LMS J. Comput. Math. 15 (2012) 140–157 © 2012 Authors
doi:10.1112/S1461157012000071

Condensation of homomorphism spaces

Klaus Lux, Max Neunhöffer and Felix Noeske

Abstract

We present an efficient algorithm for the condensation of homomorphism spaces. This provides an improvement over the known tensor condensation method which is essentially due to a better choice of bases. We explain the theory behind this approach and describe the implementation in detail. Finally, we give timings to compare with previous methods.

1. *Introduction*

Computational methods have been particularly successful in the modular representation theory of sporadic groups. From the days when Parker and Thackray devised the MeatAxe [**9, 12**], to recent progress in the modular Atlas project [**13**], the majority of results can be attributed to the application of computers. However, diverting complex calculations to a machine, while expediting the answer and simultaneously precluding man-made miscalculations, does not mean that a push-of-a-button strategy is always met with success. In fact, most open problems in the modular Atlas project have resisted a direct computational approach when they were first considered. To regain computational tractability, Thackray introduced a method called fixed-point reduction in his PhD thesis [**12**], which allowed him to study large modules by only considering certain subspaces. This method is a special case of what has become known as 'condensation'.

The precise connection is as follows. Let F be a field of characteristic p greater than zero, G a finite group, FG the group algebra, and V a finite-dimensional FG-module. Furthermore, let $e \in FG$ be an idempotent. Then we consider the *condensation functor* $- \cdot e : \mathsf{mod}\text{-}FG \to \mathsf{mod}\text{-}eFGe$, under which V is mapped to Ve and a homomorphism $\varphi \in \mathrm{Hom}_{FG}(V, W)$ mapped to its restriction $\varphi|_{Ve} \in \mathrm{Hom}_{eFGe}(Ve, We)$. We refer to Ve as the *condensed module* of V and e as the *condensation idempotent*. The condensation functor has a number of interesting properties, details of which are given in [**1**, Section 6] or [**11**], for example.

The wide array of different condensation algorithms available for group algebras, providing implementations which allow the condensation of, for instance, permutation modules [**3**], induced modules [**6, 8**] and tensor products of modules [**5, 8**], is testimony to the method's usefulness. In this article we present an efficient algorithm for the condensation of homomorphism spaces of FG-modules for some finite group G. As homomorphism spaces may be viewed as tensor products and vice versa, the method we present also improves the condensation of tensor products of FG-modules.

We fix some further notation which will remain in effect throughout the paper. Let V and W be finite-dimensional FG-modules. Then $\mathrm{Hom}_F(V, W)$ is also a finite-dimensional FG-module, where the action is defined as

$$\varphi g : v \mapsto \varphi(vg^{-1})g, \quad v \in V \tag{1}$$

for all $\varphi \in \mathrm{Hom}_F(V, W)$ and $g \in G$.

We choose a subgroup $K \leqslant G$ whose order is coprime to the characteristic of F. Let Λ denote a one-dimensional FK-module affording the linear representation λ. In the present work we

Received 20 January 2011; revised 1 November 2011.

2010 Mathematics Subject Classification 20C05, 20C20, 20C40 (primary).

condense with idempotents of the form

$$e_\lambda := \frac{1}{|K|}\sum_{k\in K}\lambda(k^{-1})k,$$

that is, e_λ is the central primitive idempotent corresponding to the simple module Λ in the semi-simple group algebra FK. Therefore we call K a *condensation subgroup*.

This paper is structured as follows. After introducing some notation in Section 2, we begin in Section 3 with building the theoretical underpinnings from which we will construct our algorithm in Sections 4–6. Section 5 details the one-off calculations needed to prepare the input for the actual condensation process, which we then describe in Section 6. In particular, in Section 5 we deal with the independently interesting problem of how to quickly calculate a semi-simplicity basis for a module. Section 6 describes the gory details of the condensation routine and analyses its complexity. We show there that the most time-critical part of the computation is improved by a factor equal to the sum of the dimensions of two composition factors. However, since this factor depends on the pair of composition factors considered, and because the condensation algorithm loops over many different pairs, the factor of improvement varies greatly across different parts of the computation, and therefore it is not possible to predict theoretically the aggregated expected speed-up with an easy formula in terms of some dimensions or group orders. The paper finishes with Section 7, in which we give some runtime examples of our algorithm to illustrate its efficiency in practice.

2. *Notation*

In general, to compose maps from right to left, we write $\alpha\circ\beta$ to mean the map which applies first β and then α. Let V and W be vector spaces over the same field F. For given bases $\mathcal{B}$ of V and $\mathcal{C}$ of W and a linear map $\phi: V\to W$, we write $\mathbf{M}_{\mathcal{B}}^{\mathcal{C}}(\phi)$ for the matrix describing the action of ϕ with respect to the two bases. We use row-convention, that is, the rows of $\mathbf{M}_{\mathcal{B}}^{\mathcal{C}}(\phi)$ contain the coefficients of the action of ϕ on the basis $\mathcal{B}$ with respect to the basis $\mathcal{C}$. For an endomorphism without basis change, that is, $\mathcal{B}=\mathcal{C}$, we simply write $\mathbf{M}_{\mathcal{B}}(\phi)$. Note that by this convention we have $\mathbf{M}_{\mathcal{B}}(\varphi\circ\phi)=\mathbf{M}_{\mathcal{B}}(\phi)\mathbf{M}_{\mathcal{B}}(\varphi)$, and the matrix is acting by right multiplication on its natural vector space.

