Journal of Algebra 322 (2009) 948–956

Contents lists available at ScienceDirect

Journal of Algebra

www.elsevier.com/locate/jalgebra

Decomposing homogeneous modules of finite groups in characteristic zero

Bernd Souvignier

Institute for Mathematics, Astrophysics and Particle Physics, Radboud University Nijmegen, Postbus 9010, 6500 GL Nijmegen, The Netherlands

ARTICLE INFO

Article history:
Received 16 January 2009
Available online 26 May 2009
Communicated by Jon Carlson

Dedicated to John Cannon and Derek Holt on the occasions of their significant birthdays, in recognition of distinguished contributions to mathematics

Keywords:
Homogeneous modules
Endomorphism ring
Singular elements

ABSTRACT

This paper discusses the decomposition of representations of finite groups in characteristic zero, with special emphasis on homogeneous modules. An improved method to compute the endomorphism ring of a representation is presented and a novel algorithm to find singular elements in the endomorphism ring is given. Several explicit examples illustrate the practicality and scope of the various techniques.

1. Introduction

In this paper, we discuss the problem of decomposing reducible modules of finite groups in characteristic zero, with special emphasis on homogeneous modules, i.e. sums of isomorphic irreducible modules. This type of modules has so far not been subject of a systematic treatment, although some ideas have been suggested.

Let G be a finite group and $\Delta : G \to GL_n(K)$ a representation of G over a field K which turns K^n into a KG-module.

Over finite fields, the *Meataxe* (see [13] and [8]) provides a powerful tool to decompose reducible modules into their irreducible constituents. Unfortunately, the adaptation of the Meataxe to characteristic zero is not straightforward, since one may face the situation to test for infinitely many vectors whether they lie in a proper submodule. However, in many situations the following decomposition techniques can be applied successfully in characteristic zero:

E-mail address: b.souvignier@math.ru.nl.

0021-8693/$ – see front matter
doi:10.1016/j.jalgebra.2009.05.008

- Direct Meataxe methods (see [14] and [17]);
- Reconstruction from modular decompositions (as described in [7]);
- Factoring minimal polynomials of elements in the center of the endomorphism ring (cf. [15]).

Common to all approaches mentioned so far is that they fail to split homogeneous modules (except for lucky exceptions). In this situation, the endomorphism ring

$$End(\Delta) := \left\{ X \in K^{n\times n} \;\middle|\; X\Delta(g) = \Delta(g)X \text{ for all } g \in G \right\}$$

is isomorphic to a full matrix ring of degree $m > 1$ over some division algebra. In particular, $End(\Delta)$ contains singular elements which allow to find proper submodules. In order to follow this route, we require an effective method to compute $End(\Delta)$.

2. Iteration method

For representations of degree above 100 it is clear that a direct computation of $End(\Delta)$ by solving a system of linear equations is impractical. Two alternative approaches have been suggested:

- In [7], D. Holt sketches an approach based on the Meataxe: For a singular element $u \in \Delta(KG)$ the action of $X \in End(\Delta)$ on the nullspace of u is constructed and then extended to the action on the full module via translates under G.
- In [15] an *iteration method* is presented which constructs $X \in End(\Delta)$ by averaging over a generating set and iterating this process.

In this section we report on improvements that have been made to the iteration method, resulting in a method that allows to compute $End(\Delta)$ over $\mathbb{Q}$ efficiently for degrees beyond 1000 and which can also be applied over (small) algebraic number fields K.

The core of the iteration method is Theorem 2.1 in [15] which states that the averaging operator

$$\rho : K^{n\times n} \to End(\Delta), \qquad X \mapsto \frac{1}{|G|} \sum_{g\in G} \Delta(g) X \Delta(g)^{-1}$$

can be approximated by iterating the averaging operator over a generating set $\{g_1, \ldots, g_s\}$ of G:

$$X_0 := X, \qquad X_{k+1} := \frac{1}{s} \sum_{i=1}^{s} \Delta(g_i) X_k \Delta(g_i)^{-1} \quad \Rightarrow \quad \lim_{k\to\infty} X_k = \rho(X).$$

The iteration method can be carried out using either floating point arithmetic or rational arithmetic. In the latter case, intermediate rounding is necessary to avoid entry swell. In any case, rounding errors spoil the approximation if too many iterations are required. Therefore it is crucial to have fast convergence.

