow_H$ and F. The list of representations of H, corresponding to the decomposition of χ_H into distinct characters from $\mathrm{Irr}_F(H)$ (with multiplicities), is described by a list exactly as for the general extension case above. For example:

  (1) For the degree-65 representation of Sz(8) (p. 166), H is a subgroup of G of index 65, and the list of corresponding representations of H is described by $[2_3, 7, 28^2]$, indicating that χ_H splits into 4 representations: a degree-2 representation written over a degree-3 number field, an irreducible rational representation of degree 7, and an irreducible rational representation of degree 28, occurring with multiplicity 2.

  (2) For the degree-8250 representation of McL (p. 187), which is written over the quadratic field $F = \mathbb{Q}(\sqrt{-7})$, the list describing the representations of H is $[140, 210, 315^{1+1}, 420, 560^{3\times1}, 640_2^{1+2}, 729^2, 896^2]$. The $560^{3\times1}$ means that there are 3 inequivalent degree-560 rational representations, each occurring with multiplicity 1. The 640_2^{1+2} means that there are 2 inequivalent degree-640 representations written over F, occurring with multiplicity 1 and 2 respectively.

  The time entry has the form $T_H + T_R$, meaning T_H seconds for the time to construct the representations of H (just as for the 'BB' case above), and T_R seconds to set up the black-box representation $\mathscr{B}$ and then do the rest of BBReductionRepresentation (the latter is typically very fast because of the modular conjugation, so the bulk of T_R typically comes from the search in IrreducibleRationalRepresentations to set up $\mathscr{B}$). This makes it clear how much time is spent on constructing the relevant representation(s) of H and how much time is spent on extending this to the representation of G.

- '**DI** iI dD': This indicates that the representation ρ was constructed as the direct induction to G of a degree-d representation ρ_H of an index-I subgroup H of G. Note that in this case, ρ_H must have been realized over a subfield of a minimal field F for χ.
- '$\rho_a \otimes \rho_b$': This indicates that the representation ρ was constructed as the direct tensor product of irreducible rational representations of G of degrees a, b respectively. These representations must have been realized over subfields of a minimal field F for χ.

Finally, if there is a discussion on the construction of the representation in the main text, there is a page reference given in parentheses.

CHAPTER 9

Representations of Quasi-simple Groups

9.1. The Hiss/Malle Classification to degree 250

A quasi-simple group G is a group that is a perfect central extension of a simple group. Hiss and Malle have classified all faithful irreducible representations $\rho : G \to \mathrm{GL}_n(F)$, where G is a quasi-simple group, $n \leq 250$, and the characteristic of the field F does not equal the defining characteristic of G if G is a group of Lie type ([HM01]; corrected version [HM02]). They give a general characterization of the irreducible representations of A_n, $\mathrm{L}_2(q)$ and $2.\mathrm{L}_2(q)$ and then they present a large table listing all the other possible representations up to degree 250.

The $\mathrm{L}_2(q)$ and $2.\mathrm{L}_2(q)$ representations will be covered in the next chapter. In this chapter we will consider the large table of Hiss & Malle which lists all the other representations. We have constructed a database containing a representation for every single ordinary (non-modular) entry in this table; every representation is written over a field of minimal degree. We present here a table which gives information on the representations and how they were constructed (see the previous chapter for details on how to read the table). Our table follows the order of the corresponding table of Hiss & Malle exactly: the only omissions are the purely modular representations, which are not of relevance to this thesis, of course. We omit the irrationalities of the characters to save space (see the original paper for details). We have also discovered some minor errors which remain in the corrected paper [HM02]:

- Degree 61, $\mathrm{U}_5(3)$ [p. 108]: there should also be a rational character, with Schur indicator + (Schur index 1).
- Degree 62/63, $\mathrm{S}_6(5)$ and $2.\mathrm{S}_6(5)$ [p. 108]: the groups are round the wrong way. That is, the degree should be 63 for $\mathrm{S}_6(5)$ and the degree should be 62 for $2.\mathrm{S}_6(5)$.
- Degree 204, $\mathrm{U}_5(4)$ [p. 123]: the Schur indicator should be '-' instead of 'o'.

Note that we have not used any external ordinary representations at all in constructing our database (there are several such in the online Atlas and also in separate databases built by D. Holt and S. Nickerson). Every representation has been computed from scratch, starting from only a permutation or modular matrix representation of the group, and using only the algorithms described in this thesis. Some representations could also be computed by other special techniques (e.g., the degree-24 representation of $2.\mathrm{Co}_1$ can be computed as the automorphism group of the Leech lattice in MAGMA in about 20 seconds), but we managed to construct all representations using only the algorithms described here. Several of the representations can be seen at the webpage [Ste11].

We note the following statistics for this table:

- There are 669 representations.

- There are 353 rational representations; of these, 323 were computed by IrreducibleRationalRepresentations (196 by IRR perm, 124 by IRR ind, 3 by IRR $\otimes$) and 30 by other methods.
- There are 316 irrational representations; of these, 117 were computed by AbsolutelyIrreducibleRepresentation (19 by AIR perm, 97 by AIR ind, 1 by AIR $\otimes$) and 199 by other methods.
- 89 representations were computed by BBReductionRepresentation.
- 81 representations were computed by IrreducibleExtension.
- 14 representations were computed by GeneralExtension.

Deg	Group	C	S	Method	N/D	Time
3	$3.A_6$ *	4	1	AIR ind i10 d6 c12	1/2	0.3 + 0.1
4	$2.A_6$ *	1	2	AIR ind i6 d8 c16	1/1	0.3 + 0.2
4	$2.A_7$ *	2	1	AIR ind i7 d8 c4	1/1	0.8 + 0.1
4	$2.U_4(2)$ *	2	1	AIR ind i40 d2 c8	1/1	1.1 + 0.1
5	A_6	1	1	IRR perm6	s 1	0.0
5	$U_4(2)$ *	2	1	AIR ind i40 d1 c6	1/1	0.3 + 0.1
6	$3.A_6$	2	1	DI i6 d1	s 1/1	0.0
6	$6.A_6$ *	2	1	AIR ind i6 d12 c16	2d/2	1.3 + 0.5
6	A_7	1	1	IRR perm7	s 1/1	0.1
6	$3.A_7$ *	2	1	AIR ind i21 d2 c10	1d/1	0.1 + 0.3
6	$6.A_7$ *	4	1	AIR ind i17 d24 c24	2d/2	5.2 + 0.4
6	$6.L_3(4)$ *	2	1	AIR ind i21 d12 c52	1/1	3.1 + 0.1
6	$U_3(3)$ *	1	2	AIR ind i36 d6 c52	s 1/1	0.2 + 0.0
6	$U_4(2)$ *	1	1	IRR perm27	s 1	0.1
6	$6_1.U_4(3)$ *	2	1	AIR ind i378 d1 c36	1/1	3.7 + 0.1
6	$2.J_2$ *	2	2	AIR ind i100 d12 c240	1/1	5.6 + 0.1
7	A_8	1	1	IRR perm8	s 1	0.1
7	$U_3(3)$	1	1	IRR ind i28 d1 c4	s 1	0.1
7	$U_3(3)$	2	1	AIR perm36	1/1	0.1 + 0.1
7	$S_6(2)$ *	1	1	IRR ind i28 d1 c26	s 1	0.2
8	A_6	2	1	AIR ind i15 d2 c6	1d/1	0.6
8	$2.A_6$	2	2	AIR ind i6 d8 c8	1d/1	0.4
8	$2.A_8$	1	1	IRR ind i8 d8 c2	1	1.1
8	A_9	1	1	IRR perm9	s 1	0.1
8	$2.A_9$ *	1	1	IRR ind i9 d8 c8	1	1.8
8	$4_1.L_3(4)$ *	4	1	AIR ind i21 d32 c96	2d/1	4.5 + 0.6
8	$2.S_6(2)$ *	1	1	IRR ind i120 d1 c14	s 1	0.2
8	$2.O_8^+(2)$ *	1	1	IRR ind i120 d1 c16	s 1	0.3
9	A_6	1	1	IRR perm10	1	0.1
9	$3.A_6$	2	1	AIR ind i15 d2 c6	1d/1	0.5 + 0.0
9	A_{10}	1	1	IRR perm10	s 1	0.0

Deg	Group	C	S	Method	N/D	Time
10	A_6	1	1	IRR perm30	1	0.1
10	$2.A_6$	2	2	DI i10 d1	s 1	0.1
10	A_7	2	1	AIR ind i35 d1 c5	1d/2	0.1 + 0.1
10	A_{11}	1	1	IRR perm11	s 1	0.1
10	$2.L_3(4)$ *	2	1	AIR ind i56 d1 c8	1d/1	0.3 + 0.1
10	$U_4(2)$	2	1	AIR ind i40 d2 c8	1d/1	0.3 + 0.1
10	$U_5(2)$ *	2	1	AIR ind i165 d4 c52	1d/1	5.0 + 0.2
10	M_{11} *	1	1	IRR perm11	s 1	0.0
10	M_{11}	2	1	AIR ind i12 d10 c14	1d/1	0.1 + 0.0
10	$2.M_{12}$ *	2	1	AIR ind i12 d20 c28	1d/1	1.2 + 0.1
10	$2.M_{22}$ *	2	1	AIR ind i22 d20 c60	1d/1	1.4 + 0.1
11	A_{12}	1	1	IRR perm12	s 1	0.1
11	$U_5(2)$	1	1	IRR ind i297 d2 c36	1d/1	0.6 + 0.1
11	M_{11}	1	1	IRR perm11	s 1	0.1
11	M_{12} *	1	1	IRR perm12	s 1	0.1
12	$6.A_6$	2	1	DI i6 d2 (i5 d4)	s 1d/2	0.7 + 0.0
12	A_{13}	1	1	IRR perm13	s 1d	0.0
12	$L_3(3)$ *	1	1	IRR perm13	s 1	0.1
12	$U_3(4)$ *	1	2	AIR ind i65 d24 c313	1d/1	5.2 + 0.1
12	$2.S_4(5)$ *	2	2	AIR ind i156 d8 c112	1d/1	8.9 + 0.3
12	$2.G_2(4)$ *	1	2	IE i2080 (ind i2 d12)	1d/1	7.8 + 0.6
12	$2.M_{12}$	1	1	IRR perm24	s 1	0.1
12	6.Suz *	2	1	IE i57480192 (i3 d24 c24)	1d/1	28 + 42
13	A_{14}	1	1	IRR perm14	s 1	0.0
13	$L_3(3)$	1	1	IRR perm26	s 1	0.1
13	$U_3(4)$	4	1	AIR ind i65 d4 c16	1d/1	0.4 + 0.3
13	$S_4(5)$ *	2	1	AIR ind i156 d1 c8	1d/1	0.2 + 0.1
13	$S_6(3)$ *	2	1	IE i155520 (ind i3 d26)	1d/1	0.8 + 0.2
14	A_7	2	1	AIR perm15	1/1	0.1
14	$2.A_7$	2	2	BB i15 d16 c36 Ri7 $[4_2, 8_4]$	1d/6	0.7 + 1.0
14	A_8	1	1	IRR perm15	s 1	0.1
14	A_{15}	1	1	IRR perm15	s 1	0.0
14	$U_3(3)$	1	1	IRR perm63 c4	s 1	0.0
14	$2.S_6(3)$ *	2	1	IE i155520 (perm 56)	1d/1	2.5 + 0.3
14	$Sz(8)$ *	2	1	AIR ind i560 d2 c88	1d/1	1.0 + 0.2
14	$G_2(3)$	1	1	IRR ind i378 d1 c30	s 1	0.8
14	J_2 *	2	1	AIR perm315 c27	1d/1	0.6 + 0.1
14	$2.J_2$	1	2	AIR ind i280 d2 c44	1d/1	1.0 + 0.1
15	$3.A_6$	2	1	DI i15 d1	s 1/1	0.0
15	A_7	1	1	IRR ind i21 d1 c3	s 1/1	0.1
15	$3.A_7$	2	1	AIR ind i21 d8 c24	1d/1	0.6 + 0.1
15	A_{16}	1	1	IRR perm16	s 1	0.0
15	$3.L_3(4)$	2	1	AIR ind i120 d2 c36	1d/1	0.3 + 0.1
15	$U_4(2)$	1	1	IRR perm36	s 1	0.2

Deg	Group	C	S	Method	N/D	Time
15	$3_1.U_4(3)$ *	2	1	AIR ind i540 d2 c84	1d/1	1.2 + 0.1
15	$S_6(2)$	1	1	IRR ind i36 d1 c4	s 1	0.2
16	$2.A_{10}$	1	1	IRR ind i10 d8 c8	1	9.2
16	$2.A_{11}$	2	1	IE i2520 (i11)	2d/1	5.3 + 1.5
16	A_{17}	1	1	IRR perm17	s 1	0.0
16	$L_3(3)$	2	1	IE i13 (RR i9 d2)	2d/27	0.5 + 0.1
16	M_{11}	2	1	AIR perm144 c6	2d/1	0.1 + 0.4
16	M_{12}	2	1	AIR perm144 c14	2d/1	0.2 + 0.5
17	A_{18}	1	1	IRR perm18	s 1	0.0
18	A_{19}	1	1	IRR perm19	s 1	0.0
18	$S_4(4)$	1	1	IRR ind i120 d1 c8	s 1d	0.2
18	$3.J_3$ *	4	1	IE i25840 (DI i18 d1)	2d/16	0.5 + 7.0
19	A_{20}	1	1	IRR perm20	s 1	0.1
20	$2.A_7$	1	2	AIR ind i7 d8 c8	1d/1	1.0 + 0.1
20	A_8	1	1	IRR perm15	s 1	0.1
20	A_{21}	1	1	IRR perm21	s 1	0.0
20	$L_3(4)$ *	1	1	IRR perm21	s 1	0.1
20	$4_2.L_3(4)$ *	2	1	AIR ind i21 d8 c24	1d/1	0.9 + 0.1
20	$U_3(5)$ *	1	2	AIR ind i50 d20 c120	1d/1	1.2 + 0.1
20	$U_4(2)$	1	1	IRR perm27	s 1	0.1
20	$2.U_4(2)$	1	1	IRR ind i40 d4 c28	1d	0.9
20	$2.U_4(2)$	2	1	AIR perm80 c12	1d/1	1.0 + 0.1
20	$2.U_4(3)$ *	1	2	AIR ind i280 d4 c88	1d/1	10.3 + 0.1
20	$4.U_4(3)$ *	2	1	AIR ind i280 d4 c84	1d/1	19.2 + 0.2
21	A_7	1	1	IRR perm42	1	0.1
21	$3.A_7$	2	1	AIR ind i7 d12 c12	1d/1	0.7 + 0.1
21	A_8	1	1	IRR perm56	s 1	0.1
21	A_8	2	1	IE 15 (DI i21 d1)	1d/8	0.1 + 0.1
21	A_9	2	1	IE 120 (AIR ind i28 d1)	1d/6	0.1 + 0.1
21	$3.L_3(4)$	2	1	AIR perm63	s 1/1	0.1 + 0.1
21	$U_3(3)$	1	1	IRR ind i28 d1 c4	s 1	0.1
21	$U_3(3)$	2	1	AIR ind i56 d1 c4	s 1d/1	0.1 + 0.1
21	$U_3(5)$	1	1	IRR perm50	s 1	0.1
21	$3.U_3(5)$ *	2	1	AIR ind i126 d2 c28	1d/1	3.5 + 0.1
21	$U_4(3)$ *	1	1	IRR perm112	s 1	0.3
21	$3_1.U_4(3)$	2	1	AIR ind i126 d2 c20	s 1d/1	0.6 + 0.1
21	$3.U_6(2)$ *	2	1	IE i228096 (AIR ind i42 d2)	1d/1	9.4 + 2.0
21	$S_6(2)$	1	1	IRR ind i28 d1 c4	s 1	0.2
21	M_{22} *	1	1	IRR perm21	s 1	0.0
21	$3.M_{22}$ *	1	1	IRR perm22	s 1	0.1
21	J_2	2	1	AIR ind i280 d1 c22	1d/1	0.4 + 0.1
22	$U_6(2)$ *	1	1	IRR perm891 c61	s 1	0.6
22	M_{23}	1	1	IRR perm23	s 1	0.1
22	HS *	1	1	IRR perm100	s 1d	0.3

Deg	Group	C	S	Method	N/D	Time
22	McL *	1	1	IRR perm275 c23	s 2d	0.7
23	M_{24} *	1	1	IRR perm24	s 1	0.0
23	Co_3 *	1	1	AIR ind i276 d1 c12	s 1d	0.3
23	Co_2 *	1	1	AIR ind i2300 d1 c74	s 1d	14.4
24	$3.A_7$	4	1	BB i7 d18 c18 Ri7 $[9,15]_2$	2d/48	0.6 + 0.7
24	$6.A_7$	4	1	BB i21 d8 c8 Ri35 $[12_2^2]$	1d/12	4.9 + 3.5
24	$2.A_8$	2	1	AIR ind i8 d8 c10	1d/1	1.4 + 0.1
24	$12_1.L_3(4)$ *	8	1	IE i21 (AIR ind i18 d8)	3d/320	26 + 3
24	$U_4(2)$	1	1	IRR perm40	s 1	0.1
24	$2.S_4(7)$ *	2	1	AIR ind i800 d6 c80	2d/4	4.5 + 3.4
24	$2.Co_1$ *	1	1	GE i98280 $[1,23]$	1d/1	15 + 4.3
25	$S_4(7)$ *	2	1	IE i1176 (ind i50 d1 c2)	1d/7	1.4 + 0.2
26	$L_3(3)$	1	1	IRR perm39	s 1	0.1
26	$L_3(3)$	2	1	AIR ind i52 d2 c8	s 1d/1	1.1 + 0.1
26	$L_4(3)$ *	1	1	IRR perm117	s 1	0.4
26	$^3D_4(2)$ *	1	1	IRR perm819 c37	s 1d	0.4
26	$^2F_4(2)'$	2	1	IE i1600 Ri8775 $[2_2,8,16]$	1d/8	0.9 + 4.9
27	A_9	1	1	IRR perm36	s 1	0.1
27	$L_3(3)$	1	1	IRR ind i39 d1 c3	s 1	0.1
27	$U_3(3)$	1	1	IRR perm28	s 1	0.1
27	$S_6(2)$	1	1	IRR perm28	s 1d	0.1
27	$3.O_7(3)$ *	2	1	IE i12636 (AIR ind i36 d2)	2d/14	11.5 + 2.9
27	$3.G_2(3)$	2	1	AIR ind i378 d2 c60	1d/1	3.1 + 0.2
27	$^2F_4(2)'$	2	1	AIR perm2304 c144	2d/1	3.0 + 0.6
28	A_8	1	1	IRR perm56	s 1	0.1
28	A_9	1	1	IRR ind i36 d1	1	0.3
28	$2.L_3(4)$	2	1	BB p112 c16 Ri1260 $[1^{4\times1},1^{2+2},2^{4\times1},2^{3+3}]$	1d/4	0.1 + 0.4
28	$4_2.L_3(4)$	4	1	BB p224 c32 Ri105 $[4_2,12_2^2]$	2d/16	2.3 + 1.1
28	$U_3(3)$	2	1	AIR ind i63 d1 c5	1d/1	0.1 + 0.2
28	$U_3(5)$	1	1	IRR perm50	s 1d	0.1
28	$O_8^+(2)$ *	1	1	IRR ind i120 d7 c56	s 1d	0.3
28	2.Ru *	2	1	IE i7238400 (i35 d1 c10)	3d/1	47 + 6.6
30	$L_3(5)$ *	1	1	IRR perm31	s 1	0.1
30	$L_5(2)$ *	1	1	IRR perm31	s 1	0.1
30	$U_4(2)$	1	1	IRR ind i36 d1 c2	s 1	0.1
30	$U_4(2)$	2	1	AIR ind i40 d1 c8	s 1d/1	0.1 + 0.1
31	$L_3(5)$	1	1	IRR perm62	s 1	0.1
31	$L_3(5)$	2	1	AIR perm124	s 1	0.2 + 0.1
32	$2.A_{12}$	1	2	IE i15400 (DI i8 d4)	1d/9	7.9 + 3.5
32	$2.A_{13}$	1	2	IE i1716 (BB Ri2 [32])	3d/357	121 + 4.1
32	$U_3(3)$	2	1	BB i63 d2 c10 Ri224 $[2^{4\times1},6^4]$	2d/9	0.1 + 0.3

Deg	Group	C	S	Method	N/D	Time
32	$2.M_{12}$	1	2	AIR ind i12 d32 c34	s 1d/1	1.5 + 0.7
34	$S_4(4)$	1	1	IRR perm85	s 1d	0.1
34	$O_8^-(2)$ *	1	1	IRR perm119	s 1d	0.3
35	A_7	1	1	IRR perm70 c5	s 1/1	0.2
35	A_8	1	1	IRR ind i56 d1 c6	s 1	0.2
35	A_9	1	1	IRR perm120 c8	1d	0.5
35	A_{10}	1	1	IRR perm45	s 1	0.2
35	$L_3(4)$	1	1	IRR perm56	s 1d	0.1
35	$U_4(3)$	1	1	IRR perm126	s 1d	0.4
35	$S_6(2)$	1	1	IRR perm36	s 1d	0.1
35	$S_6(2)$	1	1	IRR perm120	s 1d	0.4
35	$S_8(2)$ *	1	1	IRR ind i120 d1 c4	s 1d	0.5
35	$O_8^+(2)$	1	1	IRR perm120	s 1d	0.2
35	$Sz(8)$	3	1	BB p520 c12		
				Ri1120 $[1^3, 4_3^{2+3+3}]$	2d/13	0.2 + 0.6
36	$2.A_7$	1	2	BB i15 d16 c36 Ri7 $[16,20]_2$	2d/9	1.1 + 1.0
36	$6.A_7$	2	1	AIR ind i21 d8 c8 (p. 76)	2d/1	2.6 + 0.3
36	A_{10}	1	1	AIR ind 45 d1 c5	s 1	0.2
36	$2.L_3(4)$	1	1	IRR ind i56 d1 c8	s 1	0.1
36	$4_2.L_3(4)$	2	1	AIR ind i120 d2 c36	2d/1	0.8 + 0.3
36	$6.L_3(4)$	2	1	AIR ind i120 d2 c6	1d/1	8.6 + 0.3
36	$12_2.L_3(4)$ *	4	1	BB i120 d4 c8 Ri21 $[16,20]_4$	1d/48	1.6 + 11.0
36	$2.U_4(2)$	2	1	AIR ind i40 d2 c4	s 1/1	1.0 + 0.3
36	$3_2.U_4(3)$ *	2	1	AIR ind i162 d2 c32	1d/1	1.0 + 0.4
36	$12_2.U_4(3)$ *	4	1	IE i162 (ind i3 d72 c36)	2d/4	43.5 + 3.3
36	J_2	1	1	IRR perm100	s 1d	0.3
39	$L_3(3)$	1	1	IRR ind i52 d1 c4	s 1	0.1
39	$L_4(3)$	1	1	IRR perm40	s 1	0.1
39	$U_3(4)$	4	1	BB p208 c5		
				Ri975 $[15:1,3:8]$	1d/8	0.7 + 1.6
40	$2.L_4(3)$ *	1	1	IRR perm80	s 1/1	0.1
40	$U_4(2)$	2	1	AIR ind i45 d2 c4	s 1/1	0.1 + 0.2
40	$S_4(5)$	1	1	IRR ind i300 d1 c26	s 1d	0.5
40	$2.S_4(9)$ *	1	2	IE i3321 (ind i12 d8 c8)	1d/6	17 + 1.3
40	$2.S_8(3)$ *	2	1	IE i39656127420 (i40 d16)	3d/4	464 + 3.5
40	$2.Sz(8)$ *	3	1	BB i65 d8 c11 Ri455 $[8^5]$	1d/8	1.6 + 2.0
41	$S_4(9)$ *	1	1	IRR ind i820 d1 c20	s 1d	4.6
41	$S_8(3)$ *	2	1	GE i39656127420 $[5_2, 36]$	1d/16	9.1 + 5.3
42	A_9	1	1	IRR perm126 c12	s 1	0.5
42	A_{10}	1	1	AIR ind i126 d1	1	1.2
42	$6.L_3(4)$	4	1	BB i120 d2 c6		
				Ri504 $[2_4, 5_2^{2+3+3}]$	2d/32	4.7 + 12.6
42	$U_3(7)$ *	1	2	IE i344 (ind i2 d42 c4)	2d/7	4.2 + 2.4
42	$U_7(2)$ *	1	2	IE i38313, i960 (ind i27 d28)	s 1/2	15 + 17