Now let V and W be FG-modules for a finite group G. Interpreting the elements of FG as the endomorphisms that they induce on V and W, we shall also write $\mathbf{M}_{\mathcal{B}}(g)$ and $\mathbf{M}_{\mathcal{C}}(g)$ for the matrices which describe the action of g on V and on W, respectively.

For a subgroup $K\leqslant G$, we denote the restricted modules by $V{\downarrow_K}$ and $W{\downarrow_K}$. If $V{\downarrow_K}=\bigoplus_{i=1}^{s}S_i$ and $W{\downarrow_K}=\bigoplus_{j=1}^{t}T_j$ are direct sum decompositions, we often choose bases $\mathcal{B}$ and $\mathcal{C}$ for V and W, respectively, by concatenating bases $\mathcal{B}_i$ of the S_i and bases $\mathcal{C}_j$ of the T_j. In this case, we denote by $M_{\mathcal{B}_i}^{\mathcal{B}_j}(g)$ the submatrix of $M_{\mathcal{B}}^{\mathcal{B}}(g)$ with rows corresponding to the basis vectors in $\mathcal{B}_i$ and columns corresponding to the basis vectors in $\mathcal{B}_j$. Formally, this is the matrix $M_{\mathcal{B}_i}^{\mathcal{B}_j}(p_j^V\circ g\circ\iota_i^V)$ of $p_j^V\circ g\circ\iota_i^V$ with respect to the bases $\mathcal{B}_i$ and $\mathcal{B}_j$, where $\iota_i^V: S_i\to V$ is the inclusion map and $p_j^V: V\to S_j$ is the projection map given by the above direct sum decomposition. Similarly, for a linear map $\varphi: V\to W$, we denote by $M_{\mathcal{B}_i}^{\mathcal{C}_j}(\varphi)$ the submatrix of $M_{\mathcal{B}}^{\mathcal{C}}(\varphi)$ which is equal to the matrix $M_{\mathcal{B}_i}^{\mathcal{C}_j}(p_j^W\circ\varphi\circ\iota_i^V)$ of $p_j^W\circ\varphi\circ\iota_i^V$, where $p_j^W: W\to T_j$ is the projection map given by the above direct sum decomposition.

3. *The theory*

To compute the action of $e_\lambda g e_\lambda$ on $\mathrm{Hom}_F(V,W)e_\lambda$, we will have to apply g to a basis of the condensed module and project the resulting images, which will in general spread out through the entire space $\mathrm{Hom}_F(V,W)$, onto $\mathrm{Hom}_F(V,W)e_\lambda$ by an application of e_λ.

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

Owing to the limited time and space resources available, this approach gives rise to two problems which are critical to any practical implementation. Firstly, when applying g, a straightforward implementation working in the potentially huge $\dim_F(V) \times \dim_F(W)$-dimensional space would confine the applicability of this method to only pocket-size examples. Secondly, when projecting with e_λ, we must equally avoid constructing the idempotent in the huge space.

The solution to both problems lies within the decomposition of $\mathrm{Hom}_F(V, W)$ into an internal direct sum of suitable FK-submodules. We give this key idea in the following straightforward theorem, which we state without its (obvious) proof.

THEOREM 3.1. *Let $V{\downarrow_K} = \bigoplus_{i=1}^{s} S_i$ and $W{\downarrow_K} = \bigoplus_{j=1}^{t} T_j$ be decompositions into simple FK-submodules S_i and T_j with projection maps $p_i^V : V \to S_i$ and $p_j^W : W \to T_j$ and inclusion maps $\iota_i^V : S_i \to V$ and $\iota_j^W : T_j \to W$. This implies the decomposition*

$$\mathrm{Hom}_F(V{\downarrow_K}, W{\downarrow_K}) = \bigoplus_{i=1}^{s} \bigoplus_{j=1}^{t} \mathcal{H}_{i,j} \tag{2}$$

as an internal direct sum of FK-submodules, where $\mathcal{H}_{i,j} = \iota_j^W \circ \mathrm{Hom}_F(S_i, T_j) \circ p_i^V$. Note that $\mathcal{H}_{i,j}$ is an FK-submodule of $\mathrm{Hom}_F(V{\downarrow_K}, W{\downarrow_K})$ since p_j^W and ι_i^V are FK-homomorphisms. Thus we have

$$\mathrm{Hom}_F(V{\downarrow_K}, W{\downarrow_K})e_\lambda = \bigoplus_{i=1}^{s} \bigoplus_{j=1}^{t} \mathcal{H}_{i,j}e_\lambda \tag{3}$$

as an internal direct sum of F-subspaces. Therefore, a basis of the whole space $\mathrm{Hom}_F(V, W)e_\lambda$ may be obtained by concatenating bases of the spaces $\mathcal{H}_{i,j}e_\lambda$ for all $1 \leqslant i \leqslant s$ and $1 \leqslant j \leqslant t$.

For the rest of the paper, we will fix a sequence of simple FK-submodules $S_1, \ldots, S_s$ of V and simple FK-submodules $T_1, \ldots, T_t$ of W such that $V{\downarrow_K} = \bigoplus_{i=1}^{s} S_i$ and $W{\downarrow_K} = \bigoplus_{j=1}^{t} T_j$, together with the projection and inclusion maps.

By Theorem 3.1, condensation preserves the direct sum decomposition (2), hence Lemma 4.2 solves the first problem of applying g without constructing its matrix on the whole space $\mathrm{Hom}_F(V, W)$.