2.1. Acceleration via the product replacement algorithm

For a fixed generating set, it is proved in [15] that the iteration method converges asymptotically with a constant contraction factor $0 < \delta < 1$. Therefore, a natural idea to accelerate convergence is to apply Aitken's δ^2-process given by

$$x_i' = x_i - \frac{(x_{i+1} - x_i)^2}{x_{i+2} - 2x_{i+1} + x_i}$$

Table 1
Convergence speed σ_n/σ_{n-10} for M_{11}.

n	20	30	40	50	60	70	80
Fixed generators	4.04	3.41	3.32	3.30	3.29	3.29	3.28
Product replacement	97.94	73.21	80.44	123.07	98.44	186.77	142.09
	97.82	68.68	128.07	59.47	60.41	19.08	115.40
	162.96	125.54	136.99	131.35	146.59	160.19	164.49

(see e.g. Section 5.10 in [18]) to the matrix elements. However, this actually tends to deteriorate the convergence, since in fact only the overall convergence for the full matrix is geometric, whereas the single entries show irregular fluctuations.

Much better results yields the *product replacement algorithm*, due to C.R. Leedham-Green (see [3]): Starting with a generating set $\{g_1, \dots, g_s\}$, two random numbers $i \neq j$ between 1 and s are produced and g_i is replaced by one of $g_i g_j^{\pm 1}$ or $g_j^{\pm 1} g_i$. This process produces – after an initialization phase – a good series of random elements from a group. An excellent overview and discussion of the product replacement algorithm is given in [12].

Our key idea is to apply the iteration method not for a fixed generating set but for a generating set that is changed by the product replacement algorithm after each iteration step.

Experiment 2.1. A quick comparison of the convergence rates can be performed by some computations in the group ring $\mathbb{Q}G$. As a measure for the quality of an approximation $\sum_{g\in G} c_g g \in \mathbb{Q}G$ with $\sum_{g\in G} c_g = 1$ to $\rho = \frac{1}{|G|}\sum_{g\in G} g$ we take the standard deviation $\sigma = (\frac{1}{|G|}\sum_{g\in G}(c_g - \frac{1}{|G|})^2)^{\frac{1}{2}}$ of the coefficients from the expectation value $\frac{1}{|G|}$.

For a group with generating set $\{g_1, \dots, g_s\}$ we compute the standard deviation σ_n of $\frac{1}{(s+1)^n}(1 + g_1 + \cdots + g_s)^n$ for $n = 10, 20, 30$ etc. and record the decrease σ_n/σ_{n-10}. The same is done for a generating system to which the product replacement algorithm is applied, i.e. for $\prod_{k=1}^{n} \frac{1}{s+1}(1 + g_1^{(k)} + \cdots + g_s^{(k)})$ where $g_i^{(k)}$ is the ith generator in the kth iteration step.

Example 2.2. For the Mathieu group $G = M_{11}$ we fix a generating set with generators g_1, g_2, g_3 of orders 11, 6 and 4 (a generating set with a good convergence rate). The convergence results are displayed in Table 1, giving three runs of the product replacement version, since the behaviour is influenced by random choices.

Analyzing the data one sees that the quality obtained with the fixed generators after 80 iterations is already reached after 30 iterations with the product replacement method.

Note that the iteration with fixed generators indeed shows the expected geometric series convergence, whereas the behaviour for the product replacement is irregular.

The key for the superior convergence behaviour with the product replacement algorithm lies in the fact that the length of the generators $g_i^{(k)}$ as words in the original generators grows exponentially with the number k of iterations.