Deg	Group	C	S	Method	N/D	Time
43	$U_3(7)$	1	1	IRR ind i344 d1 c8	s 1d	0.4
43	$U_3(7)$	2	1	AIR ind i688 d1 c14	2d/1	3.0 + 0.6
43	$U_3(7)$	4	1	AIR ind i344 d4 c28	3d/1	5.1 + 5.3
43	$U_7(2)$	2	1	GE i61997056 $[1, 7_2, 35_2]$	2d/243	85 + 22
44	A_{11}	1	1	IRR perm55	s 1	0.2
44	$U_5(2)$	1	1	IRR perm165 c15	s 1d	0.2
44	M_{11}	1	1	IRR perm55 c3	s 1	0.1
44	$2.M_{12}$	2	1	IE i12 (AIR ind i55 d1 c5)	2d/15	0.3 + 0.4
45	A_8	2	1	DI i15 d3 (AIR ind i21 d1)	s 1d/1	0.6
45	A_{11}	1	1	IRR ind i55 d1 c7	s 1	0.5
45	$L_3(4)$	2	1	BB p280 c8 Ri21 $[15^{3\times1}]$	1d/8	0.3 + 0.9
45	$3.L_3(4)$	4	1	BB i21 d30 c90 Ri63 $[15^{3\times1}]$	2d/32	0.3 + 5.0
45	$U_4(2)$	2	1	AIR ind i40 d3 c10	2d/1	0.2 + 0.8
45	$3_2.U_4(3)$	2	1	IE i567 (i10 d12 c16)	1d/64	4.7 + 1.9
45	M_{11}	1	1	IRR ind i55 d1 c5	s 1	0.1
45	M_{12}	1	1	IRR perm144 c14	s 1d	0.2
45	M_{22}	2	1	IE i77 (AIR ind i10 d6 c8)	1d/8	0.4 + 1.8
45	$3.M_{22}$	2	1	IE i77 (AIR ind i60 d1 c8)	1d/16	4.1 + 1.1
45	$3.M_{22}$	4	1	IE i77 (AIR ind i60 d2 c16)	2d/64	7.6 + 2.0
45	M_{23}	2	1	IE i253 (AIR ind i30 d6 c18)	1d/4	1.1 + 0.4
45	M_{24}	2	1	IE M_{23}	1d/4	+0.3
48	$2.A_8$	1	2	BB i8 d40 c24 Ri15 [48]	1/4	0.5 + 2.9
48	A_9	1	1	IRR perm84	s 1	0.1
48	$2.A_9$	2	1	BB i840 d1 c56 Ri120 [21, 27]	2d/168	1.1 + 6.2
48	$2.A_{10}$	2	1	BB i10 d96 c52 Ri945 [24, 24]	2d/80	2.9 + 61
48	$12_1.L_3(4)$	8	1	BB i56 d24 c192 Ri105 $[8_4^{3\times2}]$	4d/192	8.7 + 23.5
48	$12_2.L_3(4)$	8	1	BB i56 d24 c192 Ri112 $[12_2, 12_4, 12_4]$	3d/960	2.3 + 35.4
48	$3.U_3(5)$	2	1	AIR ind i50 d12 c64	1d/1	1.6 + 0.6
48	$2.S_6(2)$	1	1	IRR ind i28 d8 c22	s 1	0.2
50	$S_4(4)$	1	1	IRR perm85	s 1	0.1
50	$O_8^+(2)$	1	1	IRR perm135	s 1	0.3
50	$2.J_2$	2	1	AIR perm200 c16	1d/1	0.3 + 1.0
51	$U_4(4)$ *	4	1	IE i1040 (i120 d1)	2d/10	5 + 21
51	$S_4(4)$	2	1	AIR ind i120 d1 c8	s 1d/1	1.8 + 0.6
51	$S_8(2)$	1	1	IRR ind i136 d1 c4	s 1d	0.2
51	$O_8^-(2)$	1	1	IRR perm136	s 1d	0.4
51	He *	2	1	BB p2058 c80 Ri187425 $[6, 21, 24_2]$	1d/8	9.8 + 8.2
52	$L_4(3)$	1	1	IRR ind i117 d1 c5	s 1d/1	0.2
52	$U_3(4)$	4	1	AIR ind i64 d4 c16	s 1d/1	1.0 + 3.4
52	$U_4(4)$	1	1	IRR perm325 c9	s 1d	8.8
52	$2.S_4(5)$	2	2	BB i156 d8 c80 Ri156 $[12, 40]_4$	2d/25	1.3 + 30.8

Deg	Group	C	S	Method	N/D	Time
52	$^3D_4(2)$	1	1	RR $(\rho_{26})^2$ c20	1d	18
52	$2.F_4(2)$	1	1	GE i139776 $[S_8(2)]$: [1, 51]	1d	32 + 20
54	A_{12}	1	1	IRR perm66	s 1d	0.3
54	M_{12}	1	1	IRR perm66	s 1d	0.1
55	A_{12}	1	1	IRR ind i66 d1 c6	s 1	0.7
55	$U_5(2)$	1	1	IRR perm176 d16	1d	0.5
55	$U_5(2)$	2	1	AIR ind i165 d2 c20	s 1d/1	0.7 + 0.2
55	M_{12}	1	1	ExteriorSquare(ρ_{11})	s 1d	0.0
55	M_{11}	1	1	IRR ind i66 d1 c6	s 1	0.1
55	M_{12}	1	1	IRR ind i66 d1	s 1d	0.1
55	M_{22}	1	1	IRR perm77	s 1d	0.1
56	A_8	1	1	IRR ind i35 d2 c4	s 1	0.4
56	$2.A_8$	2	1	AIR ind i28 d8 c8	2d/1	2.8 + 0.6
56	$2.A_8$	2	1	BB i15 d8 c3 Ri15 $[8, 16]$	2d/48	0.5 + 1.9
56	A_9	1	1	IRR ind i84 d1 c10	s 1	0.4
56	$2.A_9$	1	1	IRR ind i9 d8 c8	1d	1.8
56	$4_1.L_3(4)$	2	1	AIR perm224 c32	s 1d/1	0.3 + 1.0
56	$L_3(7)$ *	1	1	IRR perm57	s 1	0.1
56	$U_3(8)$ *	1	2	IE i513 (i3 d112 c16)	2d/4	42 + 22
56	$2.U_4(3)$	1	1	IRR ind i112 d1 c8	s 1d	0.5
56	$2.U_6(2)$ *	1	2	IE i20736 (ind i176 d1 c16)	1d	5.3 + 14.0
56	$S_6(2)$	1	1	IRR ind i63 d1 c7	s 1	0.1
56	$2.O_8^+(2)$	1	1	IRR ind i120 d8 c64	1d	1.6
56	$2.Sz(8)$	3	1	BB i65 d8 c11 Ri455 $[8^7]$	3d/40	1.8 + 3.2
56	$2.M_{22}$	1	1	IRR ind i176 d1 c16	1d	1.0
56	$4.M_{22}$ *	4	1	IE i22 (DI i56 d1)	1d/12	0.7 + 1.4
56	J_1 *	2	1	BB p2665 c14 Ri1463 $[1^{1+1}, 3_2^{1+1+2+2}, 4^{2+2}, 5^{2+2}]$	2d/120	0.2 + 0.5
56	$2.J_2$	2	2	BB i315 d8 c180 Ri100 $[56]$	2d/9	0.5 + 6.3
56	$2.HS$	1	1	IRR ind i100 d56 c272	1d	11.9
57	$L_3(7)$	1	1	IRR perm114	s 1	0.2
57	$3.L_3(7)$	2	1	DI i57 d1	s 1	0.2
57	$U_3(8)$	2	1	AIR ind i513 d2 c48	2d/1	3.8 + 0.6
57	$3.U_3(8)$ *	6	1	BB i513 d6 c162 Ri3648 $[7^{5\times1+2}, 8]_6$	2d/36	7 + 220
60	$6.L_3(4)$	4	1	BB i21 d12 c36 Ri210 $[4^{1\times3}, 6^{1+2+2+3}]_2$	2d/48	2.3 + 20.3
60	$12_2.L_3(4)$	8	1	BB i56 d24 c192 Ri56 $[6, 6, 9^2, 15^3]_4$	4d/3200	8.9 + 123.4
60	$U_4(2)$	1	1	IRR perm120	s 1d	0.1
60	$2.U_4(2)$	1	1	IRR perm120	s 1d	0.2
60	$2.U_4(2)$	1	2	AIR ind i40 d4 c8	s 1d/1	0.4 + 0.6
60	$2.U_4(2)$	2	1	AIR ind i40 d4 c16	s 1d/1	0.7 + 0.6
60	$U_5(3)$ *	1	2	IE i81984 (i360 d1 c36)	1d/9	187 + 64

Deg	Group	C	S	Method	N/D	Time
60	$2.S_4(11)$ *	2	1	IE i7381 (p2640, Ri144)	2d/110	596 + 40
61	$U_5(3)$	1	1	IRR perm13664 c176	1d	8.6
61	$U_5(3)$	2	1	GE i4941 $[20,20,21]_2$	2d/4	1755 + 10
61	$S_4(11)$ *	2	1	IE i7260 (ind i122 d1)	2d/11	10.0 + 2.4
62	$L_6(2)$	1	1	IRR perm63	s 1d	0.1
62	$2.S_6(5)$ *	2	1	IE i78000000 (ei 126, 3)	s 1d/5	13 + 5.3
63	$L_3(4)$	2	1	AIR perm252	s 1d/1	0.2 + 0.6
63	$3.L_3(4)$	2	1	AIR perm252	s 1d/1	0.7 + 0.6
63	$3.L_3(4)$	4	1	AIR ind i63 d6 c18	s 1d/1	3.1 + 8.3
63	$S_6(5)$ *	2	1	IE i377812500 $[J_2]$	2d/10	0.3 + 5.3
63	J_2	1	1	IRR perm100	s 1d	0.3
64	A_8	1	1	IRR ind i56 d2 c10	s 1	0.5
64	$2.A_8$	1	2	AIR ind i15 d16 c16	s 1d/1	1.5 + 1.0
64	$2.A_{10}$	1	1	IRR ind i10 d8 c10	s 1d	3.3
64	$2.A_{14}$	1	2	IE i135135 (ind i64 d2 c4)	s 1d/8	5.4 + 3
64	$2.A_{15}$	2	1	IE i1401400 (IE i10)	2d/54	8.5 + 2.3
64	$L_3(4)$	1	1	IRR ind i21 d4 c12	s 1d	0.1
64	$2.L_3(4)$	1	1	IRR ind i21 d4 c12	s 1d	0.4
64	$4_1.L_3(4)$	2	1	AIR ind i120 d2 c36	2d/1	1.9 + 1.0
64	$4_2.L_3(4)$	2	1	AIR ind i21 d8 c24	s 1d/1	2.9 + 0.9
64	$U_3(4)$	1	1	IRR perm64	s 1d	0.2
64	$U_4(2)$	1	1	IRR ind i45 d8	s 1d	0.2
64	$2.U_4(2)$	1	2	AIR ind i40 d6 c12	2d/1	3.8 + 1.1
64	$2.S_6(2)$	1	2	IE i135 $((\rho_8)^2)$	2d/2d	2.4 + 25.0
64	$Sz(8)$	1	1	IRR perm65	s 1d	0.0
64	$2.Sz(8)$	1	1	IRR ind i65 d8 c11	s 1d	1.7
64	$G_2(3)$	2	1	IE i351 (i63 d2 c4)	2d/21	14 + 14
64	$2.J_2$	2	2	AIR ind i315 d8 c344	3d/1	20.9 + 23.9
65	A_{13}	1	1	IRR perm78	s 1d	0.1
65	$L_4(3)$	1	1	IRR ind i117 d1 c5	s 1d	0.2
65	$U_3(4)$	1	1	IRR ind i65 d2 c10	s 1d	0.2
65	$U_3(4)$	4	1	AIR ind i65 d8 c32	3d/1	8.2 + 16.1
65	$S_4(5)$	1	1	IRR perm156	s 1d	0.1
65	$Sz(8)$	3	1	BB p455 c14		
				Ri65 $[2_3, 7, 28^2]$	2d/16	0.2 + 0.6
65	$G_2(4)$ *	1	1	IRR perm416	s 1d	1.0
66	A_{13}	1	1	ExteriorSquare(ρ_{12})	s 1d	0.0
66	$U_5(2)$	2	1	AIR ind i172 d2 c22	s 1d/1	5.1 + 0.9
66	M_{12}	1	1	IRR perm132	s 1d	0.2
66	$6.M_{22}$ *	4	1	GE i77 $[30,36]_2$ (p. 110)	2d/32	3.4 + 2.8
66	$3.Suz$ *	2	1	IE i2358720 (i3 d66)	2d/12	15.2 + 7.8
70	A_8	1	1	DI i35 d2	1d	0.5
70	$2.L_3(4)$	1	1	IRR ind i21 d6	1d	0.3
70	$2.U_4(3)$	1	1	IRR ind i126 d1	1d	2.7

Deg	Group	C	S	Method	N/D	Time
70	$2.U_4(3)$	2	1	AIR ind i126 d6	3d/1	13.0 + 2.0
70	$S_6(2)$	1	1	IRR ind i336 d1	1d	3.4
70	J_2	2	1	BB i525 d1 c39 Ri100 $[14, 56]$	2d/27	0.4 + 1.3
72	$L_3(8)$ *	1	1	IRR perm73	1d	0.2
72	$U_3(9)$ *	1	2	IE i730 (ind i8, d18)	2d/9	20.9 + 5.8
73	$L_3(8)$	6	1	AIR perm511	1d/1	5.3 + 31.9
73	$U_3(9)$	1	1	IRR ind i730 d1 c10	1d	3.8
73	$U_3(9)$	4	1	BB i730 d4 c36 Ri730 $[2_4, 72_4]$	2d/9	3.2 + 12.6
75	A_{10}	1	1	IRR perm120	1d	1.1
75	$U_3(4)$	4	1	BB i416 d1 c30 Ri65 $[12_2, 15, 48]$ (p. 140)	3d/240	3.0 + 1.9
76	J_1	1	1	IRR perm266	1d	0.4
77	A_{14}	1	1	IRR perm91	s 1	0.2
77	J_1	1	1	IRR perm266	1d	0.3
77	J_1	2	1	BB p1045 c11 Ri1463 $[1, 3_2^{3+3}, 4^{3+3}, 5^{2+3}, 6^2]$	2d/120	0.5 + 1.4
77	HS	1	1	IRR perm100	1d	0.3
78	A_{14}	1	1	ExteriorSquare(ρ_{13})	s 1	0.0
78	$S_4(5)$	2	1	BB p312 c24 Ri156 $[1, 3_2, 20_2, 24, 30_2]$	2d/50	5.6 + 2.8
78	$S_6(3)$	1	1	RR $(\rho_{26})^2$ c20	2d/1	+1.9
78	$O_7(3)$ *	1	1	IRR ind i351 d1	1d	4.0
78	$G_2(3)$	1	1	IRR ind i351 d1	1d	1.9
78	$G_2(4)$	1	1	IRR ind i2080 d1 c100	1d	6.6
78	$^2F_4(2)'$	1	1	IRR perm1755 c117	1d	3.9
78	3.Suz	2	1	IE i1782 (i3 d78)	3d/4	52.8 + 6.0
78	Fi_{22} *	1	1	IE $^2F_4(2)'$	1d/1	+2.0
80	$4_1.L_3(4)$	4	1	BB p1344 c36 Ri56 $[4, 4, 16^2, 20^2]_2$	3d/180	2.4 + 4.4
80	$4_2.L_3(4)$	4	1	BB i56 d8 c24 Ri105 $[4, 8^2, 12^5]_2$	2d/96	2.3 + 15.3
80	$2.U_4(2)$	1	2	DI i40 d2	s 1d/1	0.7
81	$U_4(2)$	1	1	IRR perm160	s 1d	0.1
84	A_9	1	1	IRR ind i120 d1 c8	s 1	0.1
84	A_{10}	1	1	IRR ind i120 d1 c10	s 1	0.6
84	$3.L_3(4)$	2	1	AIR ind i21 d10 c18	1d/1	3.2 + 1.4
84	$12_2.L_3(4)$	4	1	BB i120 d4 c8 Ri105 $[4^{2+2+2}, 12^5]_4$	2d/48	36 + 9.8
84	$L_4(4)$ *	1	1	IRR perm85	1d	2.9
84	$U_3(5)$	1	1	IRR perm525	1d	1.0
84	$3.U_3(5)$	2	1	AIR ind i525 d2 c122	2d/1	12.9 + 1.8
84	$6_1.U_4(3)$	2	1	AIR $\rho_{12} \otimes \rho_{30}$ c8	2d/1	14 + 1.7

Deg	Group	C	S	Method	N/D	Time
84	$12_1.U_4(3)$ *	4	1	IE $12_2.L_3(4)$ Ri112 $[24,60]_4$	2d/27	+12 + 112
84	$2.S_4(13)$ *	2	2	IE i14365 (DI i2 d42)	4d/702	73 + 4.8
84	$S_6(2)$	1	1	IRR perm120	s 1d	0.9
84	$O_8^+(2)$	1	1	IRR perm120	s 1d	0.4
84	$O_8^-(2)$	1	1	IRR perm119	s 1d	0.4
84	$2.J_2$	1	2	AIR ind i100 d12 c72	2d/1	7.9 + 10.5
84	$L_4(4)$	1	1	IRR perm85	s 1d	0.7
85	$L_4(4)$	2	1	AIR perm255 c3	s 1/1	3.0 + 1.6
85	$U_8(2)$ *	2	1	IE i3766321152 (ei 1008)	2d/12	5 + 10
85	$S_4(4)$	1	1	IRR perm120	s 1d	0.3
85	$S_4(13)$ *	2	1	IE i14196 (i85 d2 c2)	2d/13	85 + 4.6
85	$S_8(2)$	1	1	IRR ind i120 d1 c8	s 1d	2.2
85	J_3 *	2	1	BB p14688 c186		
				Ri6156 [17, 68] (p. 138)	2d/120	0.7 + 49
86	$U_8(2)$	1	1	GE i1844412416 [36, 50]	2d	40 + 8
90	A_{10}	1	1	IRR perm126	s 1d	0.9
90	A_{15}	1	1	IRR perm105	s 1	1.1
90	$2.L_3(4)$	1	1	IRR ind i21 d6 c18	s 1d	0.4
90	$6.L_3(4)$	2	1	AIR ind i21 d12 c36	1d/1	15.8 + 1.3
90	$L_3(9)$ *	1	1	IRR perm91	s 1d	2.1
90	$L_4(3)$	1	1	IRR perm117	s 1d	1.2
90	$U_4(3)$	1	1	IRR perm112	s 1d	1.2
90	$6_2.U_4(3)$ *	2	1	IE i112 (DI i90 d1)	1d/9	1.7 + 1.8
90	$S_4(5)$	1	1	IRR perm156	s 1d	1.3
90	J_2	1	1	IRR perm280	s 1d	0.6
91	A_{15}	1	1	ExteriorSquare(ρ_{14})	s 1	0.0
91	$L_3(9)$	1	1	IRR perm182	s 1d	1.6
91	$L_3(9)$	2	1	DI i91 d1	s 1/1	0.2
91	$L_3(9)$	4	1	DI i91 d1	s 1/1	0.2
91	$S_6(3)$	2	1	RR $(\rho_{26})^2$ c12	2d/1	+3.8 + 2.8
91	$O_7(3)$	1	1	IRR ind i364 d1 c20	s 1d	5.6
91	$Sz(8)$	1	1	IRR perm520 c12	s 1d	0.6
91	$G_2(3)$	1	1	IRR perm364 c28	s 1d	0.4
96	$L_3(5)$	10	1	IE i31 (DI i24 d4) (p. 89)	5d/$(5^5 \cdot 67)$	0.2 + 11.9
96	$3.L_3(7)$	2	1	IE i57 (DI i16 d6)	1d/7	22.0 + 3.1
99	M_{12}	1	1	IRR perm220 c20	s 1d	0.4
99	M_{22}	1	1	IRR perm330 c30	s 1d	0.7
99	$3.M_{22}$	2	1	AIR ind i22 d30 c60	3d/1	17.7 + 3.4
104	A_{16}	1	1	IRR perm120	s 1	0.2
104	$U_4(5)$ *	2	1	IE i1575 (ind i2, d104)	3d/3	9.1 + 3.5
104	$2.U_4(5)$ *	1	2	IE i1575 (ind i312, d8)	3d/2	123 + 61
104	$2.U_4(5)$	2	1	IE i1575 (ind i156, d8)	3d/1	99 + 23
104	$S_4(5)$	1	1	IRR ind i156 d4 c20	1d	1.0
104	$2.S_4(5)$	1	2	AIR ind i156 d8 c40	3d/1	32.2 + 5.2