Therefore we are now left with the second problem, namely that of finding an efficient way to describe the linear map on $\mathrm{Hom}_F(V, W)$ induced by e_λ. As we will see, this is closely related to finding a 'nice' basis for $\mathrm{Hom}_F(V, W)$, a problem we will deal with next.

Following the strategy of Theorem 3.1, we aim to construct a basis for the condensed space $\mathrm{Hom}_F(V, W)e_\lambda$ by concatenating bases for the direct summands of (3). Hence, we will take a closer look at condensing homomorphisms between simple FK-modules. The starting point of what follows is Lemma 3.2, which says that if we condense any space of F-linear maps between two modules with a linear idempotent, we may identify the resulting space with a space of FK-homomorphisms. This will be very useful later.

LEMMA 3.2. *Let V be an FK-module. Denote by V^λ the FK-module with the same underlying F-space structure and the (twisted) K-action given by $v * k := \lambda(k)vk$. Then, noting that $\mathrm{Hom}_F(V, W) = \mathrm{Hom}_F(V^\lambda, W)$, we obtain that the F-linear map on $\mathrm{Hom}_F(V, W)$ induced by e_1 and the F-linear map induced by e_λ on $\mathrm{Hom}_F(V, W)$ $(= \mathrm{Hom}_F(V^\lambda, W))$ are identical.*

Proof. Let $\varphi \in \mathrm{Hom}_F(V^\lambda, W) = \mathrm{Hom}_F(V, W)$. We have

$$\varphi e_1(v) = \sum_{k \in K} \varphi(v * k^{-1})k = \sum_{k \in K} \lambda(k^{-1})\varphi(vk^{-1})k = \varphi e_\lambda(v).$$

Hence e_1 and e_λ induce the same F-linear map on $\mathrm{Hom}_F(V, W)$. □

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

REMARK 3.3. Considering V^λ and e_1 instead of V and e_λ in Lemma 3.2 amounts to a basis change for the algebra FK: the idempotent e_λ is none other than the trace idempotent for the group $\{\lambda(k^{-1})k \mid k \in K \leqslant FK \leqslant FG\} \cong K$, also contained in FK. Hence the transition from e_1 to e_λ is a basis change in FK from K to the isomorphic group.

The consequences of Lemma 3.2 are far-reaching: when deriving a basis of $\mathrm{Hom}_F(V, W)e_\lambda$ that allows an efficient computational treatment, we can focus on the case where λ is the trivial character of K, by replacing V with its twist V^λ.

For the remainder of this section, we shall therefore assume without loss of generality that Λ is the trivial K-module and write e for the idempotent e_1.

As a first consequence, we can deduce with the help of Schur's lemma that homomorphisms between simple FK-modules often condense to zero.

COROLLARY 3.4. *Let S and T be simple FK-modules. Then we have*

$$\mathrm{Hom}_F(S, T)e = \begin{cases} 0 & \text{if } S \ncong_{FK} T, \\ \alpha \circ \mathrm{End}_{FK}(S) & \text{if } \alpha \text{ is an } FK\text{-isomorphism in } \mathrm{Hom}_{FK}(S, T). \end{cases}$$

Note that $E := \mathrm{End}_{FK}(S)$ is a field and, as such, is isomorphic to the splitting field of S and T.

Proof. This follows from the fact that $\mathrm{Hom}_{FK}(S, T) = \mathrm{Hom}_F(S, T)e$, Schur's lemma and Wedderburn's theorem on finite division rings. □

The idea now is to consider how to use the isomorphism of Corollary 3.4 to our advantage. Therefore, let us fix two isomorphic simple FK-summands $S := S_i \leqslant V{\downarrow_K}$ and $T := T_j \leqslant W{\downarrow_K}$, and let us keep the notation of Corollary 3.4. Thus E denotes the splitting field of S and T. We set $n := [E : F]$ to be the degree of the extension and θ to be a primitive element of E over F. Furthermore, set $d := \dim_F S$. We now give an alternative description of the projection induced by e on $\mathrm{Hom}_F(S, T)$. We will show that this description yields a computationally more efficient method for computing the action of e on $\mathrm{Hom}_F(S, T)$.

DEFINITION 3.5. Let $\alpha \in \mathrm{Hom}_{FK}(S, T)$ be an isomorphism. We define a non-degenerate symmetric bilinear form $\langle \cdot\,, \cdot \rangle$ on $\mathrm{Hom}_F(S, T)$ by setting

$$\langle \varphi, \psi \rangle := \mathrm{trace}(\alpha^{-1} \circ \varphi \circ \alpha^{-1} \circ \psi)$$

for any two $\varphi, \psi \in \mathrm{Hom}_F(S, T)$.

LEMMA 3.6. *We use the same notation as in Definition 3.5. The bilinear form $\langle \cdot\,, \cdot \rangle$ is K-invariant; that is,*

$$\langle \varphi k, \psi k \rangle = \langle \varphi, \psi \rangle$$

for all $\varphi, \psi \in \mathrm{Hom}_F(S, T)$ and $\langle \varphi e, \psi \rangle = \langle \varphi, \psi \rangle$ if $\psi \in \mathrm{Hom}_{FK}(S, T)$.

