Journal of Algebra ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Algebra

www.elsevier.com/locate/jalgebra

Construction of the irreducible modular representations of a finite group

John J. Cannon, Allan K. Steel, William R. Unger [*,1]

School of Mathematics and Statistics, University of Sydney, Sydney, Australia

ARTICLE INFO

Article history:
Received 18 December 2018
Available online xxxx
Communicated by Derek Holt

Dedicated to the memory of Charles Sims who opened our eyes to the power of computation in group theory

Keywords:
Representation theory
Brauer characters
Irreducible modules
Finite groups

ABSTRACT

A complete procedure is described for constructing the irreducible KG-modules and their Brauer characters, where K is a finite field of characteristic p and G is a finite permutation or matrix group. The central idea is to construct a sequence $\{S_1, \ldots, S_n\}$ of KG-modules, each having relatively small dimension, such that each S_i has one or more irreducible constituents that are not constituents of $S_1, \ldots, S_{i-1}$. The Meataxe, used in conjunction with condensation, is used to extract the new irreducibles from each S_i. The algorithm has been implemented in MAGMA and is capable of constructing irreducibles of dimension over 200 000.

1. Introduction

An important goal in computational representation theory is to find algorithms for constructing the Brauer characters and irreducible KG-modules for a finite group G, where K is a finite field of characteristic p (a prime). We present an algorithm that constructs all of the irreducible KG-modules, together with the corresponding Brauer

* Corresponding author.
E-mail address: william.unger@sydney.edu.au (W.R. Unger).

[1] The authors acknowledge the support of Australian Research Council grants DP0452427, DP0772386, DP1096599 and DP130104534.

https://doi.org/10.1016/j.jalgebra.2019.08.015
0021-8693/

characters, given only a set of permutation or matrix generators for G. A variation constructs all of the irreducible modules belonging to a specified p-block.

While a number of papers [10,35] describe approaches to constructing all complex irreducibles, there is little published on the problem of constructing the set of all p-modular irreducibles for an arbitrary group. The reason for this may partly be due to the fact that the table of ordinary characters of G can be constructed quickly without the need to construct any irreducibles but this is currently not the case for the table of Brauer characters unless G is soluble. For soluble groups the irreducible KG-modules can be obtained using an algorithm originally due to Schur which constructs the irreducibles by inductively working up a composition series for G. The modules for the i-th term of the composition series are obtained by applying induction and extension to the modules constructed for the $(i-1)$-th term [4]. In the case of arbitrary non-soluble groups, no effective algorithm is known for directly constructing the p-modular irreducibles. Approaches such as those used in MOC [13] require considerable user direction as well as the explicit construction of some irreducible KG-modules.

A number of papers describe the construction of a specific irreducible KG-module M for some group of interest [28,41,29,12,14]. In most cases a KG-module S having M as a constituent is constructed by applying theoretical knowledge as well as specific knowledge of the structure of G. Then the irreducible module M is extracted for S using the Meataxe algorithm [27,16] to split S into its irreducible constituents. The module S is typically constructed from known KG-modules by applying operations such as tensoring, induction and extension. The latter two operations require some knowledge of the subgroup structure of G.

Our approach is based on the Burnside-Steinberg-Brauer Theorem [3] (referred to here as the BSB Theorem) which states that given a faithful KG-module M, then each irreducible KG-module occurs as a constituent of some tensor power of M. A key assumption is that conjugacy classes and subgroups of G are easily computed, something which is made possible through the notions of base and strong generating set (BSGS) as conceived by Charlie Sims. The algorithm makes extensive use of the table of ordinary characters and although the table of Brauer characters is not known at the outset, use is made of individual Brauer characters. The actual construction of the irreducible modules makes use of techniques mentioned in the previous paragraph as well some new ones.

The algorithm has been implemented in Magma [2] and it is capable of determining all of the irreducible KG-modules for a wide collection of groups. For example, it can compute the Brauer characters for all of the simple groups and most of the "simple groups with decorations" appearing in [18]. But its range extends far beyond this collection. For example, applying the algorithm to the Rudvalis, Suzuki and third Conway sporadic groups produces the Brauer characters for all modular characteristics in 110, 287 and 105 hours respectively, when the algorithm is boosted by parallelising the linear algebra.

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

This algorithm has many potential applications including:

- Identification and correction of errors in the tables of Brauer characters appearing in [18]. Researchers close to that project have indicated that they expect a significant number of errors to surface.
- Expansion of the collection of tables of Brauer characters appearing in [18] to a much wider body of groups.
- There is interest in having irreducible modules/Brauer characters tabulated for the maximal subgroups of the simple groups listed in the Atlas of Finite Groups.
- There is a range of areas, including some outside of mathematics, where specific irreducible KG-modules need to be constructed.

2. Overview of the algorithm

It is convenient to construct absolutely irreducible KG-modules since the irreducible modules over $GF(p^n)$, for a positive integer n, can be obtained from the absolutely irreducible modules with little effort. While the method described in this paper will work well for soluble groups, the Schur method [4] may be faster in some cases. Consequently, our primary interest is to devise an effective method for non-soluble groups.

The algorithm proceeds by constructing a sequence

$$\{S_1, \ldots, S_n\}$$

of KG-modules, each having dimension as small as possible, such that each S_i has one or more irreducible constituents that are not constituents of $S_1, \ldots, S_{i-1}$. The subsequent terms include tensor products of earlier terms. For efficiency reasons subsequent terms may also be constructed using induction of KH-modules, H a subgroup of G, and p-reduction of $\mathbb{Q}G$-modules. Since the number of absolutely irreducible KG-modules is equal to the number of p-regular classes of G, there is a natural termination condition. As each new irreducible is found its Brauer character is calculated. Thus, the algorithm returns both the irreducible KG-modules and the corresponding Brauer characters. The two main computational problems are finding good candidates for the modules S_i, and splitting them into irreducible modules.

The first approach to identifying a suitable source S_i of new irreducibles is to examine tensor products of the known irreducibles. Since their Brauer characters are known it is straightforward to determine if such a tensor product has an irreducible constituent that is new. In a similar way the induction of KH-modules, H a subgroup of G, is used to generate candidates for S_i. The second approach for finding a suitable S_i is to consider a quotient of a p-reduction of the irreducible $\mathbb{Q}G$-module T corresponding to some ordinary character χ. The p-reduction of χ is used to determine whether T will yield new irreducibles. An algorithm previously developed by A. Steel [35, Sec. 3.5] constructs a $\mathbb{Q}G$-module T affording a given irreducible rational character χ by extracting it

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

from a virtual permutation or induced module T using condensation. An important advance described in this paper is a variant of this algorithm which computes a non-trivial submodule of a p-reduction T_p of T without explicitly constructing the $\mathbb{Q}G$-module T. A key feature is the use of the ordinary characters of G to identify a submodule of the condensed version of T which corresponds to a non-trivial submodule of T_p.

The modules S_i are split using the Meataxe algorithm [16] in conjunction with condensation [29]. Because the dimension of a module S_i will often be much larger than the dimensions of its new irreducible constituents, the time spent splitting the S_i can dominate runtimes.

In order to construct KG-modules S_i that have new irreducibles as constituents we need to be able to compute considerable information about the structure of potentially very large groups G and for this we use the large body of algorithms based on the notions of *base* and *strong generating set* (BSGS) that were introduced in 1967 and 1970 by Charlie Sims [32,33]. Let G be generated by permutations acting on the set Ω. A base for G is a sequence $B = [\beta_1, \ldots, \beta_k]$ of distinct points of Ω with the property that only the identity of G fixes B pointwise. A *strong generating set* for G is a set of generators that includes generators for each stabilizer

$$G \supseteq G_{\beta_1} \supseteq G_{\beta_1,\beta_2} \ldots \supseteq G_{\beta_1,\ldots,\beta_k} = 1.$$

It is easy to see that if a BSGS is known for G then we can immediately deduce its order. Further, given a permutation π acting on Ω, it is straightforward to decide if π is an element of G. If so we can write π as a word in the strong generators of G. Building on this a range of algorithms for computing many aspects of group structure has been constructed. Descriptions of many of these algorithms can be found in the Handbook of Computational Group Theory [15]. In particular, algorithms based on the BSGS concept make it possible to compute with an arbitrary subgroup [33], find the conjugacy classes [8,34], determine some or all of the subgroup structure [7,5,6,9], and construct the table of ordinary characters [11,31,40]. For groups G where the length of its base B is small compared to its degree, these algorithms can be quite fast. Finally, efficient algorithms have been developed for constructing the BSGS for quite large groups. While Sims introduced the BSGS in the context of permutation groups, it was soon extended to matrix groups by Butler and Cannon. The BSGS concept also plays an important role in algorithms for testing isomorphism and finding the automorphism group of combinatorial objects such as graphs, codes [20], designs and Hadamard matrices etc.