Deg	Group	C	S	Method	N/D	Time
104	$2.Sz(8)$	1	1	IRR ind i520 d2 c11	s 1d	1.4
104	$G_2(3)$	1	1	IRR perm364	s 1d	2.0
104	$2.G_2(4)$	2	2	IE i2016 (DI i2 d52)	2d/10	29.1 + 14.7
105	A_9	1	1	IRR ind i84 d2 c18	s 1d	0.8
105	A_{16}	1	1	ExteriorSquare(ρ_{15})	s 1	0.0
105	$U_3(5)$	1	1	IRR ind i126 d1	s 1d	0.2
105	$3.U_3(5)$	2	1	AIR ind i126 d2	s 1d/1	4.1 + 3.7
105	$3_1.U_4(3)$	2	1	AIR ind i126 d2	s 1d/1	8.5 + 3.9
105	$U_4(5)$	1	1	IRR perm756	s 1d	11.9
105	$S_6(2)$	1	1	IRR ind i28 d6	s 1d	1.5
105	$S_6(3)$	1	1	IRR perm1120	s 1d	1.7
105	$O_7(3)$	1	1	IRR perm756	s 1d	2.2
105	$3.M_{22}$	4	1	BB i231 d2 c20		
				Ri22 $[21, 84]_2$	3d/1440	1.9 + 6.5
110	A_{11}	1	1	IRR perm165 c15	s 1	1.1
110	$U_3(11)$ *	1	2	IE i1332 ($\rho_1 \otimes \rho_{110}$)	3d/11	23 + 5.5
110	$U_5(2)$	1	2	AIR ind i165 d16 c128	3d/1	18.9 + 4.3
110	$U_5(2)$	2	1	AIR perm495 c35	s 1d/1	3.4 + 1.5
110	$2.M_{12}$	2	1	BB i12 d20 c20		
				Ri12 $[10_2, 45, 55]$	2d/44	0.8 + 3.2
111	$U_3(11)$	1	1	IRR ind i1332 d1 c32	s 1d	6.6
111	$U_3(11)$	2	1	BB i1332 d2 c66		
				Ri5328 $[1, 110]$	2d/11	8.4 + 11.9
111	$3.U_3(11)$ *	2	1	BB i3996 d1 c186		
				Ri5328 $[1_2, 110_2]$	2d/121	52 + 24
111	$3.U_3(11)$	4	1	BB i1332 d4 c132		
				Ri5328 $[1_2, 110_2]$	2d/11	62 + 40
112	$2.A_9$	1	1	IRR ind i120 d1 c8	s 1d	0.8
112	$2.S_6(2)$	1	1	IRR ind i120 d1 c14	s 1d	0.3
112	$2.O_8^+(2)$	1	1	IRR ind i120 d1 c16	s 1d	0.4
119	A_{17}	1	1	IRR perm136	s 1	1.3
119	$S_8(2)$	1	1	IRR perm120	s 1	0.3
120	A_9	1	1	IRR perm280 c20	s 1d	0.3
120	$2.A_9$	2	1	DI i120 d1	s 1	0.1
120	A_{11}	1	1	IRR ind i165 d1	s 1d	0.4
120	A_{17}	1	1	ExteriorSquare(ρ_{16})	s 1	1.3
120	$12_1.L_3(4)$	4	1	AIR ind i360 d2 c8	s 1/1	41 + 17
120	$L_5(3)$ *	1	1	IRR perm121	s 1	0.4
120	$2.U_4(3)$	1	1	IRR ind i540 d1 c56	1d	1.7
120	$4.U_4(3)$	2	1	AIR ind i540 d2 c104	3d/1	17 + 4.5
120	$6_1.U_4(3)$	2	1	AIR ind i126 d2 c16	s 1d/1	2.2 + 3.1
120	$12_1.U_4(3)$ *	4	1	BB i540 d4 c160		
				Ri112 $[60, 60]_4$	1d/54	98 + 75
120	$U_5(2)$	1	1	IRR perm165	1	0.4

Deg	Group	C	S	Method	N/D	Time
120	$6.U_6(2)$ *	2	1	IE $6.M_{22}$	2d/84	+5.4
120	$S_6(2)$	1	1	IRR perm288	s 1d	0.6
120	$2.S_6(2)$	1	1	IRR perm240	s 1d	0.2
120	$2.S_6(2)$	1	1	IRR perm288	s 1d	0.3
120	M_{12}	1	1	IRR ind i220 d1 c20	s 1d	0.8
120	$2.M_{12}$	1	1	IRR ind i12 d11 c12	s 1d	0.9
120	$2.M_{12}$	1	1	IRR ind i220 d1 c20	s 1d	0.6
120	$2.M_{22}$	1	1	IRR ind i176 d1 c16	s 1d	2.2
120	$6.M_{22}$	2	1	BB i672 d2 c124		
				Ri22 $[120_2]$	1d/8	1.2 + 37
120	$12.M_{22}$ *	8	1	IE i22 (i360 d2 c64)	2d/84	142 + 167
120	J_1	3	1	BB i1463 d1 c77 Ri266		
				$[10^{1+2+2}, 11^2, 24^2]$	3d/660	0.6 + 3.7
121	$L_5(3)$	1	1	IRR perm242 c2	s 1	1.9
121	$S_{10}(3)$ *	2	1	GE$[\neg\chi]$ i74032324732080		
				$[60_2, 61]$	2d/72	1100 + 11
122	$2.S_{10}(3)$ *	2	1	G[I]E$[\neg\chi]$ i74032324732080	2d/54	2311 + 23
124	$L_3(5)$	1	1	DI i31 d4	s 1	0.2 + 0
124	$L_3(5)$	2	1	DI i31 d4	s 1/1	0.5 + 0
124	$L_3(5)$	4	1	DI i31 d4	s 1d/3	2.0 + 0
124	$L_5(2)$	1	1	IRR perm155 c2	s 1	0.2
124	$Sz(32)$ *	2	1	IE i1025 (DI 124 d1)	1d/16	2.0 + 3.9
124	$G_2(5)$ *	1	1	IRR ind i3906 d4 c504	2d	108
125	$L_3(5)$	1	1	IRR ind i31 d5 c5	s 1d	0.2
125	$U_3(5)$	1	1	IRR perm126	s 1d	0.1
126	A_{10}	1	1	IRR ind i210 d1 c14	s 1	2.3
126	A_{11}	2	1	IE i11 (IRR ind i210 d1)	1d/10	2.5 + 2.2
126	$L_7(2)$ *	1	1	IRR perm127	s 1	1.1
126	$U_3(5)$	1	1	IRR ind i175 d1 c19	s 1d	0.2
126	$U_3(5)$	2	1	BB i126 d4 c48 Ri525		
				$[1, 4^{1+3}, 4_2^2, 5^{2+3}, 6^3, 6_2^{3+4}, 8^2]$	2d/720	0.7 + 6.3
126	$3.U_3(5)$	2	1	BB i126 d4 c56 Ri525		
				$[1, 4, 4^{3\times2}, 4^3, 5^{2+3}, 6^4, 12^3]_2$	2d/240	2.1 + 5.8
126	$3.U_3(5)$	4	1	BB i126 d8 c96 Ri525 $[1_2,$		
				$4_2^{1+2+2+2+3}, 5_2^{1+5}, 6_2^3, 6_4^{3+4}]$	3d/480	3.6 + 7.0
126	$3_2.U_4(3)$	2	1	AIR perm378 c36	s 1/1	1.5 + 3.5
126	$6_1.U_4(3)$	2	1	AIR ind i378 d2 c28	s 1/1	3.5 + 3.6
126	$6_2.U_4(3)$ *	2	1	DI i126 d1	s 1/1	0.1
126	$6_2.U_4(3)$	4	1	BB i540 d14 c264		
				Ri112 $[36, 90]_2$	1d/27	3.4 + 17
126	$S_4(7)$	1	1	IRR ind i1176 d1 c50	1d	3.5
126	$2.M_{22}$	2	1	BB i77 d6 c42 Ri22 [36, 90]	2d/56	1.5 + 6.0
126	$6.M_{22}$	4	1	BB i77 d12 c84 Ri22 $[36, 90]_2$	4d/336	20.1 + 51.2
126	J_2	1	1	IRR perm280 c22	s 1	0.4

Deg	Group	C	S	Method	N/D	Time
126	2.J_2	2	2	BB i100 d12 c72		
				Ri100 $[6_2, 56, 64]$	3d/189	2.8 + 6.2
126	3.McL	4	1	IE i275 $[3_2.U_4(3)]$	2d/324	5 + 67
128	2.A_{16} *	1	1	IE i2627625 (IE i81, DI i8)	2d/1	223 + 5
128	2.A_{17} *	2	1	IE i24310 (IE i29400, DI i64)	2d/6	3150 + 15
130	$S_4(5)$	1	1	IRR ind i156 d1 c8	s 1d	0.3
132	A_{11}	1	1	IRR perm462 c30	s 1d	1.4
132	A_{12}	1	1	IRR perm462 c14	s 1	1.1
132	$L_3(11)$ *	1	1	IRR perm133	s 1d	0.6
133	$L_3(11)$	1	1	DI i133 d1	s 1	0.4
133	$L_3(11)$	4	1	DI i133 d1	s 1/1	0.4
133	$U_3(8)$	1	1	IRR perm3648 c180	s 2d	8.3
133	J_1	1	1	IRR perm1045 c11	2d	9.5
133	J_1	2	1	BB p1596 c19 Ri1463		
				$[1, 3_2^3, 3_2^4, 4^4, 4^4, 5^5, 5^6, 6^4]$	2d/120	0.2 + 3.5
133	HN *	2	1	GE A_{12} [1, 132] (p. 122)	2d/320	1.1 + 15
135	A_{18}	1	1	IRR perm153 c15	s 1	0.5
135	$S_8(2)$	1	1	IRR perm136 c20	s 1	0.6
136	A_{18}	1	1	IRR perm306 c28	s 1	1.1
140	$U_4(3)$	1	1	IRR perm162 c18	s 1d	0.3
140	4.$U_4(3)$	2	1	BB i540 d2 c104		
				Ri112 $[20, 60, 60]_2$	1d/54	122 + 15
143	Suz *	1	1	GE i1782 [65, 78] (p. 111)	2d	10 + 3.2
144	2.A_{11}	1	1	IRR ind i11 d16 c4	1d	304
144	$U_3(5)$	2	1	BB p750 c15		
				Ri126 $[8^{3\times1}, 20^3, 20_2^3]$	2d/500	7.0 + 3.8
144	3.$U_3(5)$	4	1	BB i50 d30 c156 Ri126		
				$[8_2^{1+1+2}, 20_2^{1+3}, 32]$	2d/500	8.0 + 8.5
144	2.$S_4(17)$ *	2	2	IE i41616 (IE i290)	4d/2890	159 + 32
144	M_{12}	1	1	IRR perm396 c10	2d	0.9
144	4.M_{22}	4	1	BB i22 d128 c256		
				Ri77 $[64_4, 80_4]$	2d/144	11 + 134
144	12.M_{22}	8	1	BB i22 d192 c384		
				Ri77 $[24_8, 120_4]$	4d/33600	17 + 370
145	$S_4(17)$ *	2	1	IE i41616 (IE i2)	3d/17	27 + 17
150	$S_4(7)$	2	1	BB i400 d6 c40		
				Ri1176 [50, 100]	2d/42	1.7 + 13
152	A_{19}	1	1	IRR perm171 c19	s 1	0.3
152	$L_3(7)$	1	1	IRR ind i57 d8 c24	1d	2.1
153	A_{19}	1	1	ExteriorSquare(ρ_{18})	s 1	0.0
153	$S_4(4)$	1	1	IRR perm1360 c80	s 2d	7.5
153	3.J_3	4	1	BB $(\rho_{72})^2$ c112		
				Ri17442 $[3_4, 15_2, 45_2^3]$	2d/32	4.7 + 43
153	He	2	1	IE i2058 (IE i2 [$S_4(4)$])	3d/8	+9.9

Deg	Group	C	S	Method	N/D	Time
154	A_{12}	1	1	IRR perm220 c12	s 1	1.5
154	$O^-_{10}(2)$ *	1	1	IRR perm495 c15	s 1d	2.2
154	M_{22}	1	1	IRR perm176 c16	s 1d	0.3
154	$2.M_{22}$	1	1	BB i330 d1 c30 Ri22 $[64, 90]$	2d/21	2.3 + 4.4
154	HS	1	1	IRR perm1110 c62	1d	3.2
155	$L_3(5)$	1	1	IRR ind i31 d6 c6	s 1d	0.4
155	$L_3(5)$	2	1	DI i31 d5 (AIR ind i12 d1)	s 1/1	1.6 + 0.1
155	$L_4(5)$ *	1	1	IRR perm156	s 1	0.3
155	$L_5(2)$	1	1	IRR perm310 c10	s 1	0.2
155	$S_{10}(2)$ *	1	1	IRR perm992 c32	s 1d	1.8
155	$O^+_{10}(2)$ *	1	1	IRR perm496	s 1d	2.7
156	$2.L_4(5)$ *	1	1	IRR perm312 c8	s 1	2.1
156	$4.L_4(5)$ *	2	1	IE i1550 (IE i2)	s 1/1	16.1 + 4.5
156	$U_3(13)$ *	1	2	IE i2198 (IE i7)	3d/507	62 + 56
156	$S_4(5)$	1	1	IRR ind i300 d1 c8	s 1d	0.7
156	$2.S_4(5)$	1	2	AIR ind i156 d2 c20	s 1/1	12.9 + 4.6
157	$U_3(13)$	1	1	IRR perm4396 c28	s 2d	24
157	$U_3(13)$	6	1	BB p15386 c94 Ri2198 $[1, 156_6]$ (p. 148)	3d/169	29 + 75
160	$2.A_9$	1	1	IRR ind i36 d8 c8	s 1d	0.9
160	A_{10}	1	1	IRR ind i120 d2 c18	s 1d	3.1
160	$2.A_{12}$	2	1	IE $2.M_{12}$	3d/11164	+34.3
160	$2.O^+_8(2)$	1	1	IRR ind i120 d7 c80	1d	2.9
160	$2.M_{12}$	2	1	BB i12 d32 c34 Ri220 $[2, 3, 3, 4, 4, 8^{3+3}, 16^6]$	2d/1296	1.7 + 6.4
160	$4.M_{22}$	4	1	BB i32 d77 c224 Ri77 $[15, 64, 80]_2$	3d/1080	17 + 234
160	J_2	1	1	IRR perm316 c27	s 1d	0.5
162	A_9	1	1	IRR ind i9 d28 c16	1d	2.5
165	A_{11}	1	1	IRR perm330 c24	s 1	1.6
165	A_{12}	1	1	IRR ind i220 d1 c12	s 1	2.8
165	$U_5(2)$	1	1	IRR ind i165 d6 c48	2d	10.2
168	A_9	1	1	IRR ind i9 d42 c22	1d	4.5
168	$2.A_9$	2	1	BB i120 d7 c56 Ri120 $[7, 8, 14, 16, 21^2, 27^3]$	2d/189	0.9 + 7.5
168	$S_6(2)$	1	1	IRR perm315 c23	s 1	0.7
168	$2.S_6(2)$	1	1	IRR ind i28 d8 c8	2d	1.7
168	$S_6(3)$	1	1	IRR perm364 c26	s 1d	4.5
168	$O_7(3)$	1	1	IRR perm351 c23	s 1d	1.7
168	$G_2(3)$	1	1	IRR perm351 c27	s 1d	1.0
170	A_{20}	1	1	IRR perm190 c16	s 1	0.8
170	$U_9(2)$ *	1	2	GE$[\neg\chi]$ J_3 $[85, 85]_2$ (p. 114)	4d/36480	+85
171	A_{20}	1	1	IRR perm380 c30	s 1	1.2

Deg	Group	C	S	Method	N/D	Time
171	$3.U_9(2)$ *	2	1	G[I]E[$\neg\chi$] $3.J_3$ [171_2] (p. 113)	2d/480	+69
171	$S_6(7)$ *	2	1	IE i4517721600 (i344, i3, BB i19 d18 Ri19 [$9_2^{19\times1}$])	2d/98	73 + 261
171	$3.J_3$	2	1	BB $(\rho_{72})^2$ c80 Ri17442 $[1,1,4,5^{1+2},15^{1+1+2},45^2]_2$	2d/480	4.5 + 36
171	$3.J_3$	4	1	BB i6156 d34 c876 Ri17442 $[3_4,3_4,15_2,15_2,45_2^3]$ (p. 143)	2d/64	7.3 + 853
172	$2.S_6(7)$ *	2	1	IE i4517721600 (i3, BB p688 c40 Ri6536 [$1,9_2^{19\times1}$])	2d/98	48 + 91
175	$S_4(7)$	1	1	IRR perm400 c10	2d	2.4
175	$O_8^+(2)$	1	1	IRR perm960 c64	1d	3.3
175	J_2	1	1	IRR ind i280 d1 c22	s 1d	1.8
175	HS	1	1	IRR perm176 c12	s 1	0.8
176	$U_5(2)$	1	1	IRR perm297 c21	s 1d	1.3
176	$2.U_6(2)$	1	1	IRR ind i1408 d1 c94	2d	11.8
176	M_{12}	1	1	IRR ind i12 d32 c34	2d	5.1
176	$4.M_{22}$	2	2	DI i22 d8 (AIR ind i21 d32)	s 2d/1	72.6
176	2.HS	2	1	AIR perm704 c36	s 1/1	1.2 + 4.0
180	$2.S_4(19)$ *	2	1	IE i65341 (BB p13680 c94 Ri14400 [$9_2^{1+1+9\times2}$])	3d/76	96 + 55
181	$S_4(19)$ *	2	1	IE i64980 (i2 d181 c2)	3d/19	65 + 26
182	$L_3(13)$ *	1	1	IRR perm183 c3	s 1	1.5
182	$U_6(3)$ *	1	2	IE i27328 (DI i91 d2)	s 1/3	1.3 + 5.6
182	$2.U_6(3)$ *	2	1	IE i4980528 [$2.S_6(3)$]	2d/32	+102
182	$2.S_6(3)$	1	2	$\rho_{13}\otimes\rho_{14}$	1d/1	62
182	$2.S_6(3)$	2	1	BB p728 c42 Ri364 $[1,20_2,36_2,45_2,80]$	2d/81	18 + 6.6
182	$O_7(3)$	1	1	IRR perm351 c23	s 1d	1.4
182	$G_2(3)$	1	1	IRR perm351 c27	s 1d	1.3
183	$L_3(13)$	1	1	IRR perm366 c6	s 1	1.7
183	$L_3(13)$	2	1	AIR perm732 c12	s 1/1	9.9 + 5.8
183	$3.L_3(13)$	2	1	DI i183 d1 c18	s 1/1	3.6
183	$3.L_3(13)$	4	1	DI i183 d1 c18	s 1/1	3.7
183	$U_6(3)$	1	1	IRR perm27328 c568	2d	53
186	$L_3(5)$	1	1	IRR ind i310 d1 c10	s 1d	0.4
186	$O_{10}^+(2)$	1	1	IRR perm527 c15	s 1d	3.9
187	$S_{10}(2)$	1	1	IRR perm1056 c32	s 1d	3.2
187	$O_{10}^-(2)$	1	1	IRR perm528 c20	s 1d	2.9
189	A_9	1	1	IRR ind i9 d35 c25	s 1d	2.3
189	A_{21}	1	1	SymmetricSquare(ρ_{20})	s 1	2.0
189	$L_4(4)$	2	1	IE i85 (i84 d6 c12)	1d/8	146 + 20
189	$3.U_3(8)$	2	1	BB i513 d42 c170 Ri513 $[21,168]_2$	1d/32	39 + 307
189	$U_4(3)$	1	1	IRR perm280 c32	s 1d	0.6

Deg	Group	C	S	Method	N/D	Time
189	$3_2.U_4(3)$	2	1	BB p540 d2 c82		
				Ri112 $[45_2, 144_2]$	2d/27	31 + 22
189	$S_6(2)$	1	1	IRR ind i63 d5 c25	s 1d	1.2
189	$3.G_2(3)$	4	1	BB i351 d54 c180 Ri1456		
				$[27_2^{1+3\times2}]$ (p. 138)	3d/648	2.8 + 94
189	J_2	2	1	BB i1008 d1 c76 Ri525		
				$[6, 9^{1+2}, 12^{1+2+2}, 24^{2+2}]$	2d/48	0.8 + 5.5
190	A_{21}	1	1	ExteriorSquare(ρ_{20})	s 1	0.0
195	$S_6(3)$	1	1	IRR perm364 c26	s 1d	2.7
195	$O_7(3)$	1	1	IRR perm364 c22	s 1d	1.6
196	$S_4(8)$ *	1	1	IRR ind i2016 d1 c32	2d	19.5
196	${}^3D_4(2)$	1	1	GE i2457 [28, 168]	2d	34 + 89
200	$2.S_4(7)$	2	1	BB p800 c28		
				Ri400 $[1, 4_2, 48, 63_2, 84_2]$	4d/2352	547 + 16
204	$U_5(4)$ *	1	2	IE i66625 (DI i51 d4) (p. 89)	1d/8	48 + 8.7
204	$S_4(4)$	2	1	AIR ind i85 d6 c30	s 2d/1	8.7 + 20
204	$O_8^-(2)$	1	1	IRR perm765 c31	s 1d	1.3
205	$5.U_5(4)$ *	4	1	GE i66625 [1, 204: i30 d136]	1d/8	402 + 65
208	A_{13}	1	1	IRR perm286 c18	s 1d	1.7
208	$2.L_4(3)$	2	1	IE i40 (AIR ind i13 d18 c18)	2d/81	23 + 19
208	$S_4(5)$	2	1	BB i325 d8 c72		
				Ri312 $[10, 20, 30, 40, 48, 60]_2$	2d/50	6.2 + 26
208	$2.S_4(5)$	2	2	BB i325 d16 c192		
				Ri624 $[10, 20, 30, 40, 48, 60]_2$	3d/100	6.2 + 27
209	A_{22}	1	1	SymmetricSquare(ρ_{21})	s 1	2.6
209	J_1	1	1	IRR perm1045 c11	s 2d	6.5
210	A_{10}	1	1	IRR ind i10 d42 c16	1d	6.6
210	A_{11}	1	1	IRR ind i330 d1 c24	s 1	2.7
210	A_{22}	1	1	ExteriorSquare(ρ_{21})	s 1	0.0
210	$U_4(3)$	1	1	IRR ind i280 d1 c32	s 1d	1.1
210	$2.U_4(3)$	2	1	AIR ind i540 d1 c56	[-]	5.8
	Ri112			$[10, 20_2, 90, 90]$	1d/54	21 + 5.9
210	$3_1.U_4(3)$	2	1	AIR ind i126 d10 c100	s 1d	66
	Ri162			$[84_2, 126_2]$	2d/60	7.0 + 8.8
210	$6_1.U_4(3)$	2	1	BB i126 d12 c104		
				Ri112 $[30, 60, 120]_2$	2d/27	11 + 18
210	$3.U_6(2)$	2	1	IE i20736 [$3.M_{22}$]	2d/64	+15
210	$S_6(2)$	1	1	IRR ind i28 d10 c16	s 1d	1.1
210	$O_8^+(2)$	1	1	IRR ind i120 d7 c56	s 1d	3.3
210	M_{22}	1	1	ExteriorSquare(ρ_{21})	s 1d	0.0
210	$2.M_{22}$	1	1	IRR ind i330 d1 c30	s 1d	1.7
210	$2.M_{22}$	1	1	IRR ind i231 d1 c21	s 1d	2.3
210	$3.M_{22}$	2	1	BB i672 d2 c28		
				Ri22 $[84, 126]_2$	2d/80	2.0 + 7.4