Proof. Denoting the linear map induced by the action of k on S by k_S and the linear map induced by the action of k on T by k_T, we have $\varphi k = k_T \circ \varphi \circ k_S^{-1}$. Now, as $k_T \circ \alpha = \alpha \circ k_S$, the equality

$$\alpha^{-1} \circ \varphi k \circ \alpha^{-1} \circ \psi k = k_S \circ \alpha^{-1} \circ \varphi \circ \alpha^{-1} \circ \psi \circ k_S^{-1}$$

is immediate, and hence the first claim follows. The second statement follows similarly by noting that $k_T^{-1} \circ \psi = \psi \circ k_S^{-1}$ if $\psi \in \mathrm{Hom}_{FK}(S, T)$. □

LEMMA 3.7. *The complement of $\mathrm{Hom}_{FK}(S, T)$ in $\mathrm{Hom}_F(S, T)$ with respect to the projection with e, namely the subspace $\mathrm{Hom}_F(S, T)(1 - e)$, is given by $\mathrm{Hom}_{FK}(S, T)^\perp$, the orthogonal*

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

complement with respect to the bilinear form of Definition 3.5. Hence the linear map induced by e on $\mathrm{Hom}_F(S,T)$ is given by the orthogonal projection of $\mathrm{Hom}_F(S,T)$ onto $\mathrm{Hom}_{FK}(S,T)$. Moreover, the restriction of the form $\langle\cdot\,,\cdot\rangle$ to $\mathrm{Hom}_{FK}(S,T)$ is also non-degenerate.

Proof. By Lemma 3.6 we have that $\mathrm{Hom}_F(S,T)(1-e)$ is contained in the orthogonal complement of $\mathrm{Hom}_{FK}(S,T)$ in $\mathrm{Hom}_F(S,T)$. Now, since $\langle\cdot\,,\cdot\rangle$ is non-degenerate, we have $\dim_F \mathrm{Hom}_F(S,T) = \dim_F \mathrm{Hom}_{FK}(S,T) + \dim_F \mathrm{Hom}_{FK}(S,T)^\perp$. Therefore we conclude that $\mathrm{Hom}_F(S,T)(1-e)$ is equal to $\mathrm{Hom}_{FK}(S,T)^\perp$ as claimed. □

The following lemma is relevant from a computational point of view.

LEMMA 3.8. *Let $(b_1,\ldots,b_n)$ be an F-basis of $\mathrm{Hom}_{FK}(S,T)$ and set*

$$B := (\langle b_i, b_j\rangle)_{1\leqslant i,j\leqslant n},$$

that is, B is the invertible (Gram) matrix of the restriction of the bilinear form to $\mathrm{Hom}_{FK}(S,T)$ with respect to the basis $(b_1,\ldots,b_n)$. Furthermore, for an arbitrary $\varphi\in\mathrm{Hom}_F(S,T)$, define

$$\kappa(\varphi) := [\langle\varphi,b_1\rangle,\ldots,\langle\varphi,b_n\rangle]\cdot B^{-1}.$$

Then the map

$$\pi : \varphi \mapsto \sum_{k=1}^{n} \kappa(\varphi)_k b_k$$

gives the projection of $\mathrm{Hom}_F(S,T)$ onto $\mathrm{Hom}_{FK}(S,T)$ induced by e; that is, we have $\pi(\varphi)=\varphi e$.

Proof. The claim can be checked by a straightforward computation. □

4. *The practice*

The aim of this section is to provide the means of addressing the problems stated at the beginning of Section 3 computationally. To this end, we give details on the steps necessary to realise the approach outlined theoretically in the previous section for practical computations.

DEFINITION 4.1. For an FK-module V with a decomposition $V = S_1\oplus S_2\oplus\ldots\oplus S_s$ into an internal direct sum of simple FK-modules as given above, we choose an F-basis of V as the concatenation of bases $\mathcal{B}_i$ for the simple direct summands S_i in such a way that for isomorphic summands S_i and S_j we have $\mathbf{M}_{\mathcal{B}_i}(k) = \mathbf{M}_{\mathcal{B}_j}(k)$ for all $k\in K$. Such a basis $\mathcal{B}$ is called *FK-symmetry adapted* (or *symmetry adapted* if K is evident). If $\mathcal{B}$ is FK-symmetry adapted and $\mathcal{C}$ is an FK-symmetry adapted basis of an FK-module W (with respect to a decomposition $W = T_1\oplus T_2\oplus\ldots\oplus T_t$ into a direct sum of simple FK-submodules), then we say that $\mathcal{B}$ and $\mathcal{C}$ are *synchronised* or *in synchronicity* if $\mathbf{M}_{\mathcal{B}_i}(k) = \mathbf{M}_{\mathcal{C}_j}(k)$ for all $k\in K$, given that S_i and T_j are isomorphic FK-modules. Note that in the latter case the F-linear map $\alpha\in\mathrm{Hom}_F(S_i,T_j)$ mapping $\mathcal{B}_i$ to $\mathcal{C}_j$ is actually an FK-isomorphism, and we will use it for the definition of the bilinear form $\langle\cdot\,,\cdot\rangle$ defined on $\mathrm{Hom}_F(S_i,T_j) = \alpha\circ\mathrm{End}_F(S_i)$.

The power of Theorem 3.1 may now be illustrated by the following lemma. Owing to the direct sum decomposition (2), we may apply a group element g to any homomorphism by dealing successively with linear maps between the simple summands of $V{\downarrow_K}$ and $W{\downarrow_K}$, namely the spaces $\mathrm{Hom}_F(S_i,T_j)$, which we consider to be subspaces of $\mathrm{Hom}_F(V,W)$ via their canonical embeddings. In this way, we gain some independence from the dimensions of the FG-modules V and W.