Proposition 2.3. *Let G be generated by $\{g_1, \dots, g_s\}$. Then after k iterations of the product replacement algorithm, the expectation value for the length of the generators $g_i^{(k)}$ as words in the original generators is*

$$\left(\frac{s+1}{s}\right)^k.$$

Proof. This is clear for $k = 1$, since there are $s - 1$ generators of length 1 and one of length 2. Assume now that a generating set after k iterations has lengths $(l_1, l_2, \dots, l_s)$ and thus average length $\bar{l} = \frac{1}{s}\sum_{i=1}^{s} l_i$. Replacing the ith generator by its product with the jth generator gives a length tuple

Table 2
Bias $\frac{|G|}{s^n}c_1^{(n)}$ towards the identity for M_{11}.

n	10	15	20	25	30	40	50	60
Fixed generators	13.28	6.49	3.41	2.04	1.39	1.07	1.02	1.003
Product replacement	0.80	1.01	1.003	1.0001	1.	1.	1.	1.
	1.07	0.97	0.999	0.9999	1.	1.	1.	1.
	0.67	0.99	0.997	0.9999	1.	1.	1.	1.

of the form $(l_1, \dots, l_i + l_j, \dots, l_s)$. Summing over the average lengths in all tuples for $i \neq j$, we get $\sum_{i=1}^{s} \sum_{j\neq i} (\bar{l} + \frac{1}{s} l_j) = s(s-1)\bar{l} + (s-1)\bar{l} = (s^2-1)\bar{l}$. To get the average length, we have to divide by $s(s-1)$. This gives $\frac{s^2-1}{s(s-1)}\bar{l} = \frac{s+1}{s}\bar{l}$, thus the average length has been multiplied by $\frac{s+1}{s}$. Since this holds for every generating tuple, the claim is proved. □

Note that the maximal possible length after k iterations is obtained by always multiplying the second-longest generator by the longest one. These maximal lengths clearly give the Fibonacci sequence.

The main obstacle for the convergence with fixed generators is that this method expands the Cayley graph stepwise from the identity element. Therefore, for the first iteration cycles the elements close to the identity element are overrepresented and this bias is only gradually levelled out. In particular, there are too many loops for the identity element.

The following experiment demonstrates this behaviour and shows that it is overcome by the product replacement algorithm, since the long words in the original generators explore all regions of the Cayley graph uniformly after a short initialization phase.

Experiment 2.4. Let $\{g_1, \dots, g_s\}$ be a generating set for G. Then the coefficient $c_g^{(n)}$ of g in $(g_1 + \cdots + g_s)^n$ is the number of paths of length n in the Cayley graph from $1 \in G$ to g. Since for $n \to \infty$ the words of length n in the g_i are uniformly distributed, $\frac{c_g^{(n)}}{s^n}$ converges to $\frac{1}{|G|}$. A bias towards the identity element can be read off from $\frac{|G|}{s^n}c_1^{(n)} > 1$.

Again, we compare the behaviour of the coefficient $c_1^{(n)}$ for a fixed generating set with that for a generating set to which the product replacement algorithm is applied.

Example 2.5. We revisit the Mathieu group $G = M_{11}$ with the same generating set $\{g_1, g_2, g_3\}$ as in Example 2.2. The values of $\frac{|G|}{s^n}c_1^{(n)}$ are displayed in Table 2, again giving three runs for the product replacement version. Entries 1. indicate that the deviation from 1 is less than 10^{-4}.

One sees that for the fixed generators the bias towards the identity element vanishes quite slowly, whereas with the product replacement algorithm there is no bias at all (after initial fluctuations).

We note that there is nothing special about the behaviour of the group M_{11}, an analogous behaviour as in Examples 2.2 and 2.5 is generally observed.

2.2. Iteration over algebraic number fields

In some situations it is desirable to apply the iteration method over an extension field K of $\mathbb{Q}$. Note that one will usually be able to work with a representation Δ written over the maximal order $R = Int(K)$ of K.

With respect to an integral basis $(a_1, \dots, a_d)$ of R, where $d = [K : \mathbb{Q}]$, the elements of R can be written as $\sum_{i=1}^{d} c_i a_i$ with $c_i \in \mathbb{Z}$. Assuming that the group order $|G|$ is known, we multiply the initial element X to which the iteration is applied by $|G|$. Then the limit of the iteration is an element $\rho(X)$ with entries in R, which can be identified by rounding the coefficients c_i to integers.