Deg	Group	C	S	Method	N/D	Time
210	$6.M_{22}$	2	1	BB i77 d12 c20		
				Ri22 $[36,84,90]_3$	4d/280	25 + 47
210	$6.M_{22}$	4	1	BB i672 d2 c14		
				Ri22 $[90,120]_2$	3d/168	18 + 44
216	A_9	1	1	IRR perm504 c36	2d	4.5
216	$2.A_{10}$	1	1	IRR ind i45 d8 c10	s 2d	3.2
216	$12_1.U_4(3)$	4	1	BB $\rho_{30} \otimes \rho_{40}$ c64 Ri112		
				$[36,60,120]_4$ (p. 139)	2d/405	202 + 91
216	$12_2.U_4(3)$	4	1	BB $\rho_{40} \otimes \rho_{72}$ c192		
				Ri112 $[36,180]_4$ (p. 139)	2d/54	40 + 139
216	$S_6(2)$	1	1	IRR perm378 c26	s 1d	0.8
216	$2.J_2$	1	2	AIR ind i280 d2 c44	s 3d/1	6.8 + 32
217	$L_5(2)$	1	1	IRR perm248	s 1	1.6
217	$L_6(2)$	1	1	IRR ind i651 d7 c75	2d	26.7
219	$^3D_4(3)$ *	1	1	IRR perm26572 c172	2d	32
220	A_{13}	1	1	IRR ind i286 d1 c18	s 1d	1.4
220	$U_5(2)$	2	1	BB i165 d4 c32 Ri297		
				$[10_2, 30_2, 40_2, 60, 80]$	2d/96	9.9 + 8.4
220	2.Suz *	1	2	IE i56609280 (BB i440 d1		
				Ri24 $[20, 45^2, 55^2]$)	3d/792	13 + 66
221	$U_4(4)$	2	1	BB p1040 c16		
				Ri325 $[17_2, 204]$	2d/16	20 + 18
224	$2.A_9$	1	1	IRR ind i84 d8 c32	2d	9.5
224	A_{10}	1	1	IRR ind i10 d35 c20	2d	6.0
224	$4.U_4(3)$	4	1	BB i126 d16 c160 Ri672		
				$[4, 20^{3\times1}, 40, 60^2]_2$	2d/81	88 + 104
224	$S_4(7)$	1	1	IRR perm400 c16	s 1d	2.3
224	$2.O_8^+(2)$	1	1	IRR ind i120 d8 c64	2d	4.1
224	J_2	2	1	BB i1008 d1 c2 Ri525		
				$[2, 6, 9^{1+2}, 12^{1+3+3}, 24^{2+2}]$	2d/384	3.4 + 6.9
225	A_{10}	1	1	IRR ind i210 d2 c26	s 1d	6.5
225	$S_4(4)$	4	1	BB i1360 d1 c80		
				Ri85 $[18, 30, 45, 72, 60]$	2d/480	7.1 + 15
225	J_2	1	1	IRR ind i525 d1 c39	2d	4.5
230	A_{23}	1	1	SymmetricSquare(ρ_{22})	s 1	2.4
230	M_{23}	1	1	IRR perm253 c11	s 1d	2.4
231	A_{11}	1	1	IRR ind i162 d2 c28	s 1	4.0
231	A_{23}	1	1	ExteriorSquare(ρ_{22})	s 1	0.0
231	$U_6(2)$	1	1	IRR perm672 c52	s 1d	3.2
231	$3.U_6(2)$	2	1	BB p2079 c141 Ri228096		
				$[21, 21^2, 84^2]_2$	2d/60	8.5 + 41
231	M_{22}	1	1	IRR ind i77 d5 c35	s 1d	2.5
231	$3.M_{22}$	2	1	AIR ind i22 d30 c60	3d/1	20 + 32
231	M_{23}	1	1	IRR ind i253 d1 c11	s 1	1.8

Deg	Group	C	S	Method	N/D	Time
231	M_{23}	2	1	IE i23 [M_{22}]	3d/120	2.5 + 13
231	M_{24}	2	1	IE i276 (ext i2: i77 d5 c35)	3d/120	5.8 + 41
231	HS	1	1	IRR ind i1100 d1 c58	2d	6.2
231	McL	1	1	IRR ind i275 d21 c197	2d	17
234	$L_4(3)$	1	1	IRR ind i130 d2 c14	s 1d	2.9
238	$S_8(2)$	1	1	IRR ind i136 d28 c116	2d	22
240	$U_3(16)$ *	1	2	IE i4097 (i17, IRR ind i30 d16)	1d/32	37 + 71
241	$U_3(16)$	16	1	GE i4097 [1, 240] (DI i15 d16: BB i68 d8 c32 Ri256 [$1_{16}^{16\times 1}$])	2d/272	352 + 6053
246	$O_8^-(3)$ *	1	1	IRR perm1066 c34	2d	43
248	$2.L_4(5)$	1	1	IRR perm3100 c88	2d	10
248	Th *	1	1	G[I]E[$\neg\chi$] $2^5.L_5(2)$ (p. 115)	1d/16	105 + 4.6

9.2. Representations of Higher Degree

The following table gives a summary of the representations of degree higher than 250 of quasi-simple groups which we have computed.

The conventions for the table are the same as before, except that we give the maximum numerator and denominator LCM separate columns, and the latter is in factored form to save space. We also give only the total time to save space (i.e., we do not split out the time to construct the representation(s) for a subgroup H when relevant). If a time is at least an hour, then we give the time Th for T hours. As before, an asterisk (*) after the group name indicates that the representation is a minimal-degree faithful representation of the group.

To summarize the chief results, we have succeeding in constructing the following faithful absolutely irreducible ordinary representations:

- The minimal-degree representation of every sporadic group and its covers except for the Monster group (degree 196883) and the double cover 2.B of the Baby Monster (degree 96256).
- All representations of every sporadic group to degree 10000 at least.
- All representations of every cover of every sporadic group to degree 1000 at least.
- All representations of every Mathieu group and its covers.
- All representations to degree 1000 at least for the following groups: $U_6(2)$ and $2.U_6(2)$, $G_2(3)$, $2.G_2(3)$, $G_2(4)$, $2.G_2(4)$, $G_2(5)$, $S_8(2)$, ${}^2F_4(2)'$.

We note the following statistics for this table:

- There are 260 representations.
- There are 158 rational representations; of these, 45 were computed by IrreducibleRationalRepresentations.
- There are 102 irrational representations; of these, none were computed by AbsolutelyIrreducibleRepresentation (since the other methods were more applicable in high degree).
- 26 representations were computed by IrreducibleExtension.
- 43 representations were computed by GeneralExtension.
- 128 representations were computed by BBReductionRepresentation.

Deg	Group	C	S	Method	N	D	Time
252	$2.J_2$	1	2	BB i100 d12 c72 Ri100 $[6_2, 14, 42, 56, 64]$	3d	$2.3^3.7$	14
252	McL	1	1	IRR perm275 c11	1d	1	2.1
252	$U_6(2)$	1	1	IRR perm693 c53	1d	1	9.6
252	M_{24}	1	1	IRR perm276 c12	1d	1	3.3
253	Co_2	1	1	ExteriorSquare(ρ_{23})	1d	1	+0.1
253	Co_3	1	1	ExteriorSquare(ρ_{23})	1d	1	+0.1
253	M_{23}	1	1	IRR perm506 c22	2d	1	2.4
253	M_{24}	1	1	IRR ind i276 d1 c12	s 1	1	2.3
260	$O_8^+(3)$ *	1	1	IE $O_7(3)$, i378, i2	1d	3^2	13
265	$S_4(23)$ *	2	1	IE $L_2(23^2)$:2, i2	3d	23	127

Deg	Group	C	S	Method	N	D	Time
272	$U_3(17)$ *	1	2	IE i4914 (i3, BB i34 d32 c32 Ri289 $[16^8, 16^9_2]$)	6d	$2^2.17^2$	520
273	$^3D_4(2)$	1	1	RR $(\rho_{26})^2$ c20	2d	1	20
273	$G_2(3)$	1	1	IRR ind i351 d1 c2	1d	1	5
273	$3.U_3(17)$ *	6	1	GE i4914 $[1, 272]_6$; ρ_{272}: IE i3, i3, BB i34 d32 c32 Ri289 $[16^8, 16^9_2]$ (p. 148)	4d	$2^2.17^2$	1333
275	Co_2	1	1	IRR perm2300 c92	2d	1	8.3
275	Co_3	1	1	IRR perm276 c12	s1	1	2.7
276	Co_1 *	1	1	GE Co_3 $[23, 253]$	2d	1	+18
280	$G_2(5)$	1	1	GE i3906 $[40, 240]$	2d	5^2	112
280	M_{22}	2	1	BB i77 d10 c70 Ri77 $[10, 15, 30, 45^2, 45^3]$	2d	2^7	12
280	$U_4(3)$	2	1	BB i112 d10 c8 Ri112 $[10, 20_2, 30, 40_2, 90, 90]$	2d	$2^2.3^4$	21
288	J_2	1	1	IRR perm1008 c76	2d	1	7.2
288	$L_3(7)$	6	1	BB p5586 c24 Ri57 $[96^{1+1+1}]$	4d	7^3	29
299	Co_1	1	1	GE Co_3 $[1, 23, 275]$	3d	$2^2.3^2.23$	+47
300	$^2F_4(2)'$	1	1	IRR ind i1600 d1 c19	2d	1	20
300	J_2	1	1	IRR ind i100 d7 c44	2d	1	10.4
300	$G_2(4)$	2	1	IE J_2	4d	2.3.7	48
320	$U_5(2)$	1	2	BB i165 d4 c8 Ri297 $[10_2, 30_2, 120, 160_2]$	2d	2^7	13
323	J_3	2	1	BB p6156 c324 Ri6156 $[1, 2:16, 17, 2:17_2, 34, 68, 120]$	4d	$2^5.3.5.17$	34
324	$^3D_4(2)$	1	1	RR $(\rho_{26})^2$ c11	2d	1	33
324	J_3	1	1	IRR perm6156 c324	3d	1	24
324	$3.J_3$	2	1	BB i6156 d2 c146 Ri14688 $[18, 19^2, 20^2, 36, 36^2, 60^2]_2$	3d	$2^2.3^2.5.19$	757
325	$^2F_4(2)'$	1	1	IRR ind i2925 d1 c19	3d	1	52
330	$2.M_{22}$	1	1	IRR ind i77 d10 c18	2d	1	8.3
330	$3.M_{22}$	2	1	DI i22 d15 (AIR ind i120 d2)	s1	1	4.3
330	$6.M_{22}$	2	1	DI i330 d1	s1	1	2.0
336	J_2	1	2	BB i525 d2 c66 Ri100 [14,21,27,42,56,64]	3d	$2^2.3^3$	15
336	$2.J_2$	1	2	BB i100 d14 c88 Ri100 $[14, 21, 27, 42, 56, 64]$	3d	$2^4.3^3.7$	31
336	$12.M_{22}$	8	1	BB i22 d384 c48 Ri22 $[12_1.L_3(4)]$ $[96, 120^2]_4$	6d	$2^8.3^5.5$	+2335
342	$U_3(19)$ *	1	2	IE i6860 (i5, BB i2 d342 c16 Ri361 $[18^4, 18^5_2, 36^5]$)	5d	$2^2.13.19^2$	575
342	$3.L_3(7)$:2	4	1	IE $L_3(7)$ (DI i57 d6)	1d	2.7	214
342	$3.O'N$ *	4	1	IE $3.L_3(7)$:2	2d	2.7^2	+16.1h

Deg	Group	C	S	Method	N	D	Time
350	$G_2(4)$	1	1	IRR perm416 c8	1d	1	19
350	$2.J_2$	1	2	BB p5600 c38 Ri100 $[7,27,42,56,64]$	3d	$2^4.3^3.7$	23.5
351	$^2F_4(2)'$	1	1	IRR perm1600 c9	3d	1	27
351	$^2F_4(2)'$	2	1	BB $(\rho_{54})^2$ c48 Ri1600 $[13,26,27,39^{1+2},52^2,64]$	3d	$2^4.3^3.13$	26
351	$3.G_2(3)$	2	1	DI i351 d1	s1	1	3
351	$^3D_4(2)$	3	1	BB $(\rho_{52})^2$ c14 Ri2457 $[2_3,21,28,48_3,84,168]$	2d	2^6	78
351	$3.Fi_{22}$ *	2	1	IE $3.^2F_4(2)'$, i3 (IRR p1600)	4d	$2^2.5$	2745
352	$2.Fi_{22}$ *	1	1	IRR perm28160 c80	3d	1	85
357	$O_8^-(2)$	1	1	IRR ind i1071 d1 c17	2d	1	18
364	$G_2(4)$	1	1	IRR perm1365 c15	2d	1	50
364	$2.G_2(4)$	1	2	GE i1365 $[60,64,120,120]_2$	3d	2^5	214
364	Suz	1	1	IE $G_2(4)$	2d	2^2	+37
364	2.Suz	2	1	IE $2.G_2(4)$	3d	2^5	+61
378	$3.G_2(3)$	2	1	DI i378 d1	s1	1	3
378	$G_2(4)$	1	1	IRR ind i2016 d1 c60	3d	1	25
378	Ru *	2	1	IE $2^6.U_3(3).2$	1d	2^5	31
384	$3.M_{22}$	2	1	BB i77 d18 c26 Ri22 $[3.L_3(4)]$ $[84^2,90,126]_2$	5d	$2^6.3.5.7$	47
384	$6.M_{22}$	2	1	BB i176 d12 c26 Ri22 $[6.L_3(4)]$ $[84,90^2,120]_2$	4d	$2^5.3.5.7$	102
384	$12.M_{22}$	4	1	BB $\rho_{42} \otimes \rho_{224}$ c56 Ri22 $[12_1.L_3(4)]$ $[48,96,120^2]_4$	5d	$2^6.3^2.5^2.7$	+670
385	M_{22}	1	1	IRR perm616 c6	s 2d	1	101
385	$U_6(2)$	1	1	BB i693 d10 c326 Ri672 $[165,220]$	3d	$2^2.3$	108
406	Ru	1	1	IRR ind i4060 d1 c140	3d	1	66
429	Fi_{22}	1	1	GE R(2) $[78,351]$	4d	$2^2.5$	99
429	3.Suz	2	1	GE sh $3.G_2(4)$ $[65,364]_2$	3d	$2^3.3$	220
440	$2.M_{22}$	1	1	IRR ind i672 d1 c30	2d	1	70
440	$U_6(2)$	1	1	IRR perm672 c51	2d	1	20
448	$G_2(3)$	2	1	BB i364 d6 c32 Ri351 $[14^2,21^2,27^2,42,56,64]$	3d	$2^6.3^3.7$	20
448	$2.J_2$	1	2	BB i100 d14 c36 Ri100 $[14,21,27,42,56,64]^2$	3d	$2^5.3^3.7$	54
462	$3.U_6(2)$	2	1	$\rho_{21} \otimes \rho_{22}$	1d	1	+0.1
468	$^3D_4(2)$	1	1	IRR perm819 c11	2d	1	274
476	$O_8^-(2)$	1	1	IRR perm765 c38	2d	1	13
476	$O_8^-(2)$	1	1	IRR ind i119 d6 c30	2d	1	38
483	M_{24}	1	1	IRR perm759 c33	2d	1	19.5
495	$3.O'N$	2	1	GE $3.L_3(7){:}2$ $[152,343]_2$	5d	$2.3^2.7.19$	2601

Deg	Group	C	S	Method	N	D	Time
506	$U_3(23)$ *	1	2	IE i12168 (BB i2 d506 c20 Ri2 $[253, 253]_2$)	5d	$3.13.23^2$	2049
510	$S_8(2)$	1	1	GE $O_8^-(2)$:2 [34, 476]	3d	2.3^2	1851
520	$2.O_8^+(3)$ *	1	1	BB p2160 c52 R $O_7(3)$ [1, 78, 168, 273]	3d	2.3.13	41
546	$G_2(3)$	1	1	IRR ind i364 d2 c16	2d	1	86
560	$2.G_2(4)$	1	2	BB i416 d24 c320 Ri416 $[2.J_2]$ $[12, 84, 128, 252]_2$	5d	$2.3^3.5^2.7$	+204
560	$4.M_{22}$	2	1	BB i77 d32 c224 Ri77 $[16, 16, 64, 80]_2$	3d	$2^3.3^3.5$	221
560	$U_4(3)$	1	2	BB p13440 c78 Ri112 $[40, 80^{1+1}, 90^{2+2}]$	2d	2.3^4	60
560	$U_6(2)$	1	2	IE i891 (DI i560 d1)	s1	2^3	85
572	2.Suz	2	1	GE i232960 $[132, 440]_2$	2d	3^4	215
595	$O_8^-(2)$	1	1	BB i119 d10 c38 Ri119 [10, 135, 180, 270]	1d	2^5	47
595	$S_8(2)$	1	1	IE $O_8^-(2)$	1d	2^5	+108
612	$U_4(4)$	4	1	IE i325 (BB i272 d6 c48 Ri3 $[204, 204, 204]_2$)	2d	2^5	591
616	2.HS	2	1	BB i5600 d1 c96 Ri100 $[2.M_{22}]$ [56, 120, 440]	4d	$2^3.3.5.7.11$	95
616	$U_6(2)$	1	1	IRR perm891 c51	2d	1	49
616	$2.U_6(2)$	1	1	IRR ind i672 d1 c10	1d	1	63
624	$^2F_4(2)'$	2	1	BB p1600 c9 Ri1755 $[4, 10^{1+1}, 16^{2+2}, 20^{6\times1}, 40^{2+2}, 64^4]$	2d	$2^9.5$	121
637	$^3D_4(2)$	1	1	BB i819 d7 c3 Ri2457 $[21, 28, 84, 168^{1+2}]$	1d	2^6	63
640	$U_4(3)$	2	1	BB i567 d5 c36 Ri112 $[30^{4\times1}, 80^{1+1}, 90^{2+2}]$	2d	$2^2.3^4$	80
646	J_3	2	1	BB p46512 c74 Ri14688 $[18, 18_2^{4\times1}, 19, 20, 60]$	5d	$2^3.3^2.5.19$	556
650	$^2F_4(2)'$	1	1	IRR perm1755 c12	3d	1	1757
650	$G_2(4)$	1	1	IRR perm1365 c15	3d	1	249
651	$G_2(5)$	1	1	GE i3906 [1, 6, 20, 24, 120, 480]	2d	5^4	170
660	$U_5(2)$	1	1	BB p1408 c28 Ri165 $[12, 16, 27, 36, 72, 81, 128, 144^{1+1}]$	2d	$2^6.3^2$	255
672	$6.U_6(2)$	2	1	BB i2016 d1 c12 Ri693 $[16, 40, 160, 216, 240]_2$	3d	$2^8.3$	1.2h
675	$^2F_4(2)'$	1	1	IRR perm1755 c12	4d	1	1015
680	He	1	1	IRR perm2058 c80	3d	1	192
693	HS	1	1	IRR perm1100 c62	2d	1	72.5
702	$^2F_4(2)'$	2	1	BB i1755 d1 c6 Ri1755 $[1, 5, 10^{1+1+2+2}, 16, 20^{1+1+1}, 35^2, 40^{2+4}, 64^4]$	2d	$2^7.5$	115

Deg	Group	C	S	Method	N	D	Time
703	R_{27} *	1	1	BB i19684 d1 c532 Ri19684 $[1, 702]$	1d	3^3	1277
728	$G_2(3)$	1	1	BB i364 d3 c28 Ri351 $[14^{1+2}, 21^{1+2}, 27^{1+2}, 42^3, 56^4, 64^3]$	3d	$2^5.3^3.7$	34
729	$G_2(3)$	1	1	BB i364 d3 c28 Ri351 $[7, 14^{1+1}, 21^{1+2}, 27^{1+2}, 42^3, 56^3, 64^4]$	3d	$2^5.3^3.7$	28
729	$2.G_2(3)$	2	1	BB $\rho_{54} \otimes \rho_{54}$ c56 Ri351 $[7, 14^{1+1}, 21^{1+2}, 27^{1+2}, 42^3, 56^3, 64^4]_2$	3d	$2^5.3^3.7$	128
760	HN	1	1	GE A_{12} $[1, 132, 165, 462]$ (p. 122)	5d	$2^3.3^2.7$	359
770	HS	1	1	IRR ind i1100 d1 c54	2d	1	58
770	HS	2	1	GE M_{22} $[210, 560]$	3d	$2^4.3^2.11$	+425
770	M_{23}	2	1	BB i1771 d1 c77 Ri23 $[210, 280_2, 280_2]$	3d	$2^2.3^2.11$	220
770	M_{24}	2	1	IE M_{23}	6d	$2^2.3^2.11.23$	+893
770	McL	2	1	BB i275 d210 c198 Ri275 $[U_4(3)]$ $[210, 560_2]$	3d	$2^3.3^5.5$	+1301
770	$U_6(2)$	2	1	BB i693 d20 c28 Ri6336 $[S_6(2)]$ $[35, 315, 420]$	3d	$2^4.3$	345
780	Suz	1	1	IRR perm1782	3d	1	100
780	6.Suz	2	1	GE i232960 $[120, 660]_2$	2d	3^4	1386
782	Fi_{23} *	1	1	IRR perm31671 c185 (p. 64)	3d	1	596
783	Ru	1	1	IRR perm4060 c140	4d	1	140
783	$3.Fi'_{24}$ *	2	1	GE$[\neg\chi]$ Fi_{23} $[1, 782]$ (p. 114)	4d	$2^6.3^2.23$	+2.0h
792	3.McL	2	1	BB i275 d72 c168 Ri22275 $[1, 35, 64, 64, 70, 90, 126]_2$	4d	$2^5.3^2.5.7$	1770
792	$2.U_6(2)$	1	1	BB i1408 d1 c18 R $U_5(2)$ $[132, 660]$	2d	$2^6.3^2$	+89
816	J_3	1	2	BB p17442 c152 Ri23256 $[1, 9, 9, 10, 16, 16, 18, 20, 20_2, 32, 40]$	3d	$2^4.3^3.5$	818
819	$G_2(3)$	1	1	BB i364 d3 c15 Ri351 $[14^{3\times1}, 21^{1+2}, 27^{2+2}, 42^3, 56^4, 64^4]$	3d	$2^6.3^3.7$	38
819	$G_2(4)$	2	1	BB $(\rho_{65})^2$ c24 Ri1365 $[3_2, 36_2, 60^{4\times1}, 180^{3\times1}]$	2d	$2^7.5$	132
825	HS	1	1	IRR perm1100 c58	2d	1	72
832	$G_2(3)$	1	1	BB i364 d6 c32 Ri351 $[7^{1+1}, 12, 14^2, 21^{1+1}, 27^{2+2}, 42^2, 56^4, 64^5]$	3d	$2^6.3^3.7$	53
833	$F_4(2)$ *	1	1	GE $S_8(2)$ [238, 595]	3d	$2^2.3$	+481
896	Co_3	2	1	IE M_{23}	5d	$2^7.3^2.5$	+1247
896	HS	2	1	GE M_{22} $[231, 280_2, 385]$	5d	$2^{10}.3^2.11$	+471
896	M_{23}	2	1	BB i23 d231 c53 Ri253 $[35, 35, 64, 64, 70, 90, 90, 126]$	4d	$2^7.3.5.7$	356
896	McL	2	1	IE $U_4(3)$	5d	$3^3.5$	+779
896	$U_4(3)$	1	1	IRR ind i112 d16 c2	3	1	657