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

LEMMA 4.2. *Let V and W be FG-modules with FK-symmetry adapted bases $\mathcal{B}$ and $\mathcal{C}$. Let S and S' (respectively, T and T') be members of the internal direct sum decomposition into simple FK-submodules of $V{\downarrow_K}$ (respectively, $W{\downarrow_K}$). Then, for a $\varphi \in \mathrm{Hom}_F(S,T)$, we have*

$$\mathbf{M}_{\mathcal{B}_{S'}}^{\mathcal{B}_S}(g^{-1}) \cdot \mathbf{M}_{\mathcal{B}_S}^{\mathcal{C}_T}(\varphi) \cdot \mathbf{M}_{\mathcal{C}_T}^{\mathcal{C}_{T'}}(g) = \mathbf{M}_{\mathcal{B}_{S'}}^{\mathcal{C}_{T'}}(\varphi \cdot g).$$

Proof. This is elementary. For the notation see Section 2. □

Putting together Lemmas 4.2 and 3.8, we arrive at the following theorem, with which we can overcome those problems stated at the beginning of Section 3.

THEOREM 4.3. *Let S, S', T and T' be simple FK-modules with isomorphisms $\alpha \in \mathrm{Hom}_{FK}(S,T)$ and $\alpha' \in \mathrm{Hom}_{FK}(S',T')$. Assume, furthermore, that the bases $\mathcal{B}_S$ and $\mathcal{C}_T$, as well as $\mathcal{B}_{S'}$ and $\mathcal{C}_{T'}$, are synchronised; that is, $\alpha(\mathcal{B}_S) = \mathcal{C}_T$ and $\alpha'(\mathcal{B}_{S'}) = \mathcal{C}_{T'}$. Take E to be the splitting field of S (and T) with primitive element θ and $n := [E:F]$. Similarly, let E' denote the splitting field of S' (and T') whose primitive element is θ' with $n' := [E':F]$.*

Then the set $\{\alpha \circ \theta^k \mid k = 1, \ldots, n\}$ is an F-basis of $\mathrm{Hom}_{FK}(S,T)$, the set $\{\alpha' \circ \theta'^l \mid l = 1, \ldots, n'\}$ is an F-basis of $\mathrm{Hom}_{FK}(S',T')$, and the image of the basis element $\alpha \circ \theta^k$ under ege for some $g \in G$ has the coefficient vector

$$v := [\langle (\alpha \circ \theta^k) \cdot ge, \alpha' \circ \theta'^0 \rangle, \ldots, \langle (\alpha \circ \theta^k) \cdot ge, \alpha' \circ \theta'^{n'-1} \rangle] \cdot B'^{-1}$$

with respect to the basis $(\alpha' \circ \theta'^l \mid 0 \leqslant l \leqslant n'-1)$, where

$$B' = (\langle \alpha' \circ \theta'^{i-1}, \alpha' \circ \theta'^{j-1} \rangle)_{1 \leqslant i,j \leqslant n'} = (\mathrm{trace}(\theta'^{i-1}\theta'^{j-1}))_{1 \leqslant i,j \leqslant n'}.$$

Thus, upon setting

$$M := \mathbf{M}_{\mathcal{B}_{S'}}^{\mathcal{B}_S}(g^{-1}) \cdot \mathbf{M}_{\mathcal{B}_S}^{\mathcal{C}_T}(\alpha \circ \theta^k) \cdot \mathbf{M}_{\mathcal{C}_T}^{\mathcal{C}_{T'}}(g) =: M = (m_{i,j}) \in F^{n' \times n'},$$

we obtain

$$v = [\mathrm{trace}(M \cdot \mathbf{M}_{\mathcal{B}_{S'}}^{\mathcal{C}_{T'}}(\alpha' \circ \theta'^0)), \ldots, \mathrm{trace}(M \cdot \mathbf{M}_{\mathcal{B}_{S'}}^{\mathcal{C}_{T'}}(\alpha' \circ \theta'^{n'-1}))] \cdot B'^{-1}.$$

Proof. By Lemma 3.2, the idempotent e fixes every element of E; in particular, it fixes a basis vector $\alpha \circ \theta^k$ for any $k \in \{0, \ldots, n-1\}$. Therefore we have $\alpha \circ \theta^k \cdot ege = \alpha \circ \theta^k \cdot ge$. By Lemma 4.2, the action of g on $\alpha \circ \theta^k$ is given by the matrix M with respect to the chosen bases. Now, by Lemma 3.2, the multiplication of $(\alpha \circ \theta^k)g$ by e is realized by the orthogonal projection of $(\alpha \circ \theta^k)g$ onto $\mathrm{Hom}_{FK}(S',T')$, that is, by applying Lemma 3.8. Also, note that

$$\langle \alpha' \circ \theta'^k, \alpha' \circ \theta'^l \rangle = \mathrm{trace}(\alpha'^{-1} \circ (\alpha' \circ \theta'^k) \circ \alpha'^{-1} \circ (\alpha' \circ \theta'^l)) = \mathrm{trace}(\theta'^k\theta'^l).$$ □

REMARK 4.4. In the above formulae, we have

$$\mathbf{M}_{\mathcal{B}_S}^{\mathcal{C}_T}(\alpha \circ \theta^k) = \mathbf{M}_{\mathcal{B}_S}^{\mathcal{B}_S}(\theta^k) \quad \text{and} \quad \mathbf{M}_{\mathcal{B}_{S'}}^{\mathcal{C}_{T'}}(\alpha' \circ \theta'^k) = \mathbf{M}_{\mathcal{B}_{S'}}^{\mathcal{B}_{S'}}(\theta^k).$$

Theorem 4.3 illustrates that a large portion of the computational effort in the calculation of representing matrices for elements of the condensed group algebra is devoted to matrix multiplications involving the primitive elements of the splitting fields. It is therefore desirable to choose special bases for the vector spaces involved in such a manner that the primitive elements are represented by matrices which are more amenable to a practical implementation.