Knowledge of the appropriate character table of G is central to finding an efficient strategy for constructing a given irreducible. In the case of arbitrary groups there are very efficient methods due to Dixon [11], Schneider [31], and Unger [40] available for computing the table of ordinary characters. For soluble groups the table of irreducible Brauer characters is easily computed using the Fong-Swan Theorem [25, Theorem 10.1].

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

However, in the case of a non-soluble group, no effective algorithm is known for directly constructing the p-modular character table.

3. Some background

In this section we state some results which will be needed later. Let G be a finite group, p a prime and p^a the p-part of $|G|$. The symbol G^0 will be used to denote the set of p-regular elements of G. If χ is an ordinary character of G the symbol χ^0 will denote the restriction of χ to G^0. It is assumed that the table of ordinary characters of G together with the character table of any of its subgroups can be computed.

3.1. The Burnside-Steinberg-Brauer Theorem

Theorem 1 *(Burnside-Steinberg-Brauer). Let K be a field of characteristic $p \geq 0$ and let M be a faithful KG-module. Let α be the Brauer character of M and suppose α takes on n distinct values. Then every absolutely irreducible p-modular KG-module occurs as a constituent in some tensor power $M^{\otimes k}$, with $0 \leq k < n$.*

Proof. See [17, Theorem 4.3] for this result in characteristic zero. See [3, Remark 4] for non-zero characteristic. □

We will describe a set of representations of G as *jointly faithful* when the intersection of the kernels of the representations is trivial. In practice we do not consider higher tensor powers of a module, but instead consider irreducible modules which are jointly faithful, and tensor products of these in pairs (not neglecting to take squares). These tensor product modules are split into irreducibles and the process continued with any newly found irreducible modules. It is clear from considering the Brauer characters and the BSB Theorem that this will result in finding all irreducible modules.

Nakayama's formulas [25, Chapter 8] show that every irreducible G-module occurs as a constituent of a module induced from a proper subgroup. Indeed this remains true if we restrict ourselves to the special case of transitive permutation modules of G. We have now three methods for finding all absolutely irreducible G-modules given an initial faithful KG-module. For reasons of computational efficiency we do not restrict ourselves to a particular choice from these methods in what follows, but employ all three.

3.2. Brauer characters

A general reference for the theory of Brauer characters is Navarro [25]. In addition, Chapter 15 of Isaacs [17] provides a useful introduction. The definition of the Brauer characters of a group depends upon the choice of a maximal ideal in a ring of algebraic integers. This corresponds to the choice of irreducible polynomial used to define the finite field $\mathrm{GF}(p^n)$. In order to make a unique choice the Conway polynomial [21, Chapter 4]

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

is normally used to define the finite field. Exactly one Conway polynomial is defined for any finite field. Apart from the uniqueness property for $\mathrm{GF}(p^n)$, it is compatible with the Conway polynomials defining each of its subfields. So when lifting elements of $\mathrm{GF}(p^n)$ to a complex value, a root α of the Conway polynomial of degree n over $\mathrm{GF}(p)$ lifts to $\zeta = \exp(2\pi/(p^n - 1))$, and, noting that α is a primitive element of the field, α^m lifts to ζ^m. The definition and proof of existence of Conway polynomials for any degree may be found in [21, Chapter 4].

Magma includes a library of Conway polynomials. However, as these polynomials are extremely expensive to compute they are only available for moderate sized fields. In the implementation of the algorithms described here it is assumed that for any finite field K arising a Conway polynomial is known.

A key step of the main algorithm is based on the observation that if we can construct a G-module S_i, whose Brauer character does not lie in the span of the known Brauer characters of G, then that G-module contains an absolutely irreducible constituent that is new. As we can often deduce the Brauer character of a prospective module S_i without computing it directly from the module S_i, the above observation enables us to identify a module S_i that contains at least one irreducible that is not a constituent of any of the modules $S_1, \ldots, S_{i-1}$. Examples of such constructions are tensor products of known irreducibles, permutation modules and induced modules of subgroups of G. Thus, only modules S_i that are guaranteed to yield a new irreducible have to be split.

When a new irreducible KG-module M of G is found its Brauer character has to be determined. If M is a tensor product or an induced representation, where the Brauer characters of the base KG-modules are known, this is straightforward. However, it is often necessary to compute the Brauer character directly from M and the following procedure is used to reduce the number of p-regular classes on which the representation has to be computed. At the outset the power map of the group is used to identify a subset of the conjugacy classes (*special classes*) which has the property that every class is a power of one of the special classes. To compute the Brauer character, the matrix corresponding to a class representative of a special class is found, and its eigenvalues computed. These are lifted to the complex numbers as described above. For the remaining p-regular classes, appropriate powers of the eigenvalues are lifted.

A step in the algorithm involves finding a characteristic zero representation of G, with known character χ, which is reduced modulo p to obtain a representation with Brauer character χ^0. While the p-modular reduction of an ordinary representation is not unique, character theory guarantees that the modular absolutely irreducible constituents of the reduced representation depend only on χ. Every irreducible Brauer character is a constituent of some χ^0 where $\chi \in \mathrm{Irr}(G)$, so this process provides one way to find all absolutely irreducible G-modules.

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

4. Condensation

4.1. Introduction

Let $\rho : G \to \mathrm{GL}_d(F)$ be a representation of the finite group G and let M be the corresponding A-module, where A is the group algebra FG, and F is a field of any characteristic. When ρ can be described by a compact construction (such as a permutation representation) instead of explicit $d \times d$ image matrices for the generators alone, *condensation* can yield a large speedup in the splitting of M, while also avoiding the explicit construction of M and the direct application of the Meataxe to M. The original examples go back to Parker and Thackray in 1979 [39]; see also [29, Sec. 2], [38] for more basic theory. Here we use *fixed-point* condensation, which is as follows. Let U be a fixed subgroup of G (called the *condensation subgroup*), whose order is coprime to the characteristic of F. Define

$$e := \frac{1}{|U|} \sum_{u \in U} u \in A.$$

Then it is easy to see that e is an idempotent, and after setting $\tilde{A} = eAe$ and $\tilde{M} = Me$, it is elementary to show that $\tilde{M}$ is an $\tilde{A}$-module and if S is a submodule of M, then Se is a submodule of $\tilde{M}$. We call $\tilde{A}$ the *condensed algebra* of A and $\tilde{M}$ the *condensed module* of M.

We develop techniques so that condensation can be used automatically for constructing irreducible FG-modules. To compute one or more submodules of M without constructing M itself explicitly, one well-known basic approach is as follows:

1. Choose a condensation subgroup U of G whose order is coprime to the characteristic of F and let e be the idempotent $\frac{1}{|U|} \sum_{u \in U} u \in A$, where $A = FG$.
2. Let $\{x_1, \ldots, x_r\}$ be random elements of G and let $\tilde{A}_0$ be the subalgebra of $\tilde{A} = eAe$ generated by $\{ex_1e, \ldots, ex_re\}$ and let $\tilde{M}_0$ be the corresponding $\tilde{A}_0$-module. Set up a reduced basis for $\tilde{M}_0$ and the corresponding reduced action on $\tilde{M}_0$ by $\tilde{A}_0$.
3. Compute a proper submodule $\tilde{S}_0$ of $\tilde{M}_0$ (by any method).
4. Embed $\tilde{S}_0$ in M by the natural vector space embedding of $\tilde{M}$ in M (this is usually called '*uncondensing*'). Then compute the submodule S of M which is spanned by this embedding of $\tilde{S}_0$ into M. This is usually done by the *spin algorithm*, which computes a basis of the invariant subspace generated by a set of vectors under the action by A.