Deg	Group	C	S	Method	N	D	Time
918	$S_8(2)$	1	1	BB p2295 c19 Ri255 $[15, 84, 315, 504]$	2d	$2^4.3^2$	265
924	2.HS	2	1	BB i1100 d2 c24 Ri100 $[2.M_{22}]$ $[120, 154_2, 210, 440]$	5d	$2^3.3^3.5.7.11$	216
924	6.Suz	2	1	GE $2.G_2(4)$ $[364, 560]_2$	5d	$2^6.3^3.5^2.7$	+982
924	$3.U_6(2)$	2	1	BB i693 d10 c44 Ri891 $[84, 840]_2$	1d	2^5	918
930	$G_2(5)$	1	1	GE i3906 $[1, 5, 24, 60, 120, 240, 480]$	2d	2.5^4	323
960	$G_2(5)$	2	1	GE i3906 $[480, 480]$	1d	3.5^2	304
990	A_{11}	1	1	IRR ind i55 d42 c39	2d	1	430
990	M_{23}	2	1	$\rho_{22} \otimes \rho_{45}$	1d	2^3	0.1
990	M_{24}	2	1	IE M_{23}	2d	$2^3.23$	1257
1000	2.HS	1	2	BB i100 d20 c60 Ri100 $[2.M_{22}]$ $[20, 252, 210, 308]$	4d	$2^3.5^2.11$	363
1001	Fi_{22}	1	1	GE $2^{10} : M_{22}$ $[385, 616]$	1d	2^5	189
1001	Suz	1	1	BB p1782 c36 R $3^5{:}M_{11}$ $[11, 110, 220, 660]$	2d	2.3^4	186
1016	Sz(128) *	1	1	IE i16385	1d	2^6	31.0h
1029	He	2	1	BB p4116 c57 Ri8330 $[1, 20, 64, 126, 168, 192, 270]$	3d	$2^5.3.5.7$	177
1035	M_{23}	1	1	IRR perm1288 c56	s 2d	1	206
1035	M_{24}	1	1	IRR perm1288 c56	2d	1	213
1035	M_{24}	2	1	$\rho_{23} \otimes \rho_{45}$	1d	1	0.1
1056	HS	1	1	BB p3850 c36 R M_{22} $[55, 154, 231, 385]$	4d	$2^5.3^2.5$	124
1056	$6.U_6(2)$	2	1	BB $\rho_{42} \otimes \rho_{56}$ c2 Ri891 $[336, 720]_2$	1d	2^5	587
1085	$G_2(5)$	1	1	GE i3906 $[1, 24, 40, 60, 240^{1+1}, 480]$	3d	$2^2.5^6$	456
1105	$F_4(2)$	1	1	GE $S_8(2)$ [510, 595]	2d	$2^6.3^3$	+460
1140	J_3	1	1	BB p6156 c73 Ri6156 $[1^{1+2},$ $16^{2+3}, 17^{2+3}, 34^{3+5}, 68^5, 120^3]$	3d	$2^5.3.5.17$	308
1155	A_{11}	1	1	IRR ind i55 d42 c30	2d	1	2.5h
1155	$U_6(2)$	1	1	IRR perm1408 c26	2d	1	1325
1155	$3.U_6(2)$	2	1	BB i693 d10 c6 Ri891 $[105, 210, 840]_2$	1d	$2^5.3$	850
1215	J_3	2	1	BB p6156 c8 Ri6156 $[1,$ $16^{2+3}, 17^{2+4}, 34^{3+3}, 60_2^{3+4}, 68^6]$	7d	$2^7.3.5.19$	2954
1232	2.HS	2	1	BB $\rho_{56} \otimes \rho_{231}$ c104 Ri100 $[2.M_{22}]$ $[210, 252, 330, 440]$	4d	$2^5.7$	2461
1232	$2.U_6(2)$	1	1	IRR ind i1408 d1 c3	2d	1	1723
1265	M_{24}	1	1	BB i1288 d1 c56 R M_{23} [230, 1035]	3d	2.11	+87
1275	He	1	1	BB p2058 c45 Ri8330 $[1, 20^3,$ $40, 60, 64, 105^2, 108, 192, 270^2]$	4d	$2^6.3^2.5.7$	197
1275	He	2	1	BB $(\rho_{102})^2$ c120 Ri8330 $[20, 30_2,$ $45_2, 60, 64, 90_2, 108, 126, 192, 270^{1+1}]$	3d	$2^8.3.5.7$	844

Deg	Group	C	S	Method	N	D	Time
1275	$S_8(2)$	1	1	BB i5440 d1 c44 Ri255			
				$[36, 105, 504, 630]$	2d	$2^5.5$	430
1300	$^2F_4(2)'$	1	1	IRR ind i1600 d1 c1	3d	1	3545
1320	A_{11}	1	1	BB i11 d252 c30			
				R A_{10} $[252, 300, 768]$	4d	$2.3^2.5^2$	433
1320	A_{12}	2	1	IE A_{11}	5d	$2^2.3^2.5^2.7$	+2269
1333	J_4 *	2	1	GE$[\neg\chi]$ 2^{11}:M_{24} $[45_2, 1288]$ (p. 116)	2d	2^7	1354
1386	HS	1	1	BB p5775 c50			
				R M_{22} $[210, 231, 385, 560]$	4d	$2^4.3^2.11$	411
1386	$U_6(2)$	1	1	IRR ind i1408 d1 c2	s1d	1	1.4h
1386	$3.U_6(2)$	2	1	BB i693 d12 c46			
				Ri891 $[21, 105, 420, 840]_2$	1d	2^6	1073
1408	HS	1	1	BB p4125 c37			
				R M_{22} $[99, 154, 210, 385, 560]$	6d	$2^7.3^2.11$	416
1430	Fi_{22}	1	1	GE 2^{10} : M_{22}			
				$[1, 21, 77, 330, 385, 616]$	3d	$2^6.11$	570
1485	A_{12}	1	1	IRR ind i66 d42 c33 (p. 72)	2d	1	2126
1485	$3.U_6(2)$	2	1	BB i6336 d2 c56			
				Ri891 $[15, 630, 840]_2$	1d	2^6	1569
1540	$U_6(2)$	1	1	BB i672 d55 c46			
				Ri891 $[280, 1260]$	1d	2^6	1202
1615	J_3	1	1	BB p6156 c73 Ri6156 $[1, 16^{2+4}$			
				$17^{4+4}, 34^{3+4}, 68^8, 120^5]$	5d	$2^5.3.5.17$	903
1638	$O_7(3)$	1	1	BB i364 d6 c30 Ri3159 $[21, 27, 35,$			
				$105^{1+1}, 120^2, 210^{1+1}, 280, 405]$	4d	$2^6.3^2.7$	701
1728	$^2F_4(2)'$	1	1	BB i1755 d2 c4 Ri1755 $[2, 10^{1+2}$			
				$16^{5+5}, 20^{1+2+2+3+4+4}, 32, 40^{4+8}, 64^{11}]$	2d	$2^8.5$	784
1728	$6.Fi_{22}$ *	2	1	IE 6.R(2) (i2, i3, RR i1775 d2)	6d	$2^6.3^4.5.13^2$	47.8h
1750	HS	1	1	BB i100 d90 c20			
				R M_{22} $[90, 99, 231, 385^2, 560]$	4d	$2^5.3^2.5.7.11$	1254
1750	McL	1	1	IRR perm2025 c50	3d	1	1.4h
1771	Co_1	1	1	IE Co_2	2d	2^7	+1.0h
1771	Co_2	1	1	BB $\rho_{23} \otimes \rho_{253}$ Ri1024650			
				$[35, 56, 420, 420, 840]$	1d	2^6	1324
1771	Co_3	1	1	BB $\rho_{23} \otimes \rho_{253}$ R $2.S_6(2)$			
				$[8, 35, 48, 105, 120, 315, 420, 720]$	4d	$2^6.3.7$	778
1771	M_{24}	1	1	DI i1771 d1	s 1	1	0.4
1792	2.HS	2	1	BB i100 d252 c208 Ri100			
				$[2.M_{22}]$ $[252, 330^2, 440^2]$	4d	$2^6.3.7$	1.4h
1848	2.HS	1	2	BB i176 d56 c88 Ri100			
				$[2.M_{22}]$ $[308, 330^2, 440^2]$	5d	$2^4.3^2.5.7$	2205
1920	He	1	1	BB p8330 c104 Ri8330 $[1, 20, 60,$			
				64, 105, 2:108, 126, 128, 2:192, 270]	4d	$2^6.3^3.5.7$	621

Deg	Group	C	S	Method	N	D	Time
1920	J_3	3	1	BB p14688 c20 Ri6156 $[16^{4+4}$ $17^{4+4}, 34^{4+4}, 68^7, 120^8]$	42d	2.5.19.39d	7.1h
1925	A_{12}	1	1	BB p2520 c36 R A_{11} $[825, 110]$	6d	2.7	2571
1925	HS	1	1	BB i176 d21 c27 R M_{22} $[154, 210, 231, 385^2, 560]$	4d	$2^5.3^2.5.11$	812
1925	HS	1	1	BB i3850 d1 c33 R M_{22} $[55, 99, 210, 231, 385^2, 560]$	5d	$2^4.3^2.5.7.11$	817
1938	${}^2E_6(2)$ *	1	1	GE$[\neg\chi]$ $F_4(2)$ [833, 1105]	4d	$2^7.3^3$	+1.2h
1938	J_3	2	1	BB i6156 d1 c8 Ri6156 $[16^{4+4}$ $1, 16^{2+3}, 17^{2+4}, 34^{3+3}, 60_2^{3+4}, 68^6]$	31d	$2^5.3.5.31$d	3.1h
1980	2.HS	2	1	BB i100 d56 c48 R 2.M_{22} $[56, 120, 154, 330, 440^3]$	5d	$2^4.3^2.5.7$	1.1h
2024	2.Co_1	1	1	GE Co_2 [253, 1771]	2d	2^7	+1.9h
2024	Co_2	1	1	IRR perm2300 c82	2d	1	1.2h
2024	Co_3	1	1	BB i276 d22 c19 R 2.$S_6(2)$ $[8, 15, 35, 84, 105, 112, 189,$ $216, 280, 420, 560]$	4d	$2^7.3^3.5.7$	890
2024	M_{23}	1	1	IRR ind i23 d99 c99	4d	1	1.3h
2024	M_{24}	1	1	IE M_{23}	5d	$2^2.3$	+1.0h
2048	${}^2F_4(2)'$	2	1	BB i2925 d2 c15 Ri1755 $[4, 5^{1+1}, 10^{1+2+2}, 16^{3+3},$ $20^{4\times2+4+4}, 32^3, 40^{8+8}, 64^{13}]$	7d	$2^{11}.3^3.5.17^2$	3491
2080	2.Fi_{22}	1	1	GE 2.$O_7(3)$ $[182, 260, 1638]$	5d	$2^6.3^3.7$	2.2h
2277	Co_2	1	1	IRR ind i2300 d1 c5	s 1d	1	6.1h
2277	M_{24}	1	1	BB i3795 d1 c165 R M_{23} [253, 2024]	6d	$2^2.3^2.5$	+1143
2310	A_{11}	1	1	BB i11 d450 c45 R A_{10} $[450, 525, 567, 768]$	3d	$2^4.3.7$	1321
2310	3.$U_6(2)$	2	1	BB i693 d20 c10 Ri891 $[210, 840, 1260]_2$	1d	2^8	2951
2380	2.$F_4(2)$	1	1	GE$[\neg\chi]$ $S_8(2)$ $[1, 51, 135, 918, 1275]$	3d	$2^6.3^2.5^2.17$	+1.9h
2432	J_3	1	1	IRR perm14688 c64	10d	1	64.2h
2464	2.$U_6(2)$	1	1	BB i693 d16 c33 Ri6237 $[40^{1+2}, 64,$ $80^{1+1}, 160^{1+1}, 240^{1+1+1}, 360^{1+2}]$	2d	$2^7.3.5$	3535
2480	$5^3.L_3(5)$	1	1	GE i31 $[80, 240^{1+1}, 480^{4\times1}]$	2d	5^4	1.4h
2480	Ly *	2	1	G[I]E$[\neg\chi]$ $5^3.L_3(5)$ [2480] (p. 116)	4d	3.5^6	+19.3h
2520	HS	1	1	BB i176 d21 c33 R M_{22} $[154^2, 210^2, 231^2, 385^2, 560]$	5d	$2^4.3^2.5.7.11$	1706
2520	3.$U_6(2)$	2	1	BB i693 d10 c4 Ri891 $[21, 84, 315, 420, 1680]_2$	2d	$2^7.3.5$	1.4h
2673	A_{12}	1	1	BB p5775 c72 R A_{11} $[693, 990^{1+1}]$	4d	$2^3.3^2$	2255

Deg	Group	C	S	Method	N	D	Time
2750	HS	1	1	BB i5775 d1 c15 R M_{22} $[90, 210, 385^2, 560^3]$	5d	$2^4.3^3.11$	2503
2754	J_3	1	1	IRR perm17442 c78	9d	1	41.7h
2772	$3.U_6(2)$	2	1	BB i693 d12 c46 Ri891 $[21^{1+1}, 105, 210, 315, 420, 1680]_2$	2d	$2^8.3$	1.9h
3003	Fi_{22}	1	1	GE $2^{10} : M_{22}$ $[77, 616, 2310]$	1d	2^6	2034
3078	J_3	1	1	IRR perm20520 c88	9d	1	169.0h
3080	Fi_{22}	1	1	GE $2^{10} : M_{22}$ $[1, 21, 55, 77, 330, 616, 1980]$	3d	$2^9.3.7.11$	1.3h
3080	$U_6(2)$	2	1	BB i693 d10 c6 Ri891 $[105, 315, 420, 2240]$	1d	$2^7.3$	3.6h
3080	$2.U_6(2)$	1	1	BB $\rho_{22} \otimes \rho_{176}$ c22 Ri6237 $[32, 40^{1+1}, 80^{1+1}, 128, 160^{1+1+2}, 240^{1+1}, 360^{3\times1}, 480]$	3d	$2^8.3^2.5$	1.4h
3200	HS	1	1	BB i176 d56 c62 R M_{22} $[99, 154, 210, 231^2, 385^3, 560^2]$	5d	$2^7.3^2.5.11$	3581
3276	Ru	1	1	BB p4060 c24 R $2^6.U_3(3).2$ $[1^2, 14^{1+1}, 21^2, 27^2, 63^{1+1+2}, 126^2, 189^{1+2+3}, 378^{1+1+2}]$	3d	$2^9.3^2.7$	2906
3312	M_{24}	1	1	BB i24 d253 c264 R M_{23} $[253,\ 1035,\ 2024]$	6d	$2^2.3^2.7$	+1534
3344	HN	1	1	GE A_{12} $[1, 54, 132^2, 462^2, 616, 1485]$ (p. 122)	6d	$2^7.3^5.5.7^2.11^3$	+3.2h
3432	Suz	1	1	BB $(\rho_{143})^2$ c155 R 3^5:M_{11} $[44, 528, 660, 792, 880]$	3d	$2^2.3^5.5$	1.2h
3465	$3.U_6(2)$	2	1	BB $\rho_{22} \otimes \rho_{420}$ c8 Ri891 $[210, 315, 420, 840^{1+1+1}]_2$	1d	$2^6.3$	1.7h
3520	Co_3	2	1	IE McL (Schur index 2)	6d	$2^3.3^8.5.103$	+4.0h
3520	M_{24}	1	1	BB i2024 d2 c176 R M_{23} $[230,\ 231,\ 1035,\ 2024]$	6d	$2^6.7.11$	+1438
3520	McL	1	1	BB i275 d21 c97 R M_{22} $[21^2, 55^2, 99, 154^2, 210^2, 231, 560^3]$	5d	$2^7.3^2.5.7.11$	1.0h
3520	McL	1	2	BB i15400 d8 c176 R $U_4(3)$ $[560^{1+1}, 1120, 1280]$	5d	$2^3.3^7.103$	+12.8h
3588	Fi_{23}	1	1	GE $2^{11}.M_{23}$ $[1, 22, 253, 506, 1288, 1518]$ (p. 118)	5d	$2^{10}.7.23$	4.4h
3654	Ru	1	1	BB i4060 d1 c31 R $2^6.U_3(3).2$ $[1, 14, 21^2, 27, 42, 63^{1+2}, 126, 189^{1+2+4}, 378^{3\times1+2}]$	3d	$2^9.3^2.7$	1.1h
4025	Co_2	1	1	BB $\rho_{23} \otimes \rho_{253}$ c10 R 2^{10}:M_{22}:2 $[21, 22, 231^{1+1}, 440, 3080]$	3d	$2^7.3.7$	1.1h
4025	Co_3	1	1	BB i276 d22 c71 R McL $[22, 231, 252, 3520]$	5d	$2^7.3^2.5.7.11$	+1.4h

Deg	Group	C	S	Method	N	D	Time
4080	He	1	1	BB i8330 d2 c121 Ri8330 [2, 40, 64, 105^{2+2}, 2:108, 126, 128, 168^2, 192^{2+2}, 252, 270^{2+2}, 420^2]	4d	$2^8.3^3.5.7$	1.4h
4123	Th	1	1	GE[$\neg\chi$] $2^5.L_5(2)$ [155, 248, 3720] (p. 115)	2d	2^9	1.3h
4158	A_{12}	1	1	BB i12 d660 c34 R A_{11} [660, 1188, 2310]	6d	$2^6.3^2.5.7$	5.2h
4352	He	1	1	BB p8330 c104 Ri8330 [1, 20^2, 40, 60^2, 64^3, 90, 105^3, 108^{3+3}, 126, 128, 192^3, 252, 270^{2+4}, 420]	4d	$2^8.3^3.5.7$	2.2h
4371	B *	1	1	GE[$\neg\chi$] Fi_{23} [1, 782, 3588] (p. 119)	5d	$2^{11}.3^5.7.23$	+35.0h
4500	McL	1	1	BB p15400 c76 R M_{22} [55, 90, 99, 154^2, 210, 231^3, 385^5, 560^2]	5d	$2^7.3^2.5.7.11$	3.2h
4752	McL	1	2	BB p178200 c276 R M_{22} [90, 210^2, 231^2, 385^4, 560^4]	6d	$2^7.3^3.5.11$	65.6h
5005	Suz	2	1	BB $(\rho_{143})^2$ c155 R 3^5:M_{11} [55, 110, 132, 220_2, 440, 528^2, 660^2, 792, 880]	3d	$2^2.3^6.5$	+8.6h
5083	Fi_{23}	1	1	GE $2^{11}.M_{23}$ [253,1288,3542] (p. 118)	2d	2^8	10.9h
5103	McL	1	1	BB p15400 c76 R M_{22} [55, 90, 99, 154^2, 210, 231^3, 385^5, 560^2]	5d	$2^7.3^3.5.7.11$	2.6h
5313	M_{24}	1	1	BB i1771 d5 c92 R M_{23} [462, 1035, 1792, 2024]	7d	$2^7.3.5.7.11.23$	+5.4h
5544	Co_3	1	1	BB i276 d22 c53 R McL [22, 252, 1750, 3520]	6d	$2^7.3^5.5^2.7.11$	+3.0h
5544	M_{24}	1	1	BB i1771 d5 c385 R M_{23} [1540, 1980, 2024]	7d	$2^4.7.23$	+7.1h
5544	McL	1	1	BB p15400 c28 R M_{22} [90, 99, 154, 210^3, 231, 385^4, 560^5]	6d	$2^7.3^2.5.7.11$	9.1h
5775	A_{12}	1	1	BB i23040 d1 c20 R A_{11} [990, 1155, 1320, 2310]	6d	$2^5.3^2.5^3.7$	2.4h
5796	M_{24}	1	1	BB p10626 c93 R M_{23} [1792, 1980, 2024]	7d	$2^7.3^3.5^2.7$	+6.0h
5940	Suz	1	1	BB $(\rho_{143})^2$ c155 R 3^5:M_{11} [1, 10, 11, 44, 110^3, 132^4, 220, 528^2, 660^3, 880^2]	3d	$2^2.3^6.5.11$	+4.9h
6272	He	1	1	BB i8330 d1 c29 Ri8330 [1, 20^2, 60, 64^{1+3}, 90^{1+1}, 105, 108^{1+1}, 126^3, 128, 168^4, 180, 192^{3+4}, 252, 270^{2+4}, 420^2]	4d	$2^7.3^2.5.7$	7.3h
6528	He	1	1	BB $\rho_{102} \otimes \rho_{306}$ c144 Ri8330 [40, 60, 64, 90, 105^{1+3}, 108^{1+3}, 126, 128^2, 168, 180, 192^{1+3}, 252^2, 270^{4+4}, 420^3]	4d	$2^9.3^3.5.7$	23.2h
7084	Co_2	1	1	BB $(\rho_{253})^2$ c319 R 2^{10}:M_{22}:2 [924, 1540, 4620]	2d	2^8	9.4h