In the case that Δ is not written over the maximal order R, it is always possible to transform Δ to a representation with small denominators. Then rounding the c_i to rational numbers can still be easily achieved, either via iterated fractions or simply by trial multiplication.

Note that this approach avoids the problem of reconstructing algebraic numbers from real approximations mentioned in [11].

2.3. Applications

Apart from the computation of $End(\Delta)$, the iteration method has various useful applications, which are obtained via different actions of G (cf. Section 5 in [15]). Amongst these are the computation of the center $Z(End(\Delta))$ of the endomorphism ring, G-invariant bilinear forms and intertwining matrices between two representations. In this section we give two explicit examples illustrating such applications.

2.3.1. Scope of the iteration method

In our applications, the iteration process using the product replacement method usually allows to read off the desired average element after 40–50 iterations. Working in dimensions up to 500, endomorphisms are thus obtained within at most a minute. In order to explore the scope of the improved method, we took the challenge of decomposing a representation of degree 2752.

Example 2.6. The representation of degree 152 with character 43*defg* of $G = U_3(7)$ (in Atlas notation, see [4]) is recorded in [11] as one of the rational representations of degree below 250 which is difficult to obtain. It occurs as a constituent of a permutation representation of degree 2752 with character decomposing as $1 + 43a + 43bc + 43defg + 301a + 301bc + 301defg + 343a$. Obtaining an element $X \in Z(End(\Delta))$ finished after 40 iterations and took about 12 minutes (on 2.2 GHz Linux-PC). The minimal polynomial of X had four linear, two quadratic and two quartic factors, in accordance with the numbers of algebraic conjugates in the character decomposition and thus yielded a complete decomposition into rationally irreducible modules. The quartic factor relevant for the character 43*defg* was $f = t^4 + 34698209312t^2 - 9160476825047184t + 603292103372399444497$.

2.3.2. Galois descent

A frequent problem in the construction of representations is to transform a representation given over some field L to one over a smaller field K. This can be achieved by either spinning up a vector in the kernel of a suitable element in $\Delta(LG)$ of minimal nullity (see [17]) or, in the case that L/K is a cyclic Galois extension with Galois group generated by σ, by a *Galois descent*. The latter requires an intertwining matrix X such that $X\Delta(g)X^{-1} = \Delta^\sigma(g)$ for all $g \in G$ and $X^{\sigma^{r-1}} \dots X^\sigma X = 1$.

A matrix X inducing the Galois automorphism can be obtained efficiently via the iteration method (in its number field version). One then has to solve a relative norm equation, since for the intertwining matrix one will only have $N(X) := X^{\sigma^{r-1}} \dots X^\sigma X = A$ for some element $A \in End(\Delta)$.

Remark 2.7. In the Galois descent as described in [2,5,6], it is assumed that Δ is absolutely irreducible. In this case, $N(X) = X^{\sigma^{r-1}} \dots X^\sigma X$ is a scalar matrix αI_n and X is adjusted to $X' := \lambda^{-1}X$ by an element $\lambda \in L$ for which $N_{L/K}(\lambda) = \alpha$.

The same actually still works for a representation which is only assumed to be irreducible over L, but not necessarily absolutely irreducible. In this case the norm equation has to be solved in the relative extension L'/K' where K' is the character field of Δ and $L' = L \otimes_K K'$ (and thus $L' \cong End_L(\Delta)$).

Example 2.8. The rational character 35*abc* of $Sz(8)$ is most easily constructed as the symmetric tensor square $14a^{[2]}$. Since the character 14*a* has character field $\mathbb{Q}(i)$, this yields a representation Δ over $\mathbb{Q}(i)$ that can be realized over $\mathbb{Q}$.