In Step 2, it is not easy in general to determine cheaply whether $\tilde{A}_0$ equals $\tilde{A}$, so that is why $\tilde{A}_0$ has to be used in practice. If there is inequality, then $\tilde{M}_0$ may split more than $\tilde{M}$, but this issue will be addressed below. Steps 3 and 4 may be repeated for other submodules of $\tilde{M}_0$. This approach has been used in many situations where theoretical

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

considerations in the particular setting yield (1) a good choice of the subgroup U and (2) information on how to identify a suitable submodule $\tilde{S}_0$ of $\tilde{M}_0$. But in general, an automatic choice of U is difficult: if U is too large, then submodules of M may map to zero in $\tilde{M}_0$ so there may be no useful result, while the smaller U becomes, the larger $\tilde{M}_0$ becomes and the closer the whole method approaches to splitting M directly by the Meataxe, which is what we want to avoid. Also, in an automatic setting, one cannot easily identify a suitable submodule $\tilde{S}_0$ in general if only one specific submodule of M with certain properties is desired.

4.2. Modular condensation

We first describe an automatic modular condensation algorithm to construct irreducible KG-modules for a finite field K. We use three types of condensation to work with a given virtual representation ρ, while avoiding the explicit construction of the matrices describing the action of the KG-module M corresponding to ρ; in each case we also note how the dimension of a condensed module $\tilde{M}$ can be precomputed from ρ and a potential condensation subgroup U which determines $\tilde{M}$, without explicit use of characters (see the references for more details):

1. *Permutation condensation* [23, 3.4]: Here ρ is a permutation representation of G and the construction of a condensed module involves counting orbit intersections under the action given by ρ. The dimension of $\tilde{M}$ for a potential U can simply be computed as the number of orbits of $\rho(U)$ as a permutation group.
2. *Induction condensation* [24]: Here a representation ρ_H of a subgroup $H < G$ is given, and ρ is the induced representation $\rho_H \uparrow^G$. Constructing a condensed module needs a special transversal of G over H which involves H-U-double cosets of G. The dimension of $\tilde{M}$ for a potential U can be computed as the rank of a matrix obtained by summing images under ρ_H of the union of a list of subgroups of H easily computed from the double cosets.
3. *Tensor condensation* [22]: Here representations ρ_1 and ρ_2 of G are given and ρ is the tensor representation $\rho_1 \otimes \rho_2$. The construction of a condensed module is quite elaborate and involves matching the decompositions of the restrictions of ρ_1, ρ_2 to U. The dimension of $\tilde{M}$ for a potential U can be computed from the dimensions of the endomorphism rings of the components in these decompositions.

The modular algorithm, given the virtual representation $\rho : G \to \mathrm{GL}_d(K)$ with associated methods for condensation, where $K = \mathrm{GF}(p)$, returns a set of irreducible KG-modules, and proceeds as follows:

1. Let d be the degree of ρ. Set B to a target dimension for the condensed module (typically around $d/10$; see discussion below). Loop over a suitable set of small subgroups $\{U_1, \ldots, U_l\}$ of G (each with order coprime to p) and let U be the U_i such

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

that the dimension d_i of the condensed module determined by U_i is closest to B (using the relevant method mentioned above to compute d_i). Let e be the idempotent corresponding to U.

2. Let r be some small positive integer (typically 20), and set $\tilde{A}_0$ to the algebra generated by $ex_1e, \ldots, ex_re$ for random elements $\{x_1, \ldots, x_r\}$ of G. Let $\tilde{M}_0$ be the corresponding $\tilde{A}_0$-module, and let $0 = \tilde{S}_0 < \tilde{S}_1 < \ldots < \tilde{S}_k = \tilde{M}_0$ be a composition series of $\tilde{M}_0$ (computed by recursive use of the modular Meataxe).
3. Set $V_0 = 0$ and for $i = 1, \ldots, k-1$, uncondense and spin a set of generators of $\tilde{S}_i$ modulo V_{i-1} to obtain submodule V_i of M and let $Q_i = V_i/V_{i-1}$. Let $Q_k = M/V_{k-1}$.
4. Compute the irreducible constituents of $Q_1 \ldots, Q_k$ (by recursive use of the modular Meataxe) and return their union to obtain the irreducible constituents of M.

Since $0 \leq V_1 \leq \ldots V_{k-1} \leq M$ form an ascending sequence of submodules of M by construction, the Q_i contain all constituents of M and so the output is clearly correct.

In Step 1, B is heuristically chosen so that the dimension of $\tilde{M}_0$ will be sufficiently smaller than M so that the splitting of $\tilde{M}_0$ will be much faster than the splitting of M itself would be, while trying to avoid the fact that the smaller the dimension of $\tilde{M}_0$ is, the more likely it is that the Q_i will be reducible, so Step 4 will be more expensive. The value of B should also depend on the type of representation being condensed, since this affects the cost of setting up the condensed module relative to the cost of the composition series computation in Step 2. In the Magma implementation, B is set to $\lceil d/10 \rceil$ in the permutation and induction case, and $\lceil d/8 \rceil$ in the tensor case. One could also use the methods of Noeske [26] to ensure that $\tilde{A}_0 = \tilde{A}$, but the random choice of generators works well enough in practice to yield a useful result. In the Magma implementation, the candidate U_i condensation subgroups are taken to be the cyclic subgroups generated by the p-regular class representatives of the group and the Sylow subgroups and all their subgroups (up to conjugacy) for all primes different to p. In practice, this is sufficient to find a suitable U.

In the Magma implementation, the modular Meataxe algorithm used in Steps 2 and 4 essentially uses the method of Holt/Rees [16], but with some optimisations. First, asymptotically-fast matrix multiplication is used in all characteristics (including Strassen multiplication [36] in all characteristics and blocking methods for $p \leq 7$, as in [1]), and matrix echelonisation and the spinning algorithm also map to this fast matrix multiplication. Next, since the computation of the characteristic polynomial of a random algebra element is typically much more expensive than any other operation, in the case that $p \leq 7$ Magma typically generates more random elements of the algebra and searches for elements of small nullity (via random shifts by random scalars) before computing the characteristic polynomial (unlike Holt/Rees, which immediately computes the characteristic polynomial for each new random element). Also, when the dimension is greater than 10000, Magma uses the asymptotically-fast algorithm of Keller-Gehrig [19] to compute a characteristic polynomial, which is particularly effective when $p > 7$.

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

One main limitation of this algorithm is that it eventually has to construct a basis of rank r, where r is the dimension of V_{k-1}, and this may be close to the degree of ρ. Also, for the induction case, the representation of H must first be constructed and it may be difficult to know beforehand which induced representations will yield useful representations of G. The rational reduction algorithm in the next subsection is usually better for the permutation and induction cases, but the modular algorithm remains very useful for the splitting of tensor products, because an irreducible module S can often be extracted from a tensor product of much smaller dimension than any permutation or induced representation which has S as a constituent.

4.3. Modular reduction of rational representations

For a finite group G, let $\mathrm{Irr}_{\mathbb{Q}}(G)$ denote the set of characters of all irreducible $\mathbb{Q}$-representations of G. These can be constructed by summing the Galois orbits of the complex irreducible characters of G, and multiplying by the Schur index common to the orbit in each case [17, Theorem 9.21].

An algorithm was developed by A. Steel [35, Sec. 3.8] which takes an irreducible rational character χ for G and constructs an ordinary G-module (written over $\mathbb{Q}$); the algorithm is completely automatic and uses condensation techniques in characteristic zero. More recently he has developed an extension which, given an irreducible rational character χ for G and a prime p, constructs a submodule of a modulo-p reduction of a $\mathbb{Q}G$-module M which affords χ. This is done without explicitly constructing M, and is described by the algorithm below.