Deg	Group	C	S	Method	N	D	Time
7084	Co_3	1	1	BB $(\rho_{253})^2$ c290 R McL $[1540, 5544]$	7d	$2^8.3^5.5^2.7.11^2$	+17.7h
7497	He	2	1	BB $\rho_{102} \otimes \rho_{680}$ c478 R $\mathrm{S}_4(4)$:2 $[50, 85^{1+1}, 153, 256^2, 340^{1+1+2+2}, 408^{2+2}, 510^{2+2}, 900]$	7d	$2^9.3^2.5^2.17$	88.5h
7650	He	1	1	BB p29155 c271 R $\mathrm{S}_4(4)$:2 $[34^{1+1}, 50, 85^{4\times1}, 102^{1+1}, 256^2, 340^{1+1+3+3}, 408^{1+1}, 510^{2+2}, 900]$	8d	$2^8.3^2.5^2.17$	17.1h
7650	He	2	1	BB $\rho_{102} \otimes \rho_{306}$ c156 R $\mathrm{S}_4(4)$:2 $[50, 85^{1+1}, 153^2, 256^2, 340^{1+1+2+2}, 408^{2+2}, 510^{2+2}, 900]$	7d	$2^7.3^2.5^2.17$	105.2h
8019	McL	2	1	BB i275 d189 c215 R $\mathrm{U}_4(3)$ $[189, 420, 560^{1+1}, 640_2^{1+2}, 729^2, 896^2, 1120]$	6d	$2^5.3^{10}.5.7$	75.5h
8250	McL	2	1	BB i22275 d1 c145 R $\mathrm{U}_4(3)$ $[140, 210, 315^{1+1}, 420, 560^{3\times1}, 640_2^{1+2}, 729^2, 896^2]$	5d	$2^5.3^8.5.7$	21.3h
8671	Fi_{24}' *	1	1	GE$[\neg\chi]$ Fi_{23} [3588, 5083] (p. 121)	4d	$2^{12}.3.7.23$	+38.6h
8855	Co_1	1	1	IE 2^{11}:M_{24} (p. 90)	1d	2^4	3.5h
8855	Co_3	1	1	BB p11178 c78 R McL $[252, 1750, 5103]$	7d	$2^7.3^3.5^3.7.11^2$	+5.0h
8778	HN	2	1	BB $(\rho_{266})^2$ c148 R A_{12} $[132, 165, 462, 1485, 2376, 4158]$	8d	$2^9.3^4.5^2.7$	+122.2h
8910	HN	1	1	BB $(\rho_{266})^2$ c148 R A_{12} $[1, 54, 132^2, 275, 462, 616, 1925, 2640, 2673]$	9d	$2^5.3^5.5^2.7^1.11$	+55.3h
9405	HN	1	1	GE A_{12} $[11, 154, 462^{1+1}, 616, 1925, 5775]$	9d	$2^5.3^5.5^2.7.11$	+110.3h
9625	Co_2	2	1	IE McL	6d	$2^8.3^9.5.7$	+122.1h
9625	Co_3	2	1	IE McL	6d	$2^8.3^9.5.7$	+125.5h
9625	McL	1	1	BB i15400 d1 c74 R $\mathrm{U}_4(3)$ $[35^{1+1}, 90, 140, 210, 315^{2+2}, 420, 729^3, 896^3, 1280^2]$	5d	$2^6.3^8.5.7$	26.9h
9856	McL	2	1	BB p92400 c392 R $\mathrm{U}_4(3)$ $[280_2^{1+1}, 315^{1+1}, 420^2, 729^2, 896^3, 1120, 1280^2]$	5d	$2^{10}.3^8.5.7$	84.1h
10395	M_{24}	1	1	BB i276 d55 c93 R M_{23} [1035, 1540, 1792, 1980, 2024]	8d	$2^7.3^3.5^2.7.11.23$	+24.2h
10944	O'N *	1	1	BB p122760 c366 R $\mathrm{L}_3(7)$:2 (p. 144)	8d	$2^9.3^5.7^6.19$	202.4h

CHAPTER 10

Representations of $\mathrm{L}_2(q)$ and $2.\mathrm{L}_2(q)$

10.1. Introduction

In this chapter we describe the ordinary representations of $\mathrm{L}_2(q)$ and $2.\mathrm{L}_2(q)$ for $q < 100$ which we have constructed. For these groups, there are some known constructions for representations [Tan67, PS83, Bög93, Per95, Nic06], but these methods generally write the result over a non-minimal field. Apart from the trivial cases which can be handled by a permutation representation or direct induction, it has generally remained a very difficult problem to write the representations over minimal fields as q increases, but the hybrid algorithm is particularly effective for constructing such representations with reasonably small entries most of the time.

The irrational representations were generally either computed by ABSOLUTELYIRREDUCIBLEREPRESENTATION if the degree was small or by the hybrid algorithm BBREDUCTIONREPRESENTATION. In the latter case, the most suitable subgroup H for reduction was always the largest maximal subgroup, which for $\mathrm{L}_2(q)$ is known as the Borel subgroup (index $q-1$) [Wil09, 3.3.3]. The other maximal subgroups are very small, comparatively, so they are not suitable in general: reducing via such usually yields large entries in the result.

For all q, the representation of degree q is trivially constructed from the permutation representation of G of degree $q+1$, so we omit such cases. Also, since $\mathrm{L}_2(q)$ is isomorphic to some other standard group for $q = 2, 3, 4, 5, 9$, we omit these cases from the tables.

At the time of writing, some representations of $2.\mathrm{L}_2(97)$ remain too difficult to construct, since they involving splitting homogeneous modules over a very large number field, or with very high multiplicity.

10.2. Representations of $L_2(q)$

$L_2(q)$, q even, Degree $(q-1)$

Deg	q	C	Method	N/D	Time
7	8	1	IRR ind i28 d1 c4	1	0.3
7	8	3	AIR ind i28 d1 c4	1d/1	0.3 + 0.2
15	16	8	IE i17 (DI i15 d1)	2d/8	0.1 + 0.2
31	32	1	IRR ind i496 d1 c16	1	0.8
31	32	5	IE i33 (DI i33 d1)	2d/8	0.1 + 0.3
31	32	5	IE i33 (DI i33 d1)	2d/16	0.1 + 2.5
63	64	2	IE i65 (DI i63 d1)	2d/32	2.6 + 3.0
63	64	6	IE i65 (DI i63 d1)	2d/32	2.6 + 8.6
63	64	24	IE i65 (DI i63 d1)	5d/32	1.0 + 64.3

$L_2(q)$, q even, Degree $(q+1)$

Deg	q	C	Method	N/D	Time
9	8	3	AIR perm28 c4	1d/1	0.1 + 0.0
17	16	1	IRR ind i17 d2 c2	1	0.3
17	16	2	AIR perm68 c5	1d/1	0.1 + 0.1
17	16	4	BB p120 c8 Ri17 $[2_4, 15]$	2d/8	0.1 + 0.2
33	32	15	BB p496 c16 Ri33 $[2_15, 31]$	4d/16	3.2 + 2.1
65	64	1	IRR ind i65 d2 c2	s 1	1.0
65	64	3	BB i65 d6 c6 Ri65 $[2_3, 63]$	1d/16	4.4 + 2.4
65	64	3	BB i65 d6 c6 Ri65 $[2_3, 63]$	1d/16	4.9 + 2.5
65	64	6	BB i65 d12 c12 Ri65 $[2_6, 63]$	2d/16	5.9 + 4.5
65	64	18	BB p2016 c32 Ri65 $[2_6, 63]$	4d/32	8.4 + 46.4

$L_2(q)$, $q \equiv 3 \pmod 4$, Degree $(q-1)/2$

Deg	q	C	Method	N/D	Time
3	7	2	AIR ind i21 d1 c5	1d/1	0.1 + 0.1
5	11	2	AIR perm55 c5	1d/1	0.0 + 0.1
9	19	2	AIR perm171 c15	2d/1	0.2 + 0.3
11	23	2	AIR ind i253 d1 c21	2d/1	0.4 + 0.5
13	27	2	AIR perm351 c21	1d/1	0.3 + 0.1
15	31	2	AIR perm930 c60	4d/1	0.6 + 1.0
21	43	2	BB p903 c33 Ri44 [21]	4d/43	0.7 + 0.3
23	47	2	BB i1081 d1 c45 Ri48 [23_2]	4d/47	6.5 + 1.5
29	59	2	BB p1771 c45 Ri60 [29_2]	5d/59	2.1 + 1.7
33	67	2	BB p2211 c51 Ri68 [33_2]	7d/67	5.9 + 2.2
35	71	2	BB i2485 d1 c69 Ri72 [35_2]	6d/71	5.5 + 6.5
39	79	2	BB p6162 c156 Ri80 [39_2]	6d/79	8.2 + 0.2
41	83	2	BB p3403 c63 Ri84 [41_2]	7d/83	6.9 + 3.8

$L_2(q)$, $q \equiv 1 \pmod 4$, Degree $(q+1)/2$

Deg	q	C	Method	N/D	Time
7	13	2	AIR perm28 c4	1d/1	0.1 + 0.0
9	17	2	AIR perm36 c4	1d/1	0.1 + 0.1
13	25	1	IRR ind i36 d1 c2	1	0.3
15	29	2	AIR perm60	2d/1	0.1 + 0.9
19	37	2	AIR perm76	3d/1	0.2 + 0.2
21	41	2	AIR ind i210 d1 c10	4d/1	1.4 + 1.0
25	49	1	AIR ind i50 d1 c2	1d/1	0.4
27	53	2	AIR perm108	4d/1	0.6 + 1.5
30	61	2	BB p124 c4 Ri62 [$1, 30_2$]	5d/549	8.0 + 1.1
37	73	2	BB p148 c4 Ri74 [$1, 36_2$]	6d/73	2.0 + 0.5
41	81	1	IRR ind i81 d1 c2	1d	1.1
45	89	2	BB p180 c4 Ri90 [$1, 44_2$]	6d/178	8.2 + 2.4
49	97	2	BB p196 c4 Ri98 [$1, 48_2$]	6d/97	10.3 + 2.8

$L_2(q)$, q odd, Degree $(q-1)$

Deg	q	C	Method	N/D	Time
6	7	1	IRR perm7	1	0.1
10	11	1	IRR ind i11 d4	1	0.3
10	11	1	IRR ind i11 d4	1	0.3
12	13	3	AIR perm78 c9	1d/1	0.1 + 0.1
16	17	1	AIR perm136 c12	1d	0.1 + 0.1
16	17	3	AIR perm102 c8	2d/1	0.3 + 0.1
18	19	2	AIR perm57 c4	1d/1	0.1 + 0.1
18	19	2	AIR ind i171 d1 c17	1d/1	0.3 + 0.1
22	23	1	AIR ind i253 d1 c21	1d/1	0.5
22	23	2	AIR ind i253 d1 c21 Ri24 [22]	2d/92	0.1 + 0.6
24	25	6	BB p300 c18 Ri26 [12, 12]	2d/25	0.9 + 1.0
26	27	3	BB p351 c21 Ri28 [26]	1d/9	0.1 + 0.3
26	27	3	BB p702 c58 Ri28 [26]	1d/9	0.1 + 0.3
28	29	1	IRR perm406 c21	1d	0.2
28	29	2	AIR perm406 c21	3d/1	0.4 + 1.0
28	29	4	AIR perm203 c8 Ri30 [28]	2d/29	0.3 + 1.2
30	31	1	IRR ind i465 d1 c29	1d	1.2
30	31	2	BB p620 c33 Ri32 [30]	3d/217	0.6 + 0.8
30	31	4	BB p248 c9 Ri32 [30]	4d/3007	0.7 + 1.2
36	37	9	BB p666 c27 Ri38 [36]	3d/37	0.7 + 1.0
40	41	1	IRR ind i820 d1 c40	1d/1	1.6
40	41	3	BB p820 c30 Ri42 [40]	3d/41	1.4 + 0.3
40	41	6	BB p574 c16 Ri42 [40]	2d/41	0.9 + 1.8
42	43	5	BB p1806 c94 Ri44 [42]	2d/43	1.4 + 0.6
42	43	5	BB p903 c33 Ri44 [42]	4d/989	1.7 + 0.7
46	47	1	IRR ind i1081 d1 c45	1d	1.4
46	47	2	BB p1081 c36 Ri48 [46]	3d/658	0.6 + 1.8
46	47	2	BB p1081 c45 Ri48 [46]	3d/235	0.6 + 2.8
46	47	4	BB p1081 c36 Ri48 [46]	4d/1974	0.6 + 6.8
48	49	2	BB p1176 c36 Ri50 [24, 24]	2d/49	0.1 + 3.0
48	49	10	BB p980 c24 Ri50 [24, 24]	3d/49	0.6 + 1.1
52	53	1	IRR perm1378 c39	1d	1.5
52	53	3	BB p1378 c39 Ri54 [52]	2d/53	1.0 + 0.5
52	53	9	BB p1378 c39 Ri54 [52]	2d/53	1.0 + 1.6

$L_2(q)$, q odd, Degree $(q-1)$ [continued]

Deg	q	C	Method	N/D	Time
58	59	1	AIR ind i1771 d1 c57	1d	3.4
58	59	2	BB i1771 d1 c57 Ri60 [58]	2d/59	1.3 + 4.0
58	59	4	BB p1711 c38 Ri60 [58]	3d/531	1.4 + 3.1
58	59	4	BB p1711 c60 Ri60 [58]	4d/1771	1.5 + 2.0
60	61	15	BB p1830 c45 BB Ri62 [60]	4d/61	3.4 + 8.4
66	67	8	BB i2211 d1 c65 Ri68 [66]	3d/67	1.7 + 8.7
66	67	8	BB p2211 c51 Ri68 [66]	8d/8d	1.7 + 8.7
70	71	1	IRR perm2485 c54	2d	1.5
70	71	1	IRR ind i2485 d1 c69	2d	6.7
70	71	1	IRR ind i2485 d1 c69	2d	7.3
70	71	2	BB p2485 c54 Ri72 [70]	3d/852	2.6 + 2.8
70	71	3	BB p2982 c61 Ri72 [70]	2d/71	2.6 + 2.7
70	71	3	BB p2982 c61 Ri72 [70]	4d/7881	2.6 + 3.2
70	71	6	BB p2485 c54 Ri72 [70]	3d/852	2.5 + 6.2
72	73	18	BB p2628 c54 Ri74 [72]	4d/73	4.2 + 10.4
78	79	4	BB p4108 c81 Ri80 [78]	5d/6d	6.8 + 1.2
78	79	8	BB p3081 c60 Ri80 [78]	4d/2370	1.6 + 9.6
80	81	20	BB p3240 c60 Ri82 [40, 40]	5d/81	10.4 + 11.7
82	83	1	IRR ind i3403 d1 c81	2d/1	9.2
82	83	1	IRR perm3403 c63	2d/1	4.9
82	83	3	BB i3403 d1 c81 Ri84 [82]	4d/16351	3.4 + 5.2
82	83	3	BB i3403 d1 c81 Ri84 [82]	2d/83	3.4 + 9.6
82	83	6	BB i3403 d1 c81 Ri84 [82]	3d/83	3.4 + 11.6
82	83	6	BB p3403 c63 Ri84 [82]	8d/7d	3.4 + 9.0
88	89	1	IRR perm3916 c66	1d	2.5
88	89	2	BB p3916 c66 Ri90 [88]	2d/89	5.5 + 1.8
88	89	3	BB p3916 c66 Ri90 [88]	2d/89	5.5 + 2.2
88	89	4	BB p3916 c66 Ri90 [88]	2d/89	5.5 + 2.8
88	89	12	BB p3916 c66 Ri90 [88]	10d/8d	5.5 + 17.6
96	97	3	BB p4656 c72 Ri98 [96]	2d/97	4.3 + 9.0
96	97	21	BB p4656 c72 Ri98 [96]	5d/97	4.4 + 38.8

$L_2(q)$, q odd, Degree $(q+1)$

Deg	q	C	Method	N/D	Time
6	7	1	IRR ind i7 d2 c2	1	0.1
12	11	2	AIR ind i11 d4 c4	1	0.1 + 0.0
14	13	1	IRR ind i14 d2 c4	1	0.1
18	17	1	IRR ind i36 d1 c4	1	0.4
18	17	3	AIR ind i72 d1 c8	1d/1	0.7 + 0.3
20	19	1	IRR ind i20 d2 c4	1d	0.3
20	19	3	BB p171 p15 Ri20 $[1_3, 9]$	2d/19	0.1 + 0.3
24	23	5	BB p253 c16 Ri24 $[2_5, 22]$	2d/23	0.2 + 2.2
26	25	1	IRR ind i26 d2 c4	1	0.7
26	25	1	IRR ind i26 d2 c4	1	0.7
26	25	1	IRR ind i52 d2 c4	1	0.4
26	25	2	BB p312 c24 Ri26 $[2_2, 12, 12]$	1d/5	1.1 + 1.3
28	27	6	BB p351 c21 Ri28 $[2_6, 26]$	2d/27	0.3 + 0.4
30	29	3	BB i30 d6 Ri30 $[2_3, 28]$	2d/29	0.3 + 0.3
30	29	3	BB p420 c28 Ri30 $[2_3, 28]$	2d/29	0.3 + 0.4
32	31	1	IRR ind i32 d2 c4	1d	0.8
32	31	2	BB p160 c10 Ri32 $[2_2, 30]$	2d/31	1.2 + 0.9
32	31	4	BB p465 c24 Ri32 $[2_2, 30]$	2d/31	1.4 + 1.2
38	37	3	BB i38 d6 c12 Ri38 $[2_3, 36]$	1d/37	0.7 + 2.0
38	37	3	BB i38 d6 c12 Ri38 $[2_3, 36]$	2d/37	0.8 + 1.9
42	41	1	IRR ind i84 d1 c4	1d/1	1.0
42	41	2	BB p420 c10 Ri42 $[2_2, 40]$	2d/41	1.8 + 0.2
42	41	2	BB p210 c10 Ri42 $[2_2, 40]$	2d/123	1.4 + 0.2
42	41	4	BB i84 d4 c16 Ri42 $[2_4, 40]$	3d/205	0.5 + 4.3
44	43	1	IRR ind i44 d2 c4	1d	0.8
44	43	3	BB p308 c14 Ri44 $[2_6, 42]$	2d/43	2.0 + 0.3
44	43	6	BB p903 c33 Ri44 $[2_6, 42]$	3d/43	2.1 + 0.8

$L_2(q)$, q odd, Degree $(q+1)$ [continued]

Deg	q	C	Method	N/D	Time
48	47	11	BB p1081 c36 Ri48 $[2_{11}, 46]$	4d/47	2.4 + 3.2
50	49	1	IRR ind i50 d2 c4	3/1	0.4
50	49	2	AIR ind i175 d1 c7	2d/4	1.0 + 4.4
50	49	2	BB i100 d2 c8 Ri50 $[2_2, 24, 24]$	2d/49	1.0 + 1.4
50	49	4	BB i100 d4 c16 Ri50 $[2_4, 24, 24]$	3d/98	1.1 + 5.8
54	53	1	IRR perm54	s 1d	0.1
54	53	6	BB p702 c26 Ri54 $[2_6, 52]$	4d/53	3.0 + 3.9
54	53	6	BB i54 d12 c24 Ri108 $[2_6, 52]$	4d/53	4.0 + 5.9
60	59	14	BB p1711 c38 Ri60 $[2_{14}, 58]$	4d/59	3.0 + 7.5
62	61	1	IRR ind i62 d2 c4	1d	1.8
62	61	1	IRR ind i62 d2 c4	1d	1.9
62	61	2	BB p310 c10 Ri62 $[2_2, 60]$	2d/61	3.3 + 2.7
62	61	2	BB i310 d4 c8 Ri62 $[2_2, 60]$	3d/61	3.3 + 2.8
62	61	4	BB p930 c30 Ri62 $[2_2, 60]$	3d/61	3.5 + 6.1
62	61	4	BB i62 d8 c16 Ri62 $[2_2, 60]$	3d/549	4.6 + 2.7
68	67	1	IRR ind i68 d2 c4	2d/1	1.4
68	67	5	BB p748 c22 Ri68 $[2_5, 66]$	3d/67	1.9 + 2.2
68	67	10	BB i68 d20 c40 Ri68 $[2_{10}, 66]$	4d/67	2.3 + 6.4
72	71	2	BB i72 d4 c8 Ri72 $[2_2, 70]$	2d/71	2.7 + 2.2
72	71	3	BB p504 c14 Ri72 $[2_3, 70]$	2d/71	2.7 + 2.2
72	71	12	BB p2485 c54 Ri72 $[2_{12}, 70]$	4d/71	4.0 + 10.7
74	73	2	BB p888 c24 Ri74 $[2_2, 72]$	3d/146	4.4 + 4.3
74	73	3	BB p666 c18 Ri74 $[2_3, 72]$	2d/73	4.5 + 5.0
74	73	3	BB p888 c36 Ri74 $[2_3, 72]$	3d/438	4.4 + 4.6
74	73	6	BB p2664 c72 Ri74 $[2_6, 72]$	4d/438	4.5 + 4.1

$L_2(q)$, q odd, Degree $(q+1)$ [continued]

Deg	q	C	Method	N/D	Time
80	79	1	IRR ind i80 d2 c4	2d/1	2.8
80	79	6	BB p1040 c26 Ri80 $[2_{12}, 78]$	3d/79	1.7 + 3.7
80	79	12	BB p3081 c60 Ri80 $[2_{12}, 78]$	4d/79	3.2 + 8.6
82	81	1	IRR ind i82 d2 c4	1d	2.6
82	81	2	BB p369 c9 Ri82 $[2_2, 40, 40]$	1d/27	7.6 + 6.1
82	81	2	BB p656 c16 Ri82 $[2_2, 40, 40]$	1d/27	8.0 + 5.1
82	81	2	BB i82 d4 c8 Ri82 $[2_2, 40, 40]$	2d/27	6.6 + 4.1
82	81	4	BB p1640 c40 Ri82 $[2_4, 40, 40]$	2d/27	8.6 + 12.0
82	81	8	BB p3280 c80 Ri82 $[2_8, 40, 40]$	3d/27	10.6 + 13.6
84	83	20	BB p3403 c63 Ri84 $[2_{20}, 82]$ (p. 140)	6d/83	11.3 + 38.6
90	89	1	IRR perm360 c8	1d	1.5
90	89	5	BB i90 d10 c20 Ri90 $[2_{10}, 88]$	4d/5963	5.9 + 5.7
90	89	5	BB p990 c22 Ri90 $[2_{10}, 88]$	3d/89	6.1 + 5.5
90	89	10	BB i90 d20 c40 Ri90 $[2_{10}, 88]$	9d/7d	7.0 + 13.5
98	97	1	IRR ind i98 d2 c4	2d	4.4
98	97	1	IRR ind i98 d2 c4	2d	4.4
98	97	1	IRR ind i98 d2 c4	2d	4.4
98	97	2	BB p784 c16 Ri98 $[2_2, 96]$	2d/194	9.3 + 3.4
98	97	2	BB p1176 c24 Ri98 $[2_2, 96]$	2d/582	9.3 + 5.2
98	97	4	BB p1568 c32 Ri98 $[2_4, 96]$	4d/3007	9.3 + 13.1
98	97	4	BB p2352 c48 Ri98 $[2_4, 96]$	4d/41807	9.3 + 11.5
98	97	8	BB p4704 c96 Ri98 $[2_4, 96]$	8d/8d	10.2 + 19.4

10.3. Representations of $2.\mathrm{L}_2(q)$

$2.\mathrm{L}_2(q)$, $q \equiv 1 \pmod 4$, Degree $(q-1)/2$

Deg	q	F	S	Method	N/D	Time
6	13	2	2	AIR ind i28 d12 c112	1d/1	0.9 + 0.6
8	17	2	2	AIR ind i272 d2 c184	1d/1	2.5 + 1.3
12	25	1	2	AIR ind i65 d8 c104	1d/1	3.6 + 0.1
12	25	1	2	AIR ind i65 d8 c104	1d/1	2.9 + 0.1
14	29	2	2	BB i812 d2 c232 c56	10d/8d	20 + 7.9
18	37	2	2	IE i38 (BB p148 Ri2 $[18_2]$)	6d/333	4.2 + 0.5
20	41	2	2	BB i1640 d2 c472 Ri574 $[4_2, 4_2, 12]$	6d/6d	0.2 + 34
24	49	1	2	AIR ind i100 d24 c348	1d/1	20.7 + 1.9
24	49	1	2	AIR ind i100 d24 c348	1d/1	20.8 + 1.8
26	53	2	2	BB i2756 d2 c424 Ri108 $[13, 13]_2$	6d/689	14.5 + 45
30	61	2	2	IE i62 (BB p248 c24 Ri4 $[30_2]$)	8d/5490	15 + 1.5
36	73	2	2	IE i74 (BB i4 d72 c32 Ri72 $[36_2]$)	10d/93440	16.7 + 1.9
40	81	1	2	AIR ind i738 d8 c212	1d/1	33.5 + 0.6
40	81	1	2	AIR ind i738 d8 c212	1d/1	33.2 + 0.6
44	89	2	2	IE i90 (BB i4 d88 c32 Ri4 $[44_2]$)	11d/7d	21.7 + 5.4
48	97	2	2	[Homogeneous split too hard]	?	?