DEFINITION 4.5. For an F-basis $\mathcal{B}_S$ of S, take $\{b_1, \ldots, b_{d/n}\} \subseteq \mathcal{B}_S$ to be a subset which is an E-basis (recall that $E = \mathrm{End}_{FK}(S)$ is the splitting field of S and $\theta \in E$ a primitive element). Using these elements, we define the sequence

$${}^{\theta}\mathcal{B}_S := (b_1, b_1\theta, \ldots, b_1\theta^{n-1}, b_2, b_2\theta, \ldots, b_2\theta^{n-1}, \ldots, b_{d/n}, b_{d/n}\theta, \ldots, b_{d/n}\theta^{n-1}),$$

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

which is again an F-basis of S; we call it *θ-adapted* or, more generally, *adapted to the splitting field* of S.

If $\mathcal{B}$ is a synchronised semi-simplicity basis, then to preserve synchronicity when concatenating the θ-adapted bases of Definition 4.5, we construct them in the following way.

REMARK 4.6. Let S' be a simple FK-module isomorphic to S. If $\mathcal{B}_S$ and $\mathcal{B}_{S'}$ are synchronised and $\{i_1, \dots, i_{d/n}\} \subseteq \{1, \dots, d\}$ are the indices of the elements of $\mathcal{B}_S$ chosen to obtain ${}^{\theta}\mathcal{B}_S$ as in Definition 4.5, then choosing the subsequence of elements with the same indices in $\mathcal{B}_{S'}$ yields a ${}^{\theta}\mathcal{B}_{S'}$ which is in synchronicity with ${}^{\theta}\mathcal{B}_S$.

Proof. By the synchronicity of $\mathcal{B}_S$ and $\mathcal{B}_{S'}$, every element of K acts the same way on both bases. Also, θ commutes with every element of K. Hence ${}^{\theta}\mathcal{B}_S$ and ${}^{\theta}\mathcal{B}_{S'}$ are synchronised. □

The bases as in Definition 4.5 facilitate an F-basis for $\mathrm{Hom}_F(S, T)$ which is particularly easy to work with.

LEMMA 4.7. *Let ${}^{\theta}\mathcal{B}_S$ and ${}^{\theta}\mathcal{C}_T$ be θ-adapted synchronised bases for S and T, and let α be the FK-isomorphism mapping ${}^{\theta}\mathcal{B}_T$ to ${}^{\theta}\mathcal{C}_S$. Then*

$$M_{{}^{\theta}\mathcal{B}_S}^{{}^{\theta}\mathcal{C}_T}(\alpha \circ \theta^k) = M_{{}^{\theta}\mathcal{B}_S}^{{}^{\theta}\mathcal{B}_S}(\theta^k) = \begin{bmatrix} C(\mu_\theta)^k & 0 & \dots & 0 \\ 0 & C(\mu_\theta)^k & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & C(\mu_\theta)^k \end{bmatrix}$$

where $C(\mu_\theta)$ denotes the companion matrix

$$C(\mu_\theta) = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 0 & 1 \\ a_0 & a_1 & \dots & a_{n-2} & a_{n-1} \end{bmatrix} \in F^{n \times n}$$

of the minimal polynomial $\mu_\theta = X^n - \sum_{i=0}^{n-1} a_i X^i$ of θ.

Proof. By our chosen bases, $\mathbf{M}_{{}^{\theta}\mathcal{B}_S}^{{}^{\theta}\mathcal{B}_S}$ maps the primitive element θ to the block-diagonal matrix having $C(\mu_\theta)$ along the diagonal, and $\mathbf{M}_{{}^{\theta}\mathcal{B}_S}^{{}^{\theta}\mathcal{C}_T}(\alpha)$ is the identity matrix. □

Thus, by choosing synchronised bases which are adapted to splitting fields, we may assume a block-diagonal matrix as in Lemma 4.7 in the formula of Theorem 4.3, instead of an arbitrary representing matrix for $\alpha \circ \theta$. In this way, we are able to exploit the special form of a companion matrix, ultimately avoiding straightforward but costly matrix multiplications wherever possible. The details of implementing this approach in our algorithm are given in Section 6.

5. *Precondensation*

The action of an element $e_\lambda g e_\lambda$ for some $g \in G$ on $\mathrm{Hom}_F(V, W)e_\lambda$ is the same as the action of ge_λ on $\mathrm{Hom}_F(V, W)e_\lambda$. Hence, as we can identify $\mathrm{Hom}_F(V, W)e_\lambda$ with $\mathrm{Hom}_F(V^\lambda, W)e_1$ by means of Lemma 3.2, we may equivalently consider the action of g on the special basis of $\mathrm{Hom}_F(V^\lambda, W)e_1$ constructed in Section 3. This also allows us to project the images of these

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

basis vectors under g back onto the fixed space $\mathrm{Hom}_F(V^\lambda, W)e_1$ without explicitly applying an idempotent.

Therefore, for the computation of a representation of $e_\lambda g e_\lambda$ on the module $\mathrm{Hom}_F(V, W)e_\lambda$, a non-trivial FK-module Λ is relevant only at the very beginning of the computation: as the special basis of the condensed homomorphism space relies only on the action of K on $\mathrm{Hom}_F(V, W)$, replacing $V{\downarrow_K}$ by $V^\lambda{\downarrow_K}$ and e_λ by $e := e_1$ lets us determine the action of ege on $\mathrm{Hom}_F(V, W)e$.