Note first the following well-known lemma: given a potential condensation subgroup U it can be used to predict the trace of an element of $\tilde{A}$ or the dimension of a submodule of $\tilde{M}$, where $\tilde{M}$ is the condensed $\tilde{A}$-module corresponding to U (while avoiding the explicit construction of $\tilde{A}$ or $\tilde{M}$).

Lemma 4.1. *Let M, $U \leq G$ and the idempotent e corresponding to U be as in Subsection 4.1. Then:*

1. *The* Trace Equation *(first stated in [37, p. 279]) gives the trace of the matrix representing the action of ege on Me, as follows:*

$$\mathrm{Tr}_{Me}(ege) = \mathrm{Tr}_M(ege) = \mathrm{Tr}_M(gee) = \mathrm{Tr}_M(ge) = \frac{1}{|U|}\sum_{u\in U}\mathrm{Tr}_M(gu) = \frac{1}{|U|}\sum_{u\in U}\chi_M(gu),$$

   where χ_M is the character of M.
2. *Setting $g = 1_G$ in the above formula, the dimension of the condensed submodule $\tilde{S} = Se$ for a submodule S of M and a potential U can be computed as:*

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

$$\frac{1}{|U|}\sum_{u\in U}\chi_S(u) = \langle \chi_S \downarrow U, 1_U\rangle,$$

where χ_S is the character of S and 1_U is the trivial character of U.

The following notation (based on the Lemma) will be used:

1. Write $\texttt{Trace}(\chi, U, g) = \frac{1}{|U|}\sum_{k\in U}\chi(gk)$ for character χ, $U \leq G$ and $g \in G$ (using the Trace Equation).
2. Write $\texttt{CondDim}(\chi, U) = \langle \chi \downarrow U, 1_U\rangle$ for character χ and for $U \leq G$, where 1_U is the trivial character for U (giving the condensed dimension of χ w.r.t. U).

The rational reduction algorithm, given an irreducible rational character χ from $\mathrm{Irr}_{\mathbb{Q}}(G)$ and a prime p, returns a non-zero submodule S_p of a mod-p reduction M_p of a $\mathbb{Q}G$-module M which affords χ, and proceeds as follows:

1. Perform a search for a minimal-degree virtual rational representation ρ of G whose character χ_ρ includes χ as a $\mathbb{Q}$-irreducible constituent. The search loops over: (1) characters of permutation representations of G of increasing degree; (2) induced characters, which are found by constructing subgroups of G of increasing index, computing their character tables and inducing all their characters to G. The search maintains a priority queue (sorted by character degree) and tests each potential χ_ρ in order until one containing χ is found (extending the queue by permutation/induced characters of increasing degree each time the queue is exhausted).
2. If induction from a subgroup H is to be used, construct the relevant rational representation of H by any method, conjugate this to an integral representation ρ_H, and let ρ denote $\rho_H \uparrow^G$; otherwise let ρ denote the relevant permutation representation affording ρ_χ. Set up the relevant machinery for condensation of ρ at any given subgroup U using the methods referenced in Subsec. 4.2 and let M (not explicitly constructed) denote the $\mathbb{Q}G$-module corresponding to ρ.
3. Let

$$\chi_\rho = \sum_{i=1}^{k} m_i \cdot \chi_i, \quad \chi_i \in \mathrm{Irr}_{\mathbb{Q}}(G)$$

(with each $m_i > 0$) be the decomposition of χ_ρ into irreducible rational characters and let I be the index such that $\chi = \chi_I$.
4. Choose a positive integer r for the number of algebra generators (typically 20 initially).
5. If $r > |G|$, then set $r = |G|$. Let $R = \{x_1 = 1, x_2, \ldots, x_r\}$ be r distinct elements of G (randomly chosen except for x_1).
6. *Find condensation subgroup U of G suitable for χ_ρ, χ:*

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

Set L to a suitable list of small subgroups $(U_1, \ldots, U_l)$ of G which at least contains the trivial subgroup. Set $D_{\text{best}} := \infty, U_{\text{best}} := \langle 1 \rangle$.
For $i = 1, \ldots, l$, do the following:

(i) Let $U = U_i$. Set $D :=$ `CondDim`(χ_ρ, U). If $D \geq D_{\text{best}}$ or if `CondDim`$(\chi_j, U) = 0$ for any j $(1 \leq j \leq k)$ then skip to the next i.

(ii) For $1 \leq j \leq k$, set $T_j :=$ (`Trace`$(\chi_j, U, x_1), \ldots,$ `Trace`$(\chi_j, U, x_r)) \in \mathbb{Z}^r$. Using a recursive search algorithm, determine whether there is a unique solution for the linear equation $\sum_{j=1}^{k} c_j T_j = m_I T_I$, where the $c_1, \ldots, c_k$ variables are all non-negative. If there is a unique solution (which must have $c_I = m_I$ and $c_j = 0$ for $j \neq I$), then set $D_{\text{best}} := D$, $U_{\text{best}} := U$.

7. Construct the partial condensed algebra $\tilde{A}_0 = \langle ex_1e, \ldots, ex_re \rangle$ and $\tilde{A}_0$-module $\tilde{M}_0$ derived from $\chi_\rho, R, U_{\text{best}}$ and corresponding idempotent e.
8. Use a rational Meataxe to obtain a direct sum decomposition of $\tilde{M}_0$ into $\mathbb{Q}$-irreducible non-isomorphic submodules:

$$\tilde{M}_0 = \oplus_{i=1}^{l} \oplus_{j=1}^{n_i} \tilde{S}_i.$$

9. If $\sum_{i=1}^{l} n_i > \sum_{i=1}^{k} m_i$, then $\tilde{M}_0$ splits more than the correct condensed $\tilde{A}$-module $\tilde{M}$ (for the full condensed algebra $\tilde{A}$) does, so $\tilde{A}_0$ must be invalid (i.e., we must have $\tilde{A}_0 \subsetneq \tilde{A} = eGe$, where e is the idempotent for U), so increase r by a small positive integer (typically 10), and return to Step 5.
10. Let i be such that the r traces of the generators of the action on $\tilde{S}_i$ equals the trace vector T_I for U (as defined in Step 6 (ii)).
11. Uncondense a set of generators of $\tilde{S}_i$ back to the underlying vector space of M, clear denominators and remove the content (GCD of all entries) for each vector to obtain a set $V_{\mathbb{Z}} \subset \mathbb{Z}^d$ of integral vectors (where d is the degree of ρ), and spin these modulo p under the mod-p reduction of the action given by ρ to obtain a non-zero submodule S_p of the mod-p reduction of M and return S_p.

Theorem 2. *The above algorithm is correct.*

Proof. In Step 2, the representation of H can always be conjugated to an integral representation in finite time in the induction case since G is finite. Step 6 finds the best condensation subgroup U of L (yielding the smallest dimension for the corresponding condensation module) which satisfies the following conditions: (1) all constituents of M do not map to zero in the condensed module $\tilde{M}$ derived from ρ and U; (2) the simple constituent $\tilde{S}$ of $\tilde{M}$ corresponding to χ can be uniquely identifiable via the T_j trace vectors. In Step 6 (ii), the first coordinate d_j of each T_j equals `CondDim`(χ_j, U) (since $x_1 = 1$) and is thus strictly positive, so whether the linear equation has a unique solution or not can be determined in finite time by a simple recursive search with a bound on each coordinate c_j given by $1 \leq c_j \leq m_I \lfloor \frac{d_j}{d_I} \rfloor$. Some U satisfying the conditions will always

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

be found, since the conditions are satisfied by the trivial subgroup, which is included in L.