Via the iteration method, applied for $\mathbb{Q}(i)$, we obtain an intertwining matrix X inducing the Galois automorphism $\sigma : i \mapsto -i$ after 48 iterations. The matrix $A = X^\sigma X$ has minimal polynomial $\mu_A = t^3 - 166231t^2 + 7686209739t - 70028664774965$, but is not a scalar matrix, since Δ is not absolutely

irreducible. The roots of μ_A generate the character field $K' := \mathbb{Q}(\zeta_{13}+\zeta_{13}^{-1}+\zeta_{13}^{5}+\zeta_{13}^{-5})$ of the character 35*a*. Taking $L' = K'(i)$, we require to solve the norm equation $N_{L'/K'}(\lambda) = \alpha$, where $\alpha \in K'$ is the element corresponding with A. The norm equation algorithm in MAGMA [1] yields such a solution λ within a second. Resubstituting $X^\sigma X$ for α in λ^{-1} finally gives a matrix Y such that $Z := XY$ fulfills $Z^\sigma Z = 1$.

3. Splitting homogeneous modules

Assuming that we have the endomorphism ring $End(\Delta)$ at our disposal, we now address the problem of decomposing a homogeneous module. In this case we have $End(\Delta) \cong D^{m\times m}$ for a division algebra D. Clearly, we can assume the dimension $z := \dim_K Z(End(\Delta)) = \dim_K Z(D)$ of the center of $End(\Delta)$ to be known and by [10] we can also obtain the Schur index s of D and thus via $\dim_K End(\Delta) = z s^2 m^2$ also the multiplicity m. In this section we assume throughout that $m > 1$.

In [15], the following methods have been suggested to find singular elements in $End(\Delta)$:

(1) computing elements in the isotropic subspace of the trace bilinear form;
(2) solving relative norm equations;
(3) computing degenerate invariant forms.

Unfortunately, all these approaches are only applicable in the case $End(\Delta) = L^{m\times m}$ for $L = Z(End(\Delta))$, moreover, (1) requires $m = 2$ and $L = \mathbb{Q}$, (2) only deals with $m = 2$ or 3, and (3) only deals with m odd and real-valued irreducible characters.

In order to have a general decomposition method for reducible homogeneous modules available, we present an algorithm which finds singular elements as vectors of small norm in a suitable lattice derived from a maximal order in $End(\Delta)$.

We restrict ourselves to representations over $\mathbb{Q}$. If an irreducible representation over a field extension K of $\mathbb{Q}$ is desired, this can be easily obtained from a rationally irreducible representation via factoring the minimal polynomial of an element of $End(\Delta)$ over K.

3.1. Singular elements as short vectors in a maximal order

The guiding principle inspiring this method is the fact that $\mathbb{Z}^{m\times m}$ is a maximal order in $\mathbb{Q}^{m\times m}$ which contains the elementary matrices (having a single entry 1 and the rest 0) and that the elementary matrices are singular elements having norm 1 for the bilinear form $\Phi(A, B) := \mathrm{tr}(AB^{\mathrm{tr}})$ on $\mathbb{Q}^{m\times m}$.

Since we assume to deal with a reducible homogeneous module, we have $End(\Delta) \cong D^{m\times m}$ for a division algebra D. For a maximal order Λ in D, $\Lambda^{m\times m}$ is a maximal order in $D^{m\times m}$ and in case D is a principal ideal domain, all maximal orders of $D^{m\times m}$ are conjugate to $\Lambda^{m\times m}$ (see Theorem 21.6 in [16]). Moreover, by Corollary 27.6 of [16], every maximal order contains elementary matrices with a single nonzero entry from Λ and these are the elements we aim at.

We will deal with a right regular representation of $End(\Delta)$, therefore conjugacy is of no concern to us. The first step is thus to construct a maximal order in $End(\Delta)$. An efficient method for this task is given in [10]: One first computes a hereditary order by the *radical idealizer process*, then a maximal overorder is obtained as the iterated idealizer of a maximal ideal.

Since we are dealing with a regular representation of $End(\Delta)$, a badly chosen basis for the maximal order Γ may still hide the singular elements. However, the following simple observation shows that a regular representation with respect to a bad basis can be improved by standard lattice reduction techniques like LLL (cf. [9]).

Note that we can restrict ourselves to the case $\Gamma \subseteq \mathbb{Q}^{m\times m}$ by replacing D by its regular representation of degree $d = \dim_{\mathbb{Q}}(D)$.