To this end, we readily identify two basic steps from Section 3 to form the framework of an algorithmic implementation.

Step 1. Determine the composition factors $\{S_1, \ldots, S_s\}$ and $\{T_1, \ldots, T_t\}$ of $V{\downarrow_K}$ and $W{\downarrow_K}$, respectively, along with their splitting fields. Compute the mutually synchronised K-semi-simplicity bases for both V and W, which are adapted to the splitting fields of the composition factors, and determine the matrix B as in Lemma 3.8.

Step 2. For all pairs of composition factors (S, T) from Step 1 for which $S \cong_{FK} T$, compute the corresponding part of the result matrix by Theorem 4.3.

Obviously, while Step 2 needs to be repeated every time, the output of Step 1 only has to be computed once, if we wish to compute representing matrices for several different algebra elements $eg_1e, \ldots, eg_ke \in eFGe$. Because in Step 2 the actual output matrix is produced, we call this step the *condensation step*. The one-off preparatory calculations of Step 1 are summed up under the name *precondensation*.

Our theoretical development in Section 3 already illustrates that a practical implementation of Theorem 4.3 relies most importantly on the underlying bases of the subspaces V and W. As we will show, the chosen approach, namely the use of synchronised and adapted bases, not only allows a nice description of the algorithm but also forms the backbone of our efficiency considerations.

Of course, the task of computing the composition factors of $V{\downarrow_K}$ and $W{\downarrow_K}$, as well as their splitting fields, is accomplished by a run of the MeatAxe [**9**, **12**]. But as a key element of precondensation is calculation of the synchronised semi-simplicity bases, the output of current MeatAxe implementations in GAP or as the C-MeatAxe 2.4 [**10**] turns out to be insufficient for our purposes. For this task, we employ our own GAP implementation of the MeatAxe, which will be made available in the form of the GAP package chop. It covers the basic functionality of the C-MeatAxe but also features an augmented decomposition algorithm `Chop`, which lets us compute the necessary bases easily.

Definition 5.1. Let V be a finite-dimensional module for some finite-dimensional F-algebra A, and let $0 = V_0 \lneqq V_1 \lneqq \ldots \lneqq V_l = V$ be a composition series of V. Then an F-basis $\mathcal{B}$ of V is said to be *adapted to the composition series* if the matrix representation of every $a \in A$ on V with respect to $\mathcal{B}$ is a block lower triangular matrix whose diagonal block B_i gives the matrix representation of a on the composition factor V_i/V_{i-1} for all $i = 1, \ldots, l$.

In the special case of determining a composition series of a semi-simple module, the adapted basis is the foundation on which we build a semi-simplicity basis. The basic idea is as follows. Since we have an explicit basis of the semi-simple module, it is easy to define an F-projection π onto a quotient by a submodule of the composition series. We transform π into an FK-module endomorphism by applying the trace map $\mathrm{Tr} : \mathrm{Hom}_F(V, W) \to \mathrm{Hom}_{FK}(V, W)$, that is, by defining

$$\mathrm{Tr}(\pi)(v) := \frac{1}{|K|} \sum_{k \in K} \pi(vk^{-1})k = \pi e.$$

Successively applying these endomorphisms to the composition series adapted basis will then yield a semi-simplicity basis.

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

LEMMA 5.2. *Let $\mathcal{B}$ be a basis of $V{\downarrow_K}$ which is adapted to a composition series, and let S be a submodule in this composition series and call the associated quotient Q. Then we may partition $\mathcal{B} = \mathcal{B}_S \sqcup \mathcal{B}_Q$ into a submodule and quotient part. By the definition of $\mathcal{B}$ we have*

$$\mathbf{M}_{\mathcal{B}}(a) = \begin{bmatrix} \mathbf{M}_{\mathcal{B}_S}(a) & 0 \\ * & \mathbf{M}_{\mathcal{B}_Q}(a) \end{bmatrix}$$

for every $a \in FK$. Let $\pi \in \mathrm{End}_F(V)$ be the projection onto S in the vector space decomposition $V = \langle \mathcal{B}_S \rangle \oplus \langle \mathcal{B}_Q \rangle$ induced by the partition of $\mathcal{B}$. Setting $\mathcal{B}'_Q := (\mathrm{id} - \mathrm{Tr}(\pi))(\mathcal{B}_Q)$ then gives a basis $\mathcal{B}_S \sqcup \mathcal{B}'_Q$ of $V{\downarrow_K}$ which yields

$$\mathbf{M}_{\mathcal{B}_S \sqcup \mathcal{B}'_Q}(a) = \begin{bmatrix} \mathbf{M}_{\mathcal{B}_S}(a) & 0 \\ 0 & \mathbf{M}_{\mathcal{B}_Q}(a) \end{bmatrix}$$

for every $a \in FK$. Note that the matrix representation on the quotient is preserved.