Since $\tilde{A}_0$ is a subalgebra of $\tilde{A}$ (the full condensed algebra eAe), the splitting of $\tilde{M}_0$ corresponds to a possible refinement of the splitting of the correct condensed module $\tilde{M} = eMe$, which must have exactly $\sum_{i=1}^{k} m_i$ $\mathbb{Q}$-irreducible components (counting multiplicities). If the component counts match in Step 9, then there is no proper refinement, so the submodules of $\tilde{M}_0$ match those of $\tilde{M}$ exactly. Then the unique condition on the trace vectors ensures that the $\tilde{S}_i$ corresponding to χ is identified correctly, so the uncondensing will be correct (matching the uncondensing of the relevant component of the correct condensed module $\tilde{M}$) and so the module S_p returned in the final step must be a submodule of the mod-p reduction of a submodule S of M with character χ (since ρ is integral and the vectors in $V_{\mathbb{Z}}$ are integral and non-zero modulo p after the content removal, the spin can be done modulo p without any denominators occurring). In the case that $\tilde{M}_0$ is invalid in Step 9, the algorithm repeats with more generators in R and thus for $\tilde{A}_0$. In the worst case, R eventually equals G, in which case $\tilde{A}_0 = \tilde{A}$ and so termination is assured. □

In the Magma implementation, the rational representation ρ_H in the induction case in Step 2 is constructed by the algorithm described in [35, Sec. 3.8], while the Rational Meataxe algorithm used in Step 8 is described in [35, Sec. 2.4]. In Step 6, the algorithm for determining the solutions to the non-negative linear equation is clearly related to the Knapsack problem. A heuristic algorithm which greatly speeds up a simple recursive search is described in [35, Section 3.2]; this uses all coordinates of the trace vectors to add extra conditions in the solution coordinates, and drastically reduces the search space so that the algorithm takes very little time in practice. The U_i subgroups are chosen in the same way as in the modular condensation algorithm above, except that subgroups with order divisible by p are included (since the characteristic is zero in the application of condensation here). Finally, in Step 11 the Magma implementation also effectively first reduces $\tilde{S}_i$ mod p, computes a composition series of this and then uncondenses and spins generators of the quotient sections as in the above modular condensation algorithm. This often yields a non-trivial pre-splitting of S_p, thus avoiding the need to apply the modular Meataxe directly to S_p. Note also that the full integral rank (the dimension of S) may not be reached in the final spin modulo p, but in practice we find that the deficit is small in general (any missing submodules modulo p tend to be small and are easily constructed by other means in the main algorithm in the next section).

The key feature of this algorithm is that it exploits the information given by the ordinary characters so as to achieve the following goals (which cannot be achieved with the above modular condensation algorithm):

1. Automatically find suitable permutation/induced representations from which we can extract desired representations: this is done purely with ordinary characters and no construction of any modules; in particular, the search phase identifies which induced

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

representations will yield the desired representation affording χ, without having to create any representations of subgroups explicitly.

2. Determine a condensation subgroup so that the dimension of the corresponding condensed $\tilde{A}$-module $\tilde{M}$ is as small as possible and correctly constructed (i.e., with enough generators for $\tilde{A}$), while ensuring that the desired condensed component $\tilde{S}$ does not map to zero.
3. Determine only a single submodule S of M corresponding to χ (by using the decomposition of χ_ρ into irreducible rational characters), thus avoiding the construction of mod-p reductions of a full composition series of M; this typically saves a huge amount of time and memory and allows the modules to be extracted from much higher-degree permutation/induced representations than would be otherwise feasible (see Table 4 below for large examples).

On the other hand, the bulk of the time in this algorithm is typically taken in the spin algorithm in the last step, so it benefits from doing the large matrix operations with matrices in a small finite field, thus avoiding very expensive computations with large matrices in characteristic zero, while also saving memory in the case that p is very small because of the efficient packing of the matrix entries. If induction is used, even though the representation ρ_H of the subgroup H has to be constructed in characteristic zero, this generally does not take much time compared to the final modular spin (and monomial representations arise commonly, which are easy to construct).

5. Construction of absolutely irreducible modules

Throughout this section K will denote a finite field of characteristic p. Our goal is to construct the absolutely irreducible KG-modules for G. It is assumed that the group G is given as a set of generators consisting of permutations or matrices defined over a finite field having the same characteristic as the field K. The absolutely irreducible KG-modules are found by constructing a sequence

$$\{S_1, \ldots, S_n\}$$

of G-modules, such that each S_i $(i = 2, \ldots, n)$ has one or more irreducible constituents that are not constituents of $S_1, \ldots, S_{i-1}$. For practical reasons, the dimensions of the S_i should be chosen to be as small as possible.

5.1. Construction of the modules $S_1, \ldots, S_n$

The methods used fall into two groups: those that generate the modules starting from a faithful KG-module and those that are obtained as a p-reduction of an irreducible $\mathbb{Q}G$-module. The first group uses tensor products and induction to construct the modules S_i and will be referred to collectively as TI-modules. The various TI-constructions used are:

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

- *Tensor product*: Given KG-modules M and N, a new KG-module can be obtained by taking their tensor product. New irreducibles will commonly be found in the product.
- *Exterior and symmetric powers*: Of particular interest are the exterior square E and symmetric square S of a d-dimensional KG-module M. These are quotients of the tensor product $P = M \otimes M$ having dimensions $d(d-1)/2$, and $d(d+1)/2$, respectively. It is often possible to split E or S when it is impossible or too expensive to split P. While higher symmetric powers may also be used we currently restrict to squares.
- *Permutation modules*: Given a subgroup H of moderate index in G it is straightforward to construct the permutation action of G on the cosets of H. Once the permutation representation ρ of G is constructed, the modular condensation algorithm can be applied to ρ and p.
- *Induction from a subgroup*: Let H be a subgroup of G. A KH-module L of degree d can be induced to G giving a KG-module of dimension de, where e is the index of H in G. Consequently, this induction is only practical for subgroups H of modest index in G and KH-modules of modest dimension.

In general, modules M generated by the above constructions will be split into irreducibles using the modular condensation algorithm (Subsec. 4.2). For each of the above applications of condensation the procedure used is guaranteed to return an irreducible from each isomorphism class of irreducible modules appearing as constituents in M. This observation ensures that the conditions of the Burnside-Steinberg-Brauer theorem (or variants) are satisfied and so the algorithm is guaranteed to construct all irreducibles.

The second source of modules comes from p-modular reductions of irreducible $\mathbb{Q}G$-modules using the algorithm in Subsec. 4.3. These will be referred to as rational-reduction modules (RR-modules).

5.2. Algorithm 1: absolutely irreducible modules for a group

The number of absolutely irreducible G-modules of characteristic p will be denoted by l. The algorithm returns a sequence *IrrM* containing the l absolutely irreducible KG-modules and a sequence *IrrB* containing the corresponding list of Brauer characters.

1. The conjugacy classes of G are found using the algorithms described in Cannon and Holt [8]. Given the classes, it is a trivial matter to write down the p-regular classes of G and hence determine the value of l for G.
2. The character table X of G is constructed using Unger's algorithm [40].
3. The set $\mathcal{U}$ of representatives of conjugacy classes of subgroups of G down to some index limit B are computed using the low index subgroups algorithm [9]. The subgroups will be used for the construction of permutation modules and, more generally, modules induced from subgroups as choices for the modules S_i. Permutation representations are highly desirable candidates since splitting them using condensation

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

has the least cost of any method and is generally practical up to degree 1,000,000 or more.

4. The linear characters are computed and stored in the list *IrrM* of modular irreducibles.
5. If G is given by a permutation representation take S_1 to be the corresponding permutation module over $GF(p)$. If G is given as a matrix group in characteristic p then take S_1 to be the natural G-module. In both of the above cases proceed to Step 7.
6. The next module $M = S_i$ is chosen as follows.
   - *TI-modules:* The character ϕ of a tensor product or induced KG-module M is easily deduced given that the characters of the known irreducibles are available. Obtaining the character of a permutation module is straightforward. By testing whether ϕ lies in the space spanned by the known irreducible Brauer characters, we can decide if a given candidate for M will yield new irreducibles, at very little cost. Those TI-module candidates that contain new irreducibles are noted in a list L. The permutation and induction representations are derived from the subgroups in $\mathcal{U}$; if every such subgroup has already been tried, then the index limit B is suitably increased and further subgroups of G up to the new bound B are constructed and inserted into $\mathcal{U}$ and so on until the list L is non-empty.
   - *RR-modules:* An RR-module derived from an irreducible rational character χ can yield new irreducibles if the Brauer character χ^0 does not lie in the Brauer character space spanned by the known irreducibles. Such modules identified as possibly yielding new KG-modules are added to the list L.