$2.\mathrm{L}_2(q)$, $q \equiv 3 \pmod 4$, Degree $(q+1)/2$

Deg	q	F	S	Method	N/D	Time
4	7	2	1	AIR perm16	1d/1	0.1 + 0.0
6	11	2	1	AIR perm24	1d/1	0.1 + 0.1
10	19	2	1	AIR perm40	1d/1	0.4 + 0.2
12	23	2	1	AIR perm48	2d/1	0.3 + 0.2
14	27	2	1	AIR perm56	1d/1	0.3 + 0.2
16	31	2	1	AIR perm64	3d/2	4.8 + 0.3
22	43	2	1	AIR ind i132 d1 c6	4d/1	2.9 + 0.6
24	47	2	1	BB p4512 c200 Ri48 $[1, 23_2]$	4d/47	2.0 + 0.6
30	59	2	1	BB p120 c4 Ri60 $[1, 29_2]$	4d/59	22.0 + 1.0
34	67	2	1	BB p136 c16 Ri68 $[1, 33_2]$	6d/67	12 + 6.7
36	71	2	1	BB p144 c32 Ri72 $[1, 35_2]$ (p. 141)	6d/71	10.7 + 0.6
40	79	2	1	BB p160 c32 Ri80 $[1, 39_2]$	6d/79	10.7 + 11
42	83	2	1	BB p168 c24 Ri84 $[1, 41_2]$	7d/83	11.7 + 14

$2.L_2(q)$, q odd, Degree $(q-1)$

Deg	q	F	S	Method	N/D	Time
6	7	2	2	AIR perm16	1d/1	0.1 + 0.0
10	11	1	2	AIR ind i11 d8 c16	1d/1	0.4 + 0.2
10	11	2	2	AIR ind i11 d8 c16	1d/1	0.4 + 0.2
12	13	3	2	BB p312 c48 Ri14 $[12_2]$	3d/13	0.8 + 0.5
16	17	1	2	AIR ind i272 d1 c32	1d/1	0.7 + 0.3
16	17	3	2	BB i272 d1 c32 Ri18 $[16_2]$	3d/17	10.9 + 1.1
18	19	1	2	AIR ind i20 d18 c40	1d/1	0.9 + 0.1
18	19	4	2	BB i20 d18 c40 Ri18 [18]	3d/418	1.2 + 1.3
22	23	2	2	BB i24 d22 c48 Ri24 [22]	2d/138	0.6 + 1.2
22	23	4	2	BB i24 d22 c48 Ri24 [22]	2d/69	0.6 + 1.6
24	25	6	2	BB i65 d8 c40 Ri26 $[12,12]_2$	6d/37100	4.8 + 2.6
26	27	1	2	AIR ind i28 d26 c56	3d/1	1.3 + 2.6
26	27	6	2	BB i28 d26 c56	1d/9	0.2 + 2.0
28	29	1	2	AIR ind i812 d1 c56	2d/3	4.5 + 0.9
28	29	2	2	BB i812 d1 c56 Ri203 $[2_4, 2_4^2, 4_2, 6_2^3]$	2d/30	2.5 + 5.5
28	29	4	2	BB i24 d22 c48 Ri30 $[28_2]$	5d/5162	2.1 + 7.9
30	31	8	2	BB i32 d30 c64 Ri32 [30]	4d/3007	2.7 + 8.8
36	37	9	2	BB p2664 c144 Ri38 $[36_2]$	5d/6d	10.1 + 25.8
40	41	1	2	BB i1640 d1 c80 Ri42 $[40_2]$	4d/7585	6.0 + 8.5
40	41	3	2	BB i1640 d1 c80 Ri42 $[40_2]$	9d/8d	10.4 + 16
40	41	6	2	BB i1640 d1 c80 Ri42 $[40_2]$	4d/1517	12 + 20
42	43	1	2	BB p3784 c184 Ri44 [42]	2d/129	12.4 + 0.2
42	43	10	2	BB p3784 c184 Ri44 [42]	5d/15179	29.1 + 11.3
46	47	4	2	BB p4512 c200 Ri48 [46]	3d/669	6.7 + 9.5
46	47	8	2	BB p4512 c200 Ri48 [46]	4d/4559	6.7 + 18.5
48	49	2	2	BB i2352 d1 c96 Ri50 $[24,24]_2$	3d/49	29 + 12.5
48	49	10	2	BB i2352 d1 c96 Ri50 $[24,24]_2$	3d/49	29 + 36.7
52	53	1	2	BB i2756 d1 c104 Ri54 $[52_2]$	3d/689	4.8 + 20
52	53	3	2	BB i2756 d1 c104 Ri108 $[52_2]$	5d/23797	5.1 + 22
52	53	9	2	BB i2756 d1 c104 Ri54 $[52_2]$	4d/25493	5.3 + 25

$2.L_2(q)$, q odd, Degree $(q-1)$ [continued]

Deg	q	F	S	Method	N/D	Time
58	59	1	2	BB p7080 c248 Ri60 [58]	3d/1003	13.7 + 17.9
58	59	2	2	BB p7080 c248 Ri60 [58]	3d/236	13.7 + 21.8
58	59	4	2	BB p7080 c248 Ri60 [58]	3d/295	9.9 + 25.4
58	59	8	2	BB p7080 c248 Ri60 [58]	5d/10561	13.2 + 31.0
60	61	15	2	BB i3660 d1 c120 Ri62 [60]	36d/36d	15 + 191
66	67	1	2	BB i68 c66 c136 Ri68 [68]	3d/335	0.9 + 6
66	67	16	2	BB i68 c66 c136 Ri68 [68]	8d/8d	0.9 + 174
70	71	2	2	BB i72 d70 c144 Ri72 [70]	4d/15549	20 + 9.4
70	71	4	2	BB p10224 c144 Ri72 [70]	3d/355	20 + 24
70	71	12	2	BB p10224 c144 Ri72 [70]	5d/98477	20 + 128
72	73	18	2	BB i5256 d1 c144 Ri296 [72]	578d/578d	0.9 + 3565
78	79	4	2	BB i80 d78 c160 Ri80 [78]	8d/8d	20 + 35
78	79	16	2	BB i80 d78 c160 Ri80 [78]	7d/7d	20 + 314
80	81	20	2	GE i82 $[40, 40]_2$	71d/143d	51 + 4047
82	83	1	2	BB i84 d82 c168 Ri84 [82]	3d/83	2.0 + 19.5
82	83	2	2	BB i84 d82 c168 Ri84 [82]	3d/166	2.0 + 24.2
82	83	6	2	BB i84 d82 c168 Ri84 [82]	5d/20833	2.0 + 69
82	83	12	2	BB i84 d82 c168 Ri84 [82]	7d/355489	2.0 + 186
88	89	1	2	BB i7832 d1 c176 Ri90 $[88_2]$	6d/6d	22 + 2.6
88	89	4	2	BB i7832 d1 c176 Ri90 $[88_2]$	5d/5874	22 + 29.0
88	89	12	2	BB i7832 d1 c176 Ri90 $[88_2]$	29d/29d	22 + 305
96	97	3	2	BB i9312 d1 c288 Ri1568 [96]	281d/281d	9 + 881
96	97	21	2	[Homogeneous split too hard]	?	?

$2.L_2(q)$, q odd, Degree $(q+1)$

Deg	q	F	S	Method	N/D	Time
8	7	1	2	AIR ind i8 d2 c2	1d/1	0.1 + 0.1
12	11	2	2	AIR ind i12 d4 c8	1d/1	0.2 + 0.6
14	13	1	2	AIR ind i28 d1 c4	1d/1	0.1 + 0.2
14	13	2	2	DI i14 d1	s 1/1	0.1
18	17	4	2	DI i18 d1	s 1/1	0.1
20	19	1	2	AIR ind i20 d2 c4	s 1/1	0.1
20	19	3	2	DI i19 d1	s 1/1	0.1
24	23	5	2	DI i24 d1	s 1/1	0.1
26	25	2	2	DI i26 d1	s 1/1	0.1
26	25	4	2	DI i26 d1	s 1/1	0.1
28	27	5	2	DI i28 d1	s 1/1	0.1
30	29	6	2	DI i30 d1	s 1/1	0.1
32	31	1	2	AIR ind i32 d2 c20	s 1d/1	3.1 + 0.1
32	31	2	2	DI i32 d1	s 1d/1	0.1
32	31	4	2	DI i32 d1	s 1d/1	0.1
38	37	1	2	AIR ind i76 d1 c4	2d/2	1.4 + 2.5
38	37	2	2	DI i38 d1	s 1/1	0.1
38	37	6	2	DI i38 d1	s 1/1	0.1
42	41	2	2	DI i42 d1	s 1/1	0.1
42	41	8	2	DI i42 d1	s 1/1	0.1
44	43	1	2	AIR ind i44 d2 c4	s 1/1	1.8 + 0.2
44	43	3	2	DI i44 d1	s 1/1	0.1
44	43	6	2	DI i44 d1	s 1/1	0.1
48	47	11	2	DI i48 d1	s 1	0.1
50	49	4	2	DI i50 d1	s 1	0.1
50	49	8	2	DI i50 d1	s 1	0.1
54	53	1	2	DI i54 d1	s 1/1	0.1
54	53	12	2	DI i54 d1	s 1/1	0.1

2.$L_2(q)$, q odd, Degree $(q+1)$ [continued]

Deg	q	F	S	Method	N/D	Time
60	59	14	2	DI i60 d1	s 1/1	0. 1
62	61	1	2	AIR ind i62 d2 c4	s 1/1	7.0 + 0.6
62	61	2	2	DI i62	s 1/1	0.3
62	61	4	2	DI i62	s 1/1	0.3
62	61	8	2	DI i62	s 1/1	0.3
68	67	1	2	AIR ind i68 d2 c4	s 1/1	5.0 + 0.6
68	67	5	2	DI i68	s 1/1	0.1
68	67	10	2	DI i68	s 1/1	0.1
72	71	2	2	DI i72 d1	s 1/1	0.4
72	71	3	2	DI i72 d1	s 1/1	0.4
72	71	12	2	DI i72 d1	s 1/1	0.4
74	73	2	2	DI i74 d1	s 1/1	0.7
74	73	4	2	DI i74 d1	s 1/1	0.7
74	73	12	2	DI i74 d1	s 1/1	0.7
80	79	6	2	DI i80 d1	s 1/1	0.6
80	79	12	2	DI i80 d1	s 1/1	0.6
82	81	4	2	DI i82 d1	s 1/1	1.2
82	81	16	2	DI i82 d1	s 1/1	2.0
84	83	20	2	DI i84 d1	s 1/1	0.8
90	89	2	2	DI i90 d1	s 1/1	1.5
90	89	20	2	DI i90 d1	s 1/1	1.5
98	97	2	2	DI i98 d1	s 1/1	5.6
98	97	20	2	DI i98 d1	s 1/1	5.6

Chapter 11

Representations of Other Kinds of Groups

11.1. Almost Simple Groups

We give a sample of some irreducible representations of groups which are almost simple. The larger cases are easily handled by induction or extension (sometimes irreducible and sometimes general) of suitable representations of quasi-simple groups which are already constructed. Sometimes when induction is used, direct induction [DI] would not yield a result over a minimal field. So to write the result over a minimal field, the expansion to $\mathbb{Q}$ of the representation over the subgroup is first constructed, then this is induced [IND] and the homogeneous result is split by the rational Meataxe; an irreducible component is then passed to BBRationalModuleSetup to set up a black-box representation $\mathscr{B}$ and the final representation is constructed from $\mathscr{B}$ by the hybrid algorithm.

Deg	Group	C	S	Method	N/D	Time
6	$U_4(2) : 2$	1	1	IRR ind i2 d6	s 1/1	0.1
12	$U_3(4) : 4$	2	1	IE i1600 (IRR perm13)	1d/1	0.1 + 0.1
12	$2.J_2.2$	1	2	IE i2016 (DI i3 d4)	1d/1	4.6 + 3.0
22	HS : 2	1	1	IRR ind i2 d22 c7	1d/1	1.0
22	$U_6(2) : S_3$	1	1	IE i672 (ei 2, ind i2 d22)	1d/1	5.7 + 0.5
28	$2.S_6(3) : 2$	1	1	IRR ind i2 d28 (p2240 c96)	1d/1	21.0
28	McL : 2	1	1	IRR ind i2 d22 c3	1d	1.1
28	$J_2 : 2$	1	1	IRR ind i2 d28 (p315 c27)	1d/1	2.9
36	$3.J_3 : 2$	2	1	IND i2 [$3.J_3$]; split;	[-]	+0.5
				BB Ri34884 $[6_2, 30]$	2d/32	53 + 0.6
65	$G_2(4) : 2$	1	1	IRR ind i2 d65 (p 416 c26)	s 1d/1	1.6
78	$Fi_{22} : 2$	1	1	DI i2 [Fi_{22}]	2d/1	+1.9
102	He : 2	1	1	IND i2 [He : $\mathbb{Q}$]; split	2d/1	+2.5
124	Sz(32) : 5	2	1	IE i1025 DI i31 d4	1d/8	10.4
143	Suz : 2	1	1	IE i2 [Suz]	2d/1	+5.3
170	$J_3 : 2$	1	1	GE J_3 [170]		+2.9
				Ri6156 $[17, 17, 68^2]$	3d/120	1.0 + 1.6
231	HS : 2	1	1	IRR ind i2 d231 c11	2d/1	30
240	$12.M_{22}.2$	4	1	IND i2 [$12.M_{22}$]; split;	[-]	+322
				BB Ri44 $[240_2]$	2d/1344	75 + 68
266	HN : 2	1	1	GE HN [266]	$2d/(2^6.5)$	5.8
429	$Fi_{22} : 2$	1	1	IE Fi_{22}	4d/20	+24
684	3.O′N : 2	2	1	GE 3.O′N $[684_2]$ (p. 111)	4d/3038	+359

11.2. Maximal Subgroups of the Monster

Out of interest, we computed a minimal-degree faithful ordinary representation of each maximal subgroup of the Monster sporadic simple group. Several of these groups have long composition length and interesting composition factors. We omitted the cases where the group is too large to compute its character table within a day or the minimal degree for a faithful representation is greater than 10000.

In the following table, the number N indicates the N-th maximal subgroup according to the numbering of [WWT$^+$], while the other fields are as for the other tables (since all the representations have Schur index 1, we omit the 'S' field).

N	Group	Deg	C	Method	N/D	Time
3	$3.\mathrm{Fi}_{24}'$	783	2	GE[$\neg\chi$] Fi$_{23}$ [1, 782]	4d/5d	+2.0h
13	$3^2 : 2 \times \mathrm{O}_8^+(3).\mathrm{S}_4$	2400	1	$\rho_8 \otimes \rho_{300}$ (RR p3369)	3d/1	63
16	$5^{1+6} : 2\mathrm{J}_2 : 4$	500	1	IE i10080		
				(GE i25 [200, 300])	3d/20	332
17	(7 : 3 × He) : 2	306	1	GE i652800 [54, 252]	2d/49	35
18	$(\mathrm{A}_5 \times \mathrm{A}_{12}) : 2$	44	1	$\rho_4 \otimes \rho_8$	s 1d	2.8
19	$5^{3+3}.(2 \times \mathrm{L}_3(5))$	3100	1	DI i31 d100 (IE i10, i3, i2)	s 2d/1	1070
20	$(\mathrm{A}_6 \times \mathrm{A}_6 \times \mathrm{A}_6).(2 \times \mathrm{S}_4)$	27	1	DI i3 d9 (IE i2025)	s 1/1	65
21	$(\mathrm{A}_5 \times \mathrm{U}_3(8){:}3_1){:}2$	224	2	$\rho_7 \otimes \rho_{56}$ (ρ_{56}: IE 513)	1d/24	50
22	$5^{2+2+4} : (\mathrm{S}_3 \times \mathrm{GL}_2(5))$	600	1	IE i10 (DI i150 d4)	s 1/1	118
23	$(\mathrm{L}_3(2) \times \mathrm{S}_4(4) : 2).2$	108	2	IE i425 (DI i18 d6)	1d/28	40
24	$7^{1+4}{:}(3 \times 2\mathrm{S}_7)$	294	1	IE i120 (DI i49 d6)	1/7	87
25	$(5^2 : [2^4] \times \mathrm{U}_3(5)).\mathrm{S}_3$	480	2	DI i24 d20 (IE i175)	s 2d/6	125
26	$(\mathrm{L}_2(11) \times \mathrm{M}_{12}) : 2$	110	2	DI i2 d55 (IE i144)	2d/33	20
27	$(\mathrm{A}_7 \times (\mathrm{A}_5 \times \mathrm{A}_5) : 2^2) : 2$	48	1	IRR perm140	s 1/1	12
28	$5^4 : (3 \times 2\mathrm{L}_2(25)) : 2\mathrm{b}$	624	1	DI i156 d4 (RR p5)	s 1/1	10
29	$7^{2+1+2} : \mathrm{GL}_2(7)$	336	1	IE i3 (i2, i7)	s 1/1	14
30	$\mathrm{M}_{11} \times \mathrm{A}_6.2^2$	90	1	IRR perm110 c6	s 1/1	1.4
31	$(\mathrm{S}_5 \times \mathrm{S}_5 \times \mathrm{S}_5) : \mathrm{S}_3$	12	1	IE i216 (i4, i4, i5, i2)	1/1	0.4
32	$(\mathrm{L}_2(11) \times \mathrm{L}_2(11)) : 4$	20	1	IRR perm110 c11	1/1	0.3
33	$13^2 : 2\mathrm{L}_2(13).4$	168	1	IRR perm338 c16	s 1/1	1.6
34	$(7^2 : (3 \times 2\mathrm{A}_4) \times \mathrm{L}_2(7)).2$	144	2	DI i16 d9 (AIRp392 c60)	1d/1	29
35	$(13 : 6 \times \mathrm{L}_3(3)).2$	144	1	IRR perm338 c21	1d/1	1.8
36	$13^{1+2} : (3 \times 4\mathrm{S}_4)$	156	1	IRR perm2197 c11	2d/1	18
37	$\mathrm{L}_2(71)$	35	2	BB i2485 d1 c69 Ri72 [35_2]	6d/71	12
38	$\mathrm{L}_2(59)$	29	2	BB p1771 c45 Ri60 [29_2]	5d/59	3.8
39	$11^2 : (5 \times 2\mathrm{A}_5)$	120	1	IRR perm121 c16	s 1/1	0.4
40	$\mathrm{L}_2(29){:}2$	28	1	IRR ind i2 d28 c7	1d/1	0.5
41	$7^2 : \mathrm{SL}_2(7){:}2$	48	1	IRR perm49 c4	s 1/1	0.1
42	$\mathrm{L}_2(19){:}2$	18	1	IRR ind i2 d18 c6	1d/1	0.3
43	41:40	40	1	IRR perm41 c2	s 1/1	0.1

11.3. Representations of some Perfect Groups

In Table 1 of [DD10], some constructed representations of perfect groups are listed. We have computed the same representations by our algorithms, which are described by the following table. See the reference for details on the groups.

Deg	$\|G\|$	C	S	Method	N/D	Time
16	1920	2	1	AIR perm240 c51	1d/1	0.8
24	7680	2	1	AIR perm160 c14	1d/1	2.0
30	15000	4	1	BB p600 c56 Ri6 $[10^{1+1+1}]_4$	2d/25	3.1
32	23040	4	1	BB RR $d5 \otimes d32$ c20 Ri6 $[8_4, 24_2]$	2d/20	4.3
56	115248	6	1	BB p1176 c56 Ri49 $[3, 4, 6, 7, 7, 8, 8, 12]_6$	2d/336	9.9
64	129024	2	1	AIR ind i72 d2 c4	1d/1	3.5
48	645120	1	1	$\rho_6 \otimes \rho_8$	1/1	4.4

Bibliography

[ABH10] Martin Albrecht, Gregory Bard, and William Hart. Algorithm 898: Efficient multiplication of dense matrices over GF(2). *ACM Trans. Math. Softw.*, 37(1), 2010.