Remark 3.1. Let $\Gamma \subseteq \mathbb{Q}^{m\times m}$ be a $\mathbb{Z}$-order with $\mathbb{Z}$-basis $B = (B_1, \ldots, B_n)$ and let ρ be the right regular representation of Γ w.r.t. B. Let $C \in \mathbb{Q}^{n\times m^2}$ be the matrix with B_i as ith row (writing an $m \times m$ matrix as a row of length m^2).

Then for $X \in \Gamma$, the jth row of $\rho(X)$ is the solution x of $xC = B_j X$, again interpreting $B_j X$ as a row of length m^2. Since B is a basis, the matrix C has a pseudoinverse $C^+ \in \mathbb{Q}^{m^2 \times n}$ with $CC^+ = I_n$ and we have $x = B_j X C^+$.

This shows that for a matrix $E \in \Gamma$ with small entries, the typical size of the entries in $\rho(B_i)$ exceeds that of the entries in $\rho(E)$ by the difference of the sizes of the entries in B_i and in E. Thus, in the case of a bad basis B for Γ, applying LLL-reduction to the basis $(\rho(B_1), \ldots, \rho(B_n))$ will result in a basis with smaller entries.

Algorithm 3.2. *Improve regular representation*

Input: Basis B of an order Γ in its right regular representation.
Output: Improved basis of Γ.
Algorithm:
Step 1: Apply LLL-reduction to the lattice with basis B, writing $n \times n$ matrices as vectors of length n^2.
Step 2: Compute the right regular representation for the LLL-reduced basis. If required, iterate.

We give an example illustrating the effectiveness of Algorithm 3.2. Starting with a bad basis of $\mathbb{Z}^{3\times 3}$ we obtain a basis containing singular elements after three iterations.

Example 3.3. We apply random elementary row and column operations to a 9×9 identity matrix until the average of the elements exceeds 100. The rows of this matrix are then taken as initial basis of $\mathbb{Z}^{3\times 3}$, three elements of this basis are given below:

$$\begin{pmatrix} 26 & 2 & -6 \\ -42 & 29 & -24 \\ 12 & 95 & -13 \end{pmatrix}, \quad \begin{pmatrix} 131 & 18 & -1 \\ 153 & 181 & -433 \\ -228 & -265 & 196 \end{pmatrix}, \quad \ldots, \quad \begin{pmatrix} -179 & 5 & 3 \\ -161 & -246 & 575 \\ 305 & 305 & -254 \end{pmatrix}.$$

Let the *norm* of an element be the sum of the squares of its entries in the right regular representation (w.r.t. to a given basis). Then the norms of the initial basis range between $8.6 \cdot 10^{15}$ and $5.3 \cdot 10^{17}$. The norm of the elementary matrix E_{11} with respect to this basis is $2.7 \cdot 10^{12}$.

Applying the improvement algorithm, we get after the first iteration a basis with norms between 9 (for the identity matrix) and 1536666, after the second iteration a basis with norms between 9 and 342 and after the third iteration a basis with norms between 4 and 45. Clearly, elements of norm 4 have to be singular.

Combining an algorithm to construct a maximal order with Algorithm 3.2, we get the following method to compute singular elements in $End(\Delta) \cong D^{m\times m}$ for $m > 1$.

Algorithm 3.4. *Find singular element*

Input: A basis of $End(\Delta)$.
Output: A singular element $Y \in End(\Delta)$.
Algorithm:
Step 1: Compute a basis B of a maximal order Γ in $End(\Delta)$ by the algorithm given in [10].
Step 2: If the norms $\operatorname{tr}(\rho(B_i)\rho(B_i)^{\mathrm{tr}})$ of all basis elements B_i in the right regular representation ρ of Γ are $\geqslant \dim End(\Delta)$, improve the right regular representation by Algorithm 3.2.
Step 3: Compute the short vectors of Γ (w.r.t. the norm $\operatorname{tr}(\rho(A)\rho(A)^{\mathrm{tr}})$) up to $\dim End(\Delta)$. For an element X with reducible minimal polynomial $\mu_X = f_1 \cdot \cdots \cdot f_s$, return $Y := f_1(X)$.