   Finally, M is selected from L on the basis that it can be split for the least cost. Note that no module is constructed during this step so that it takes very little time.
7. If M is a TI-module, then the modular condensation algorithm (Subsec. 4.2) is used to construct the set T of irreducible constituents of M; otherwise the rational reduction algorithm (Subsec. 4.3) combined with the modular Meataxe applied to the output is used to construct the set T of irreducible constituents of M. Any new module N that is not absolutely irreducible is rewritten over a splitting field and then split into conjugates which replace N in the list T (a minimal-degree splitting field E and an absolutely irreducible constituent of M written over E are constructed by the algorithm described in [16, Sec. 3]). Then T is reduced to a list of non-isomorphic modules which are also not isomorphic to any previously constructed irreducible module (isomorphism testing for two irreducible modules is performed using the algorithm described in [16, Sec. 4]).
8. The characters of the new irreducible KG-modules are determined as follows:
   - *TI-modules:* The character of each module M was calculated when choosing M, and we also know the Brauer character for each known constituent of M. If there is a single new constituent, then its character can be obtained by simple character arithmetic. If there are r new constituents of M then it will be necessary to compute the characters of $r-1$ of the new irreducible modules by direct calculation of the characters from the modules.

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

- *RR-modules:* This is similar to the TI case, except that if the rank of the submodule S_p returned by the rational reduction algorithm is less than the degree of the rational character χ, then the Brauer character of the last constituent cannot be inferred by simple character arithmetic. However, in the case that χ is absolutely irreducible, has Schur index one and lies in a p-block having zero deficiency, then the Brauer character is χ^0.

9. The set T is replaced by its closure under the operations of duality and Galois conjugacy. The lists *IrrM* and *IrrB* are updated with the new modules and characters. If l absolutely irreducible modules have been found the algorithm returns *IrrM* and *IrrB* and terminates. Otherwise, we return to Step 6.

We consider the termination and correctness of the algorithm. The algorithm terminates when l distinct absolutely irreducible modules have been found, which is well known to be the full set of such modules. To show correctness we must show that all the absolutely irreducible modules will be found.

Since $O_p(G)$ lies in the kernel of every irreducible representation of G, in what follows 'faithful representation' will refer to a faithful representation of $G/O_p(G)$. Once a set of jointly faithful irreducible representations of G is computed, Theorem 1, and the discussion following its statement, guarantees that every absolutely irreducible G-module occurs as a constituent of some tensor product which will be considered as a TI-module. The methods used to split these tensor product modules find all constituents, so we need only show that such a jointly faithful set will be found.

In this paper it is assumed that G is given either by a permutation representation or as a matrix group over a finite field of characteristic p, so that the constituents of S_1 form such a jointly faithful set.

The selection of the modules S_i is quite difficult. The cost of splitting them depends to a significant degree upon how the module is constructed. For example, the time taken to split a permutation module of dimension 100,000 is much less than the cost of splitting a tensor product of that dimension. Often there are several choices for S_i and it is difficult to obtain good estimates of the cost of splitting a particular choice.

RR-modules alone can be used to obtain most of the modular irreducibles. However, in some cases the reduction stage of this approach can be very time-consuming or even impossible while one of the TI type constructions is faster. The TI constructions are often faster for finding irreducibles of smaller dimension while increasing the chances of success with large examples. On the other hand the RR construction can often find irreducible modules whose dimension is beyond the capability of TI constructions.

The algorithm finds the irreducibles roughly in order of increasing degree and it can be terminated at the point where irreducibles of a specified degree d have been found. While there is no guarantee that all irreducibles of degree less than d have been found, this feature frequently makes it possible to find an irreducible of specified degree in a situation where it is not possible to construct all the irreducibles of that characteristic.

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

5.3. Algorithm 2: absolutely irreducible modules for a block

In many situations knowledge of only some of the irreducible modules is required. If, say, a single irreducible module N is required where either its Brauer character or the corresponding complex irreducible character is known then it can be found at the cost of splitting a single module. Unfortunately this knowledge is rarely available. However, it is possible to compute all the irreducible modules belonging to a specified p-block by considering at most l modules $S_1, \ldots, S_l$, where l is the number of nonequivalent irreducible modules in the block.

The block algorithm is a simple variation of Algorithm 1. Let B be a p-block for G. Using characters we can easily calculate the number of KG-modules associated with B. As RR-modules are defined by the irreducible complex characters of G we know immediately from the p-blocks which RR-modules belong to B. Given an arbitrary Brauer character we can compute its projections onto the blocks and thereby determine if any of the constituents of its module belong to B. These two observations replace the test for a candidate module M having new constituents in B in Step 6 of Algorithm 1. The module associated with a defect zero block can be obtained by applying the RR method to the appropriate rational character. More generally, there are sometimes advantages in producing the irreducibles block-by-block. Knowing the complex characters that belong to a given block may allow us to tighten our upper bound on the possible dimensions of its KG-modules.

6. Performance

The algorithm presented here is a general-purpose algorithm designed to work for any type of finite group. While there are undoubtedly better methods for particular types of group, such as soluble groups, the goal was to design an algorithm which would work for any type of finite group. From our experience there is demand to calculate the Brauer characters and/or irreducible KG-modules for a wide range of group types.

The availability of both the rational reduction (RR) construction together with the tensor/induction (TI) constructions results in an algorithm capable of succeeding with a wide range of groups. When constructing irreducibles, the TI-approach is often faster for modules M of dimension up to a few thousand while the RR-approach comes into its own for higher degrees. Failure to find all irreducibles by either method is usually due to not being able to find a module S_i containing new irreducibles that is small enough to split.

The operations that are most expensive are the modular spin algorithm in both of the condensation algorithms, splitting a module, testing two modules for equivalence, testing whether a module is absolutely irreducible and computing the Brauer character of a module directly. The complexity of all of these is at least cubic in the dimension of the module. The size of the characteristic p plays a significant role as multiplication of matrices over fields of cardinality less than 8 will be significantly faster than say over

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

Table 1
Timings for calculating the Brauer characters and the irreducible G-modules for a selection of non-simple groups.

		2	3	5	7	11
	l	9	89			
$S_4 \wr S_4$	d	64	486			
	t	1	9			
T42–2100	l	28	20		162	
$3^7.2^6.7.2$	d	448	14		448	
	t	13	6.6		252	
T42–2500	l	81	20		477	
$3^7.2^6.7.3.2$	d	448	14		448	
	t	34	14		10470	
T42–2000	l	38	20		132	
$3^7.2^2.L_2(7)$	d	336	7		336	
	t	3.1	1.8		75	
Max Co_2	l	8	79	104	103	
$(2^4 \times 2^{1+6}).A_8$	d	64	2520	2520	2520	
	t	1	219	392	333	
Max Co_2	l	6	63	74	71	78
$2^{10} : M_{22} : 2$	d	140	13860	13860	13860	13860
	t	1	3453	25286	11359	105690
Max Co_2	l	8	59	94	95	
$2^{1+8}.S_6(2)$	d	512	6480	7680	7560	
	t	5	629	3435	2225	
T42–4612	l	74	72		378	
$3^2.(2 \times A_7^2).2^2$	d	1600	600		4900	
	t	107	101		18862	

GF(17). Other factors include the existence of many classes of Galois conjugates and the degree of extensions of GF(p) that are needed to realise the absolutely irreducible modules. In particular, splitting a module over large degree extension significantly degrades the performance of the Meataxe.