[ABM99] John Abbott, Manuel Bronstein, and Thom Mulders. Fast deterministic computation of determinants of dense matrices. In *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation (Vancouver, BC)*, pages 197–204 (electronic), New York, 1999. ACM.

[AHU75] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The design and analysis of computer algorithms.* Addison-Wesley Publishing Co., Reading, Mass.-London-Amsterdam, 1975. Second printing, Addison-Wesley Series in Computer Science and Information Processing.

[Bau91] Ulrich Baum. Existence and efficient construction of fast Fourier transforms on supersolvable groups. *Comput. Complexity*, 1(3):235–256, 1991.

[BC94] Ulrich Baum and Michael Clausen. Computing irreducible representations of supersolvable groups. *Math. Comp.*, 63(207):351–359, 1994.

[BC03] John N. Bray and Robert T. Curtis. Monomial modular representations and symmetric generation of the harada-norton group. *Journal of Algebra*, 268(2):723 - 743, 2003.

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[BEO01] Hans Ulrich Besche, Bettina Eick, and E. A. O'Brien. The groups of order at most 2000. *Electron. Res. Announc. Amer. Math. Soc.*, 7:1–4 (electronic), 2001.

[BF91] László Babai and Katalin Friedl. Approximate representation theory of finite groups. In *32nd Annual Symposium on Foundations of Computer Science (San Juan, PR, 1991)*, pages 733–742. IEEE Comput. Soc. Press, Los Alamitos, CA, 1991.

[Bli05] H. F. Blichfeldt. The finite, discontinuous primitive groups of collineations in four variables. *Math. Ann.*, 60(2):204–231, 1905.

[Bli07] H. F. Blichfeldt. The finite, discontinuous primitive groups of collineations in three variables. *Math. Ann.*, 63(4):552–572, 1907.

[BN70] C. Brott and J. Neubüser. A programme for the calculation of characters and representations of finite groups. In J. Leech, editor, *Computational problems in abstract algebra.* Oxford - Pergamon, 1970.

[Bög93] S. Böge. Realisierung (p - 1)-dimensionaler Darstellungen von PSL(2, p). *Arch. Math.*, 60:121–127, 1993.

[BR90] László Babai and Lajos Rónyai. Computing irreducible representations of finite groups. *Math. Comp.*, 55(192):705–722, 1990.

[Bra67] Richard Brauer. Über endliche lineare Gruppen von Primzahlgrad. *Math. Ann.*, 169:73–96, 1967.

[Brü98] Herbert Brückner. *Algorithmen für endliche auflösbare Gruppen und Anwendung.* PhD thesis, Zugl.: Aachen, Techn. Hochsch., Diss., 1998, Aachen, 1998.

[BS92] Dave Bayer and Mike Stillman. Computation of Hilbert functions. *Journal of Symbolic Computation*, 14(1):31 – 50, 1992.

[BW93] Thomas Becker and Volker Weispfenning. *Gröbner Bases.* Graduate Texts in Mathematics. Springer, New York–Berlin–Heidelberg, 1993.

[CC82] T.W.J. Chou and G.E. Collins. Algorithms for the solution of systems of linear diophantine equations. *SIAM J. Computing*, 11(4):687–708, 1982.

[CCN$^+$85] J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker, and R. A. Wilson. *Atlas of finite groups.* Oxford University Press, Eynsham, 1985. Maximal subgroups and ordinary characters for simple groups, With computational assistance from J. G. Thackray.

[CH04] J. J. Cannon and D.F. Holt. Computing maximal subgroups of finite groups. *J. Symbolic Comp.*, 37(5):589–609, 2004.

[CHSS05] John J. Cannon, Derek F. Holt, Michael Slattery, and Allan K. Steel. Computing subgroups of bounded index in a finite group. *J. Symbolic Comput.*, 40(2):1013–1022, 2005.

[CIW97] Arjeh M. Cohen, Gábor Ivanyos, and David B. Wales. Finding the radical of an algebra of linear transformations. *J. Pure Appl. Algebra*, 117/118:177–193, 1997. Algorithms for algebra (Eindhoven, 1996).

[CLG97] Frank Celler and Charles R. Leedham-Green. Calculating the order of an invertible matrix. In Larry Finkelstein and William M. Kantor, editors, *Groups and Computation II*, volume 28 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 55–60. AMS, 1997.

[CLGM$^+$95] Frank Celler, Charles R. Leedham-Green, Scott H. Murray, Alice C. Niemeyer, and E. A. O'Brien. Generating random elements of a finite group. *Comm. Algebra*, 23(13):4931–4948, 1995.

[CLO96] David Cox, John Little, and Donal O'Shea. *Ideals, Varieties and Algorithms.* Undergraduate Texts in Mathematics. Springer, New York–Berlin–Heidelberg, 2nd edition, 1996.

[Coh93] Henri Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Graduate Texts in Mathematics.* Springer, Berlin–Heidelberg–New York, 1993.

[CP96] John Cannon and Catherine Playoust. MAGMA: a new computer algebra system. *Euromath Bull.*, 2(1):113–144, 1996.

[CR81] Charles W. Curtis and Irving Reiner. *Methods of representation theory. Vol. I.* John Wiley & Sons Inc., New York, 1981. With applications to finite groups and orders, Pure and Applied Mathematics, A Wiley-Interscience Publication.

[CR87] Charles W. Curtis and Irving Reiner. *Methods of representation theory. Vol. II.* Pure and Applied Mathematics (New York). John Wiley & Sons Inc., New York, 1987. With applications to finite groups and orders, A Wiley-Interscience Publication.

[DA03] Vahid Dabbaghian-Abdoly. *An algorithm to construct representations of finite groups.* PhD thesis, Ottawa, Ont., Canada, Canada, 2003. AAINQ83515.

[DA05] Vahid Dabbaghian-Abdoly. An algorithm for constructing representations of finite groups. *J. Symbolic Comput.*, 39(6):671–688, 2005.

[Dab08] Vahid Dabbaghian. Repsn: A gap4 package for constructing representations of finite groups. URL: http://www.gap-system.org/Packages/repsn.html, 2008.

[DD10] V. Dabbaghian and J. D. Dixon. Computing matrix representations. *Math. Comp.*, 79(271):1801–1810, 2010.

[DER84] I.S. Duff, A.M. Erisman, and J.K. Reid. *Direct methods for sparse matrices.* Monographs on Numerical Analysis. Oxford University Press, 1984.

[DG93] J. D. Dixon and H. Gollan. Computing primitive linear groups of small order. 1993.

[Dix70] John D. Dixon. Computing irreducible representations of groups. *Math. Comp.*, 24:707–712, 1970.

[Dix82] John D. Dixon. Exact solution of linear equations using p-adic expansions. *Numer. Math.*, 40(1):137–141, 1982.

[Dix93] John D. Dixon. Constructing representations of finite groups. In *Groups and computation (New Brunswick, NJ, 1991)*, volume 11 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 105–112. Amer. Math. Soc., Providence, RI, 1993.

[DPW05] Jean-Guillaume Dumas, Clément Pernet, and Zhendong Wan. Efficient computation of the characteristic polynomial. In *Proceedings of the 2005 international symposium on Symbolic and algebraic computation*, ISSAC '05, pages 140–147, New York, NY, USA, 2005. ACM.

[DZ98] J. D. Dixon and A. E. Zalesskii. Finite primitive linear groups of prime degree. *J. London Math. Soc. (2)*, 57(1):126–134, 1998.

[DZ08] J. D. Dixon and A. E. Zalesskii. Corrigendum: "Finite primitive linear groups of prime degree" [J. London Math. Soc. (2) **57** (1998), no. 1, 126–134]. *J. Lond. Math. Soc. (2)*, 77(3):808–812, 2008.

[EHV92] D. Eisenbud, C. Huneke, and W. Vasconcelas. Direct methods for primary decomposition. *Inventiones Mathematicae.*, 110:207–235, 1992.

[Fau99] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *Journal of Pure and Applied Algebra*, 139 (1-3):61–88, 1999.

[FB67] S. Flodmark and E. Blokker. A computer program for calculation of irreducible representations of finite groups. *International Journal of Quantum*

Chemistry, pages 703–11, 1967.

[Fie09] Claus Fieker. Minimizing representations over number fields. II. Computations in the Brauer group. *J. Algebra*, 322(3):752–765, 2009.

[Fie11] Claus Fieker. Private communication. 2011.

[GG90] H. Gollan and J. Grabmeier. Algorithms in Representation Theory and their Realization in the ComputerAlgebra System Scratchpad. *Bayreuther Mathem. Schriften*, Heft 33:1–23, 1990. ISSN 0172-1062.

[Gra06] Markus Grassl. Constructing matrix representations of finite groups in characteristic zero. In *Proceedings 10th Rhine Workshop on Computer Algebra (RWCA06)*, pages 143–148, Basel, March 2006.

[GTZ88] Patrizia M. Gianni, Barry M. Trager, and Gail Zacharias. Grbner bases and primary decomposition of polynomial ideals. *J. Symb. Comput.*, pages 149–167, 1988.

[HHR93] George Havas, Derek F. Holt, and Sarah Rees. Recognizing badly presented z-modules. *Linear Algebra and its Applications*, 192:137–164, 1993.

[HM01] Gerhard Hiss and Gunter Malle. Low-dimensional representations of quasi-simple groups. *LMS J. Comput. Math.*, 4:22–63 (electronic), 2001.

[HM02] Gerhard Hiss and Gunter Malle. Corrigenda: "Low-dimensional representations of quasi-simple groups" [LMS J. Comput. Math. 4 (2001), 22–63; MR1835851 (2002b:20015)]. *LMS J. Comput. Math.*, 5:95–126 (electronic), 2002.

[Hol98] Derek F. Holt. The Meataxe as a tool in computational group theory. In *The atlas of finite groups: ten years on (Birmingham, 1995)*, volume 249 of *London Math. Soc. Lecture Note Ser.*, pages 74–81. Cambridge Univ. Press, Cambridge, 1998.

[HP89] D.F. Holt and W. Plesken. *Perfect Groups.* Oxford University Press, 1989.

[HR94] Derek F. Holt and Sarah Rees. Testing modules for irreducibility. *J. Austral. Math. Soc. Ser. A*, 57(1):1–16, 1994.

[Hup98] Bertram Huppert. *Character theory of finite groups*, volume 25 of *de Gruyter Expositions in Mathematics.* Walter de Gruyter & Co., Berlin, 1998.

[HW76] W. C. Huffman and D. B. Wales. Linear groups of degree eight with no elements of order seven. *Illinois J. Math.*, 20(3):519–527, 1976.

[HW78] W. C. Huffman and D. B. Wales. Linear groups of degree nine with no elements of order seven. *J. Algebra*, 51(1):149–163, 1978.

[Isa06] I. Martin Isaacs. *Character theory of finite groups.* AMS Chelsea Publishing, Providence, RI, 2006. Corrected reprint of the 1976 original [Academic Press, New York; MR0460423].

[Jac89] Nathan Jacobson. *Basic algebra. II.* W. H. Freeman and Company, New York, second edition, 1989.

[Jan66] G. J. Janusz. Primitive idempotents in group algebras. *Proc. Amer. Math. Soc.*, 17:520–523, 1966.

[KB79] R. Kannan and A. Bachem. Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix. *SIAM J. Computing*, 9:499–507, 1979.

[KP02] James Kuzmanovich and Andrey Pavlichenkov. Finite groups of matrices whose entries are integers. *The American Mathematical Monthly*, 109(2):173–186, 2002.

[LaM91] B.A. LaMacchia. Basis reduction algorithms and subset sum problems. Sm thesis, Dept. of Elect. Eng. and Comp. Sci., Massachusetts Institute of Technology, 1991. URL:http://www.farcaster.com/papers/sm-thesis/index.htm.

[Lin71] J. H. Lindsey, II. Finite linear groups of degree six. *Canad. J. Math.*, 23:771–790, 1971.

[LLL82] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.

[LP10] Klaus Lux and Herbert Pahlings. *Representations of Groups.* Cambridge University Press, 2010.

[LS03] Klaus M. Lux and Magdolna Szőke. Computing homomorphism spaces between modules over finite dimensional algebras. *Experiment. Math.*, 12(1):91–98, 2003.

[Lüb02] F. Lübeck. On the computation of elementary divisors of integer matrices. *J. Symbolic Comp.*, 33:57–65, 2002.

[Lux97] Klaus Lux. *Algorithmic methods in modular representation theory.* Habilitationsschrift, RWTH Aachen, 1997.

[LW98] Klaus Lux and Markus Wiegelmann. Condensing tensor product modules. In *The atlas of finite groups: ten years on (Birmingham, 1995)*, volume 249 of *London Math. Soc. Lecture Note Ser.*, pages 174–190. Cambridge Univ. Press, Cambridge, 1998.

[Min96] Torsten Minkwitz. Extensions of irreducible representations. *Appl. Algebra Engrg. Comm. Comput.*, 7(5):391–399, 1996.

[MNRW02] Jürgen Müller, Max Neunhöffer, Frank Röhr, and Robert Wilson. Completing the Brauer trees for the sporadic simple Lyons group. *LMS J. Comput. Math.*, 5:18–33 (electronic), 2002.

[Mon04] Michael Monagan. Maximal quotient rational reconstruction: an almost optimal algorithm for rational reconstruction. In *Proceedings of the 2004 international symposium on Symbolic and algebraic computation*, ISSAC '04, pages 243–249, New York, NY, USA, 2004. ACM.

[MR99] Jürgen Müller and Jens Rosenboom. Condensation of induced representations and an application: the 2-modular decomposition numbers of Co_2. In *Computational methods for representations of groups and algebras (Essen, 1997)*, volume 173 of *Progr. Math.*, pages 309–321. Birkhäuser, Basel, 1999.

[Mül04] Jüergen Müller. Computation representation theory: Remarks on condensation. Preprint, RWTH Aachen, `www.math.rwth-aachen.de/~Juergen.Mueller/preprints/jm102.pdf`, 2004.

[MW01] Daniele Micciancio and Bogdan Warinschi. A linear space algorithm for computing the Hermite normal form. In *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation*, pages 231–236 (electronic), New York, 2001. ACM.

[Nic06] S.J. Nickerson. *An atlas of characteristic zero representations.* PhD thesis, University of Birmingham, 2006. URL: `http://www.maths.qmul.ac.uk/~raw/SJNphd.pdf`.

[Noe07] Felix Noeske. Tackling the generation problem in condensation. *J. Algebra*, 309(2):711–722, 2007.

[NS09a] Gabriele Nebe and Allan Steel. Recognition of division algebras. *J. Algebra*, 322(3):903–909, 2009.

[NS09b] Phong Q. Nguyen and Damien Stehlé. An lll algorithm with quadratic complexity. *SIAM Journal on Computing*, 39(3):874–903, 2009.

[NVe09] P. Q. Nguyen and B. Vallée (editors). *The LLL Algorithm: Survey and Applications.* Information Security and Cryptography. Springer-Verlag, 2009.

[O'B06] E. A. O'Brien. Towards effective algorithms for linear groups. In *Finite geometries, groups, and computation*, pages 163–190. Walter de Gruyter, Berlin, 2006.

[Par84] R. A. Parker. The computer calculation of modular characters (the meat-axe). In *Computational group theory (Durham, 1982)*, pages 267–274. Academic Press, London, 1984.

[Par98] Richard A. Parker. An integral meataxe. In *The atlas of finite groups: ten years on (Birmingham, 1995)*, volume 249 of *London Math. Soc. Lecture Note Ser.*, pages 215–228. Cambridge Univ. Press, Cambridge, 1998.

[Per95] Martin Pergler. Complex representations of $\mathrm{GL}(2, q)$. *C. R. Math. Rep. Acad. Sci. Canada*, 17(5):207–212, 1995.

[Poh93] Michael E. Pohst. *Computational algebraic number theory*, volume 21 of *DMV Seminar.* Birkhäuser Verlag, Basel, 1993.

[PS83] Ilya Piatetski-Shapiro. *Complex representations of* $\mathrm{GL}(2, K)$ *for finite fields* K, volume 16 of *Contemporary Mathematics.* American Mathematical Society, Providence, R.I., 1983.

[PS96] Wilhelm Plesken and Bernd Souvignier. Constructing rational representations of finite groups. *Experiment. Math.*, 5(1):39–47, 1996.

[PS97] W. Plesken and B. Souvignier. Analysing finitely presented groups by constructing representations. *J. Symbolic Comput.*, 24(3-4):335–349, 1997. Computational algebra and number theory (London, 1993).

[PS98] W. Plesken and B. Souvignier. Constructing representations of finite groups and applications to finitely presented groups. *J. Algebra*, 202(2):690–703, 1998.

[Püs02] Markus Püschel. Decomposing monomial representations of solvable groups. *J. Symbolic Comput.*, 34(6):561–596, 2002.

[Rei03] Irving Reiner. *Maximal Orders*, volume 28 of *LMS Monographs*. Oxford University Press, 2003.

[Ros10] Tobias Rossmann. Irreducibility testing of finite nilpotent linear groups. *Journal of Algebra*, 324(5):1114 – 1124, 2010. Computational Algebra.

[Ryb90] A. J. E. Ryba. Computer condensation of modular representations. *J. Symbolic Comput.*, 9(5-6):591–600, 1990. Computational group theory, Part 1.

[Sch04] I. Schur. Über die Darstellung der endlichen Gruppen durch gebrochene lineare Substitutionen. 127:20–50, 1904.

[Sch11] I. Schur. Über die Darstellung der symmetrischen und der alternierenden Gruppe durch gebrochene lineare Substitutionen. 139:155–250, 1911.

[Sch02] T. Schulz. *Konstruktion rationaler Darstellungen endlicher Gruppen (Construction of rational representations of finite groups).* PhD thesis, RWTH Aachen, 2002.

[SE91] C.-P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. In *Fundamentals of computation theory (Gosen, 1991)*, volume 529 of *Lecture Notes in Comput. Sci.*, pages 68–85. Springer, Berlin, 1991.

[Sey93] M. Seysen. Simultaneous reduction of a lattice basis and its reciprocal basis. *Combinatorica*, 13(3):363–376, 1993.

[Sim05] Denis Simon. Solving quadratic equations using reduced unimodular quadratic forms. *Math. Comp.*, 74(251):1531–1543 (electronic), 2005.

[Smi61] Henry John Stephen Smith. On systems of linear indeterminate equations and congruences. *Philosophical Transactions of the Royal Society*, 151:293–326, 1861.

[Sou09] Bernd Souvignier. Decomposing homogeneous modules of finite groups in characteristic zero. *J. Algebra*, 322(3):948–956, 2009.

[Ste97] Allan Steel. A new algorithm for the computation of canonical forms of matrices over fields. *J. Symbolic Comput.*, 24(3-4):409–432, 1997. Computational algebra and number theory (London, 1993).

[Ste09] Damien Stehlé. *Floating-point LLL: theoretical and practical aspects*, chapter 5, pages 179–214. Information Security and Cryptography. Springer-Verlag, 2009.

[Ste11] Allan Steel. Ordinary representations of finite groups [webpage]. URL: http://magma.maths.usyd.edu.au/users/allan/reps/ or http://tinyurl.com/OrdReps, 2011.

[Str69] Volker Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969.

[SW97] Ibrahim A. I. Suleiman and Robert A. Wilson. The 2-modular characters of Conway's third group Co_3. *J. Symbolic Comput.*, 24(3-4):493–506, 1997.

[Sze99] Fernando Szechtman. Weil representations of unitary groups. *J. Algebra*, 221(1):161–187, 1999.

[Tan67] Shun'ichi Tanaka. Construction and classification of irreducible representations of special linear group of the second order over a finite field. *Osaka J. Math.*, 4:65–84, 1967.

[Tha81] J.G. Thackray. *Modular representations of some finite groups.* PhD thesis, University of Cambridge, 1981.

[Tra76] Barry M. Trager. Algebraic factoring and rational function integration. In R.D. Jenks, editor, *Proc. SYMSAC '76*, pages 196–208. ACM press, 1976.

[Ung06] W. R. Unger. Computing the character table of a finite group. *J. Symbolic Comput.*, 41(8):847–862, 2006.

[Ung09] William Unger. An algorithm for computing Schur indices of characters. Submitted to J. Algebra (Section on Computational Algebra), 2009.

[Ung10] William Unger. Private communication. 2010.

[vzGG03] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra.* Cambridge University Press, Cambridge, second edition, 2003.

[Wal68] David B. Wales. Finite linear groups in seven variables. *Bull. Amer. Math. Soc.*, 74:197–198, 1968.

[Wal69] David B. Wales. Finite linear groups of degree seven. I. *Canad. J. Math.*, 21:1042–1056, 1969.

[Wha] R. Clint Whaley. Automatically tuned linear algebra software (atlas). http://math-atlas.sourceforge.net/.

[Wil96] Robert A. Wilson. Standard generators for sporadic simple groups. *J. Algebra*, 184(2):505–515, 1996.

[Wil98a] Robert A. Wilson. An atlas of sporadic group representations. In *The atlas of finite groups: ten years on (Birmingham, 1995)*, volume 249 of *London Math. Soc. Lecture Note Ser.*, pages 261–273. Cambridge Univ. Press, Cambridge, 1998.

[Wil98b] Robert A. Wilson. A representation for the Lyons group in $\mathrm{GL}_{2480}(4)$, and a new uniqueness proof. *Arch. Math. (Basel)*, 70(1):11–15, 1998.

[Wil99] Robert A. Wilson. Construction of finite matrix groups. In *Computational methods for representations of groups and algebras (Essen, 1997)*, volume 173 of *Progr. Math.*, pages 61–83. Birkhäuser, Basel, 1999.

[Wil02] Robert A. Wilson. Condensation. Unpublished notes, University of Sydney, 2002.

[Wil09] Robert A. Wilson. *The finite simple groups*, volume 251 of *Graduate Texts in Mathematics.* Springer-Verlag London Ltd., London, 2009.

[WP05] R. Clint Whaley and Antoine Petitet. Minimizing development and maintenance costs in supporting persistently optimized BLAS. *Software: Practice and Experience*, 35(2):101–121, February 2005. http://www.cs.utsa.edu/~whaley/papers/spercw04.ps.

[WWT+] R. Wilson, P. Walsh, J. Tripp, I. Suleiman, S. Rogers, R. Parker, S. Norton, S. Nickerson, S. Linton, J. Bray, and R. Abbott. Atlas of finite group representations. URL: http://brauer.maths.qmul.ac.uk/Atlas/v3.