Note that applying LLL-reduction to the bases of the intermediate orders in Step 1 typically results in a fairly good basis for the maximal order. Therefore the improvement stage (Step 2) is often not required at all or just once (see Table 3).

Table 3
Performance of Algorithm 3.4 for finding singular elements in $End(\Delta)$.

Group	char	$\deg\Delta$	$\dim_{\mathbb{Q}} E$	$\dim_{\mathbb{Q}} Z$	m	Maxord	Improve
$L_2(23)$	$2\cdot 22de$	88	8	2	2	23^3	1
$L_2(27)$	$2\cdot 26def$	156	12	3	2	$3^3 7^2$	1
$Sz(8)$	$2\cdot 35abc$	210	12	3	2	2^4	0
$Sz(8)$	$2\cdot 65abc$	390	12	3	2	$2^3 7^2$	1
$L_2(11)$	$3\cdot 10a$	30	9	1	3	11^2	0
$L_2(11)$	$4\cdot 11a$	44	16	1	4	$2^4 3^2$	0
$L_2(11)$	$3\cdot 12ab$	72	18	2	3	11^3	1
$SL_2(7)$	$4\cdot 6bc$	48	32	2	2	$2^2 7^2$	2
$SL_2(7)$	$6\cdot 8b$	48	36	1	3	$3^1 7^2$	2

3.2. Examples

In this section we present a number of typical situations in which reducible homogeneous modules occur and illustrate how Algorithm 3.4 performs on a collection of examples for these cases.

(1) Many irreducible representations in characteristic zero are most easily obtained as constituents of permutation representations (cf. [11]). The problematic case are irreducible modules that do not occur with multiplicity 1 in any permutation representation of the group.
We give two examples of this situation, the character $22de$ (in Atlas notation) for the group $L_2(23)$, and the character $26def$ for $L_2(27)$, for which the rational representation is currently not available in [19]. These characters occur with multiplicity $m = 2$ in permutation representations of degrees 253 and 351, respectively.

(2) If an irreducible representation that can be realized over a field K is actually written over a field extension L of K, this yields a homogeneous module in which the irreducible representation occurs with multiplicity $m = [L : K]$.
An instance of this situation was already presented in Example 2.8. If the symmetric tensor square $14a^{[2]}$ of $Sz(8)$, written over $\mathbb{Q}(i)$, is inflated to a rational representation, one obtains a homogeneous module with character $2\cdot 35abc$.
As a second example, we note that the characters $65a$, $65b$, $65c$ of $Sz(8)$ can be obtained by inducing a nontrivial linear character of the Frobenius subgroup $2^{3+3}:7$ to $Sz(8)$. This yields a representation over $\mathbb{Q}(\zeta_7)$ (as e.g. contained in [19]) that can actually be realized over the real subfield of $\mathbb{Q}(\zeta_7)$. Inflating to a rational representation results in a representation with character $2\cdot 65abc$.

(3) Restricting representations to subgroups often yields homogeneous modules with higher multiplicities. We give an example where multiplicities $m = 3$ and $m = 4$ occur:
Restricting the representation $176a$ of M_{12} to the maximal subgroup $L_2(11)$ gives the character $5ab + 3\cdot 10a + 2\cdot 10b + 4\cdot 11a + 3\cdot 12ab$ of $L_2(11)$.

(4) Induction from (small) subgroups is also a source for reducible homogeneous modules. We give an example in which representations with nontrivial Schur index occur with multiplicities $m > 1$: For $G = SL_2(7)$ we induce the nontrivial rational representation of a cyclic subgroup of order 3 to G. This gives a rational representation of degree 224 with character $2\cdot 3ab + 2\cdot 4ab + 4\cdot 6a + 4\cdot 6bc + 4\cdot 7a + 6\cdot 8a + 6\cdot 8b$. We are particularly interested in the homogeneous modules with characters $4\cdot 6bc$ and $6\cdot 8b$, since they involve irreducible characters with Schur index $s = 2$.