All examples in Tables 1, 2 and 3 were run on an Intel Core i7-7700 CPU @ 3.60GHz with 64GB of memory.

Table 1 presents timings for a selection of non-simple groups. The first three groups are soluble, the next four are non-soluble with large soluble radicals, and the final group has two copies of A_7 with small soluble groups above and below. The first group is the wreath product of S_4 with itself, a permutation group of degree 16. Groups labelled T42 are transitive groups of degree 42, where the number in the list of transitive groups is given after the hyphen, and chief factors are noted. The remaining are maximal subgroups of Co_2, with the structures noted. The line labelled 'l' is the number of p-regular classes of the group, the line labelled 'd' gives the dimension of the largest absolutely irreducible module found, while the line labelled 't' gives the time in seconds for the computation. Columns 2 to 6 give information about the computation of the KG-modules for characteristics $p = 2, 3, 5, 7$, respectively.

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

Table 2
Timings for calculating the Brauer characters and the irreducible G-modules for each of the maximal subgroups of Co_3.

		2	3	7		2	3	7
	l	9	10	28		10	14	26
$A_4 \times S_5$	d	4	18	18	$U_3(5) : S_3$	288	288	252
	t	0.1	0.1	0.2		3.2	3.4	9.4
	l	16	6	30		8	21	41
$S_3 \times L_2(8) : 3$	d	24	27	42	$2.S_6(2)$	512	405	560
	t	0.2	0.1	0.3		1.5	3.5	56
	l	5	26	45		11	16	37
$[2^{10}3^3]$	d	6	36	72	$3^5 : (2 \times \mathrm{M}_{11})$	880	45	880
	t	1.4	1.7	2.7		1.6	0.5	29
	l	6	27	30		11	13	13
$2 \times \mathrm{M}_{12}$	d	144	99	176	M_{23}	896	1035	1034
	t	0.3	0.54	1.4		16	17	96
	l	8	16	25		7	26	42
$L_3(4) : D_{12}$	d	128	126	252	$U_4(3).(2^2)_{133}$	1280	729	1120
	t	1.3	1.4	2.0		5.1	5.1	96
	l	8	17	21		9	19	23
$2^4.A_8$	d	64	315	315	HS	1408	2520	2520
	t	0.3	1.4	2.9		23	61	307
	l	12	18	54		10	20	31
$3^{1+4} : 4.S_6$	d	320	12	360	McL : 2	19712	16038	20790
	t	2.1	1.4	20		589	547	4361

In order to present an unbiased sample of diverse groups, Table 2 gives some information about the computation of modular representations and Brauer characters for each of the 14 maximal subgroups of Co_3 in characteristics $p = 2, 3, 7$. (The computations have been done for all characteristics but the information for those greater than 7 have been omitted to save space.) The groups given as input to the algorithm were permutation groups of degree 276 as constructed by Magma's `MaximalSubgroups` function.

Timings for a selection of simple groups are given in Table 3. Generators for most of the groups in this table are taken from the online Atlas of Finite Group Representations [42]. The generators for those simple groups of Lie type included in Table 3 but which have no generators in the Atlas are provided by the Lie group construction functions in Magma. These generators were compiled by Rylands and Taylor [30].

Table 4 describes the computation of all Brauer characters for the sporadic simple groups Ru, Suz and Co_3 respectively, for all modular characteristics, running on an Intel Xeon Gold 6146 CPU @ 3.2GHz, where the linear algebra is also parallelised in the very large matrix multiplications. This uses a simple variant of the algorithm where the irreducible modules corresponding to the defect-zero characters with degree greater than 5000 are not explicitly constructed, since the corresponding Brauer characters are trivially determined (while the irreducible modules of dimension up to 5000 are still constructed since they may be useful for tensor products). For each group (indicated in

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

Table 3
Timings for calculating the Brauer characters and the irreducible G-modules for a sample of simple groups.

		${}^2F_4(2)'$	McL	He	A_{13}	$O_8^+(2)$	$O_8^-(2)$	$U_3(11)$	$S_4(8)$
	l	5	13	16	21	16	16	20	64
2	d	2048	9856	21504	8008	4096	4096	1440	4096
	t	21	263	3108	3231	49	15	355	43
	l	18	13	22	25	21	19	42	37
3	d	1728	8019	14400	10296	6075	2835	1440	5265
	t	43	2163	3847	596	89	852	5082	7534
	l	20	17	30	41	44	30	32	70
5	d	1300	9625	22050	21450	6075	5355	1440	5265
	t	44	6157	13182	7579	278	1661	2710	21278
	l		20	18	48	52	36		44
7	d		10395	23324	15015	4200	5355		4096
	t		14050	7574	24532	804	3344		20170
	l		22		54			41	
11	d		10395		21450			1331	
	t		24974		19237			361	
	l	20			53				68
13	d	1374			21450				5265
	t	105			19232				77222
	l			31			35		
17	d			20825			5355		
	t			85975			5867		
	l							36	
37	d							1332	
	t							13354	

the column labelled by 'G') the columns labelled by p, '#C', '#M', 'Time' and 'Mem' give each modular characteristic p, the number of p-regular classes (and thus the number of Brauer characters mod p), the number of non-isomorphic irreducible modules actually constructed, the total run time (in hours) and total memory usage (in gigabytes), respectively. The other columns are as follows:

- The columns headed by 'Max Tensor' describe the most expensive step which uses the modular condensation algorithm applied to a tensor product (Subsec. 4.2); here $d_\otimes$ labels the dimension of the virtual tensor product module T and d_{S_p} labels the dimension of the largest irreducible constituent (not already constructed at that point) which is extracted from T.
- The columns headed by 'Max Rational' describe the most expensive step which uses the rational reduction algorithm (Subsec. 4.3); here d_ρ labels the degree of the virtual permutation or induced rational representation ρ which is selected by the algorithm, d_χ labels the degree of the given irreducible rational character χ afforded by a constituent of ρ, and d_{S_p} labels the dimension of the largest irreducible

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

Table 4
Computation of the Brauer characters for larger simple groups.

G	p	# C	# M	Max Tensor		Max Rational			Time (hrs)	Mem (GB)
				$d_\otimes$	d_{S_p}	d_ρ	d_χ	d_{S_p}		
Ru	2	9	9	34888	16036	417600	102400	102400	1.4	25
	3	29	16	(Not used)		579072	91350	91350	5.4	59
	5	28	22	142884	65975	316680	95004	58099	8.2	60
	7	32	13	164836	80650	377000	75400	75400	4.0	59
	13	32	17	142884	75342	1319500	110592	23346	76.6	157
	29	34	17	142884	12531	593775	98280	85749	102.3	250
Suz	2	17	17	31240	9328	370656	79872	79872	1.6	12
	3	20	20	306240	160380	673596	93555	5103	16.9	150
	5	35	22	51909	40040	926640	243243	116127	54.3	259
	7	39	19	283920	183052	370656	146432	135707	35.8	249
	11	42	13	20449	3432	648648	189540	125099	77.2	211
	13	41	13	20449	5005	720720	197120	191181	101.1	230
Co3	2	16	16	206184	88000	655776	226688	131584	3.2	77
	3	22	22	97020	55891	860706	184437	93312	5.2	145
	5	33	24	246400	173075	170775	73600	73325	11.0	146
	7	39	27	162932	98164	860706	184437	154836	16.2	253
	11	38	22	20608	22999	708400	26082	25186	18.7	164
	23	40	19	245504	158753	708400	93312	83687	57.8	311
A_{16}	2	37	37	826880	536576	1681680	582400	501760	47.6	350

mod-p constituent (not already constructed at that point) which is extracted from the corresponding virtual integral module S.

In addition, the last line of the table describes the computation of all mod-2 irreducible modules and their Brauer characters for A_{16}, with the same information given for the most expensive tensor product condensation and rational reduction steps.

Acknowledgment

The authors wish to thank Derek Holt for many helpful suggestions.

References

[1] Martin Albrecht, Gregory Bard, William Hart, Algorithm 898: efficient multiplication of dense matrices over GF(2), ACM Trans. Math. Software 37 (1) (2010).
[2] Wieb Bosma, John Cannon, Catherine Playoust, The Magma algebra system. I. The user language, in: Computational Algebra and Number Theory, London, 1993, J. Symbolic Comput. 24 (3–4) (1997) 235–265.
[3] Richard Brauer, A note on theorems of Burnside and Blichfeldt, Proc. Amer. Math. Soc. 15 (1964) 31–34.
[4] Herbert Brückner, Algorithmen für endliche auflösbare Gruppen und Anwendungen, PhD thesis, RWTH-Aachen, 1998.
[5] John J. Cannon, Bruce C. Cox, Derek F. Holt, Computing the subgroups of a permutation group, in: Computational Algebra and Number Theory, Milwaukee, WI, 1996, J. Symbolic Comput. 31 (1–2) (2001) 149–161.

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

[6] John J. Cannon, Derek F. Holt, Computing chief series, composition series and socles in large permutation groups, in: Computational Algebra and Number Theory, London, 1993, J. Symbolic Comput. 24 (3–4) (1997) 285–301.
[7] John J. Cannon, Derek F. Holt, Computing maximal subgroups of finite groups, J. Symbolic Comput. 37 (5) (2004) 589–609.
[8] John J. Cannon, Derek F. Holt, Computing conjugacy class representatives in permutation groups, J. Algebra 300 (1) (2006) 213–222.
[9] John J. Cannon, Derek F. Holt, Michael Slattery, Allan K. Steel, Computing subgroups of bounded index in a finite group, J. Symbolic Comput. 40 (2) (2005) 1013–1022.
[10] V. Dabbaghian, J.D. Dixon, Computing matrix representations, Math. Comp. 79 (271) (2010) 1801–1810.
[11] J.D. Dixon, High speed computation of group characters, Numer. Math. 10 (1967) 446–450.
[12] Anne Henke, Gerhard Hiss, Jürgen Müller, The 7-modular decomposition matrices of the sporadic O'Nan group, J. Lond. Math. Soc. (2) 60 (1) (1999) 58–70.
[13] G. Hiss, C. Jansen, K. Lux, R.A. Parker, Computational modular character theory, http://www.math.rwth-aachen.de/MOC/CoMoChaTa.
[14] Gerhard Hiss, Jürgen Müller, Felix Noeske, Jon Thackray, The Brauer characters of the sporadic simple Harada-Norton group and its automorphism group in characteristics 2 and 3, LMS J. Comput. Math. 15 (2012) 257–280.
[15] Derek F. Holt, Bettina Eick, Eamonn A. O'Brien, Handbook of Computational Group Theory, Discrete Mathematics and Its Applications (Boca Raton), Chapman & Hall/CRC, Boca Raton, FL, 2005.
[16] Derek F. Holt, Sarah Rees, Testing modules for irreducibility, J. Aust. Math. Soc. A 57 (1) (1994) 1–16.
[17] I. Martin Isaacs, Character Theory of Finite Groups, Dover Publications, Inc., New York, 1994. Corrected reprint of the 1976 original, Academic Press, New York.
[18] Christoph Jansen, Klaus Lux, Richard Parker, Robert Wilson, An Atlas of Brauer Characters, London Mathematical Society Monographs. New Series, vol. 11, The Clarendon Press, Oxford University Press, New York, 1995, Appendix 2 by T. Breuer and S. Norton, Oxford Science Publications.
[19] Keller-Gehrig Walter, Fast algorithms for the characteristic polynomial, Theoret. Comput. Sci. 36 (2–3) (June 1985) 309–317.
[20] Jeffrey S. Leon, Computing automorphism groups of error-correcting codes, IEEE Trans. Inform. Theory 28 (3) (1982) 496–511.
[21] Klaus Lux, Herbert Pahlings, Representations of Groups: A Computational Approach, Cambridge Studies in Advanced Mathematics, vol. 124, Cambridge University Press, Cambridge, 2010.
[22] Klaus Lux, Markus Wiegelmann, Condensing tensor product modules, in: The Atlas of Finite Groups: Ten Years on, Birmingham, 1995, in: London Math. Soc. Lecture Note Ser., vol. 249, Cambridge Univ. Press, Cambridge, 1998, pp. 174–190.
[23] Jürgen Müller, Max Neunhöffer, Frank Röhr, Robert Wilson, Completing the Brauer trees for the sporadic simple Lyons group, LMS J. Comput. Math. 5 (2002) 18–33 (electronic).
[24] Jürgen Müller, 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, in: Progr. Math., vol. 173, Birkhäuser, Basel, 1999, pp. 309–321.
[25] Gabriel Navarro, Characters and Blocks of Finite Groups, London Mathematical Society Lecture Note Series, vol. 250, Cambridge University Press, Cambridge, 1998.
[26] Felix Noeske, Tackling the generation problem in condensation, J. Algebra 309 (2) (2007) 711–722.
[27] R.A. Parker, The computer calculation of modular characters (the meat-axe), in: Computational Group Theory, Durham, 1982, Academic Press, London, 1984, pp. 267–274.
[28] R.A. Parker, R.A. Wilson, The computer construction of matrix representations of finite groups over finite fields, J. Symbolic Comput. 9 (1990) 583–590.
[29] A.J.E. Ryba, Computer condensation of modular representations, J. Symbolic Comput. 9 (5–6) (1990) 591–600, Computational Group Theory, Part 1.
[30] L.J. Rylands, D.E. Taylor, Matrix generators for the orthogonal groups, J. Symbolic Comput. 25 (1998) 351–360.
[31] Gerhard J.A. Schneider, Dixon's character table algorithm revisited, J. Symbolic Comput. 9 (5–6) (1990) 601–606, Computational Group Theory, Part 1.
[32] Charles C. Sims, Computational methods in the study of permutation groups, in: Computational Problems in Abstract Algebra, Proc. Conf., Oxford 1967, Pergamon, Oxford, 1970, pp. 169–183.

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015

[33] Charles C. Sims, Computation with permutation groups, in: SYMSAC '71 Proceedings of the Second ACM Symposium on Symbolic and Algebraic Manipulation, ACM, 1971, pp. 23–28.
[34] Charles C. Sims, Determining the conjugacy classes of a permutation group, in: Computers in Algebra and Number Theory, Proc. SIAM-AMS Sympos. Appl. Math., New York, 1970, in: SIAM–AMS Proc., vol. IV, Amer. Math. Soc., Providence, R.I., 1971, pp. 191–195.
[35] A.K. Steel, Construction of Ordinary Irreducible Representations of Finite Groups, PhD thesis, University of Sydney, 2012.
[36] Strassen Volker, Gaussian elimination is not optimal, Numer. Math. 13 (1969) 354–356.
[37] Ibrahim A.I. Suleiman, Robert A. Wilson, The 2-modular characters of Conway's group co2, Math. Proc. Cambridge Philos. Soc. 116 (2) (1994) 275–283.
[38] Ibrahim A.I. Suleiman, Robert A. Wilson, The 2-modular characters of Conway's third group Co_3, J. Symbolic Comput. 24 (3–4) (1997) 493–506.
[39] J.G. Thackray, Modular Representations of Some Finite Groups, PhD thesis, University of Cambridge, 1981.
[40] William R. Unger, Computing the character table of a finite group, J. Symbolic Comput. 41 (8) (2006) 847–862.
[41] Robert A. Wilson, Construction of finite matrix groups, in: Computational Methods for Representations of Groups and Algebras, Essen, 1997, in: Progr. Math., vol. 173, Birkhäuser, Basel, 1999, pp. 61–83.
[42] Robert A. Wilson, et al., Atlas of finite group representations - version 3, http://brauer.maths.qmul.ac.uk/Atlas/v3/.

Please cite this article in press as: J.J. Cannon et al., Construction of the irreducible modular representations of a finite group, J. Algebra (2019), https://doi.org/10.1016/j.jalgebra.2019.08.015