In Table 3 we display how our algorithm performs on the examples just described. The columns of the table give the group, the character of the homogeneous module, the degree $\deg\Delta$ of the rational representation Δ, the dimension $\dim_{\mathbb{Q}} E$ of the endomorphism ring, the dimension $\dim_{\mathbb{Q}} Z$ of the center of the endomorphism ring and the multiplicity m with which the rationally irreducible module occurs. Note that from this data the rational Schur index s can be read off, since $\dim_{\mathbb{Q}} E = \dim_{\mathbb{Q}} Z \cdot (sm)^2$. The column with heading *Maxord* describes the steps by which a maximal order is obtained

from the order $End(\Delta) \cap \mathbb{Z}^{n\times n}$. A symbol of the form $p_1^{s_1}\, p_2^{s_2} \ldots p_r^{s_r}$ indicates that for the prime p_i the order was enlarged in s_i steps. Finally, the column with heading *Improve* displays by how many iterations of Algorithm 3.2 the regular representation was improved.

Acknowledgment

The author wishes to thank G. Nebe for fruitful discussions and the referees for their constructive comments.

References

[1] W. Bosma, J. Cannon, C. Playoust, The Magma algebra system. I. The user language, J. Symbolic Comput. 24 (1997) 235–265.
[2] H. Brückner, Algorithmen für endliche auflösbare Gruppen und Anwendungen (Algorithms for finite soluble groups and applications), PhD thesis, RWTH Aachen, 1998, 102 pp.
[3] F. Celler, C.R. Leedham-Green, S. Murray, A. Niemeyer, E.A. O'Brien, Generating random elements of a finite group, Comm. Algebra 23 (1995) 4931–4948.
[4] J.H. Conway, R.T. Curtis, S.P. Norton, R.A. Parker, R.A. Wilson, An Atlas of Finite Groups, Oxford University Press, 1998.
[5] C. Fieker, Minimizing representations over number fields, J. Symbolic Comput. 38 (2004) 833–842.
[6] S.P. Glasby, R.B. Howlett, Writing representations over minimal fields, Comm. Algebra 25 (1997) 1703–1711.
[7] D.F. Holt, The Meataxe as a tool in computational group theory, in: R. Curtis, R. Wilson (Eds.), The Atlas of Finite Groups: Ten Years On, Cambridge University Press, 1998, pp. 74–81.
[8] D.F. Holt, S. Rees, Testing modules for irreducibility, J. Aust. Math. Soc. Ser. A 57 (1994) 1–16.
[9] A.K. Lenstra, H.W. Lenstra Jr., L. Lovász, Factoring polynomials with rational coefficients, Math. Ann. 261 (1982) 515–534.
[10] G. Nebe, A. Steel, Recognition of division algebras, J. Algebra 322 (2009) 903–909.
[11] S.J. Nickerson, An atlas of characteristic zero representations, PhD thesis, University of Birmingham, 2006, 230 pp.
[12] I. Pak, What do we know about the product replacement algorithm? in: W.M. Kantor, A. Seress (Eds.), Groups and Computation III, de Gruyter, Berlin, 2001, pp. 301–347.
[13] R.A. Parker, The computer calculation of modular characters – the Meat-Axe, in: M.D. Atkinson (Ed.), Computational Group Theory, Academic Press, London, 1984, pp. 267–274.
[14] R.A. Parker, An integral 'Meat-axe', in: R. Curtis, R. Wilson (Eds.), The Atlas of Finite Groups: Ten Years On, Cambridge University Press, London, 1998, pp. 215–228.
[15] W. Plesken, B. Souvignier, Constructing rational representations of finite groups, Experiment. Math. 5 (1996) 39–47.
[16] I. Reiner, Maximal Orders, Academic Press, London, 1975.
[17] T. Schulz, Konstruktion rationaler Darstellungen endlicher Gruppen (Construction of rational representations of finite groups), PhD thesis, RWTH Aachen, 2002, 83 pp.
[18] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, third ed., Springer, New York, 2002.
[19] R. Wilson, P. Walsh, J. Tripp, I. Suleiman, S. Rogers, R. Parker, S. Norton, S. Nickerson, S. Linton, J. Bray, R. Abbott, Atlas of finite group representations, available online at http://brauer.maths.qmul.ac.uk/Atlas/v3.