Proof. Since $\mathrm{id} - \pi$ is the projection onto a vector space complement of S, the FK-homomorphism $\mathrm{id} - \mathrm{Tr}(\pi)$ projects onto a K-invariant complement of S. The matrix representation on the quotient is maintained because for all $v \in \mathcal{B}_Q$ and $k \in K$ the equation $(\mathrm{id} - \mathrm{Tr}(\pi))(v)k = vk - \mathrm{Tr}(\pi)(vk) = (\mathrm{id} - \mathrm{Tr}(\pi))(vk - \pi(vk))$ holds. □

To quickly compute a semi-simplicity basis from a basis which is adapted to a composition series, we exploit the inherently recursive nature of this problem: once we obtain the direct sum of a submodule and a quotient by applying Lemma 5.2, we may restrict all further computations to either the submodule or the quotient. A subsequent iteration of this procedure benefits greatly from the decreasing sizes of the matrices involved. In order to maximise this speed-up, we aim to split the currently considered module into a submodule and a quotient of approximately the same size.

Algorithm 1 SemiSimplicityBasis

Input: semi-simple module V with basis $\mathcal{B}$ adapted to a composition series C.
Output: $\mathcal{B}$ is a semi-simplicity basis for V.

Choose $S \leqslant V$ in C such that $\dim_F S$ is close to $\frac{1}{2} \dim_F V$.
Extend a basis $\mathcal{B}_S$ for S to a basis $\mathcal{B} = \mathcal{B}_S \sqcup \mathcal{B}_Q$ of V (see Lemma 5.2), thus defining an F-projection $\pi : V \twoheadrightarrow S$.
Compute $\mathrm{Tr}(\pi)$ with respect to the basis $\mathcal{B}$.
$\mathcal{B}_Q \leftarrow (\mathrm{id} - \mathrm{Tr}(\pi))(\mathcal{B}_Q)$ {Lemma 5.2}
if S is reducible **then**
  $\mathcal{B}_S \leftarrow$ SemiSimplicityBasis(S, $\mathcal{B}_S$)
end if
if Q is reducible **then**
  $\mathcal{B}_Q \leftarrow$ SemiSimplicityBasis(Q, $\mathcal{B}_Q$)
end if

LEMMA 5.3. *Let $K' \leqslant K$ be a subgroup and denote by $K'\backslash K$ a right transversal of K' in K. Let V be some FK-module with basis $\mathcal{B}$, and choose some $v \in V$. Then we have*

$$\mathbf{M}_{\mathcal{B}}(\mathrm{Tr}(\pi)) = \frac{1}{|K|} \sum_{k \in K'\backslash K} \mathbf{M}_{\mathcal{B}}(k^{-1}) \left(\sum_{k' \in K'} \mathbf{M}_{\mathcal{B}}(k'^{-1}) \mathbf{M}_{\mathcal{B}}(\pi) \mathbf{M}_{\mathcal{B}}(k') \right) \mathbf{M}_{\mathcal{B}}(k).$$

Proof. As we may write every element $x \in K$ uniquely as $x = k'k$ for a $k' \in K'$ and a $k \in K'\backslash K$, and since $\mathbf{M}_{\mathcal{B}}(k'k) = \mathbf{M}_{\mathcal{B}}(k')\mathbf{M}_{\mathcal{B}}(k)$, the claim follows. □

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

To use Lemma 5.3 to its full extent, we have to apply it several times: after choosing a subgroup chain $\{1\} = K_0 \lneqq K_1 \lneqq \ldots \lneqq K_l = K$ for K, we may iterate Lemma 5.3, and therefore we only need to compute the transversal elements in $K_{i-1}\backslash K_i$ for $i = 1, \ldots, l$. Thus, instead of computing $\mathrm{Tr}(\pi)$ and needing $2|K|$ matrix multiplications, we now only need $2\sum_{i=1}^{l}[K_i : K_{i-1}]$. For example, choosing K to be an ℓ-group for some prime ℓ different from p with $|K| = \ell^m$, there exists a composition series of K of length m whose composition factors are all cyclic of order ℓ. Thus Lemma 5.3 allows us to compute the projection with only $2m\ell$ matrix multiplications, in contrast to the $2\ell^m$ that a straightforward implementation would take.

Being able to quickly calculate semi-simplicity bases, we now turn to the second open problem, that of synchronising bases and adapting the basis of a composition factor to a primitive element of its splitting field (see Definition 4.5). The nature of these two tasks allows a simultaneous treatment of both.

To compare two simple modules, that is, to test whether they are isomorphic or not, the MeatAxe uses Parker's standard basis technique (see [**9**]). Of course, in compliance with Definition 4.1, standard bases may be used to achieve synchronicity, because with respect to a standard basis every isomorphic composition factor affords the same matrix representation. In general, however, a standard basis of a composition factor does not need to be adapted to a primitive element of its splitting field. Therefore we have to do a little more work here.

During the computation of the module's composition factors, for every isomorphism type of an FK-module S which occurs in the composition series, the degree of its splitting field is determined. This is done by the method introduced by Holt and Rees in [**2**, Section 3].

In order to produce θ-adapted synchronised bases for two isomorphic modules, we transform both to standard basis first. Then, by Remark 4.6, the above procedure yields the desired result. In particular, the basis change required only needs to be calculated once, and can then be applied to any isomorphic module in standard basis.

Therefore, we may incorporate the computation of a module's synchronised basis which is adapted to the primitive elements of the splitting fields of its composition factors into the precondensation algorithm as follows.

While chopping a module into its composition factors, the MeatAxe compares every composition factor found with every element in the database of isomorphism types of composition factors already found. In particular, it determines the degree of the splitting field. If the composition factor is isomorphic to an already known one in the database, it is transformed into the corresponding standard basis. Therefore, in light of Theorem 4.3, to ensure that the bases for the two restricted modules $V{\downarrow_K}$ and $W{\downarrow_K}$ are mutually synchronised (that is, any two isomorphic composition factors afford the same matrix representation, irrespective of the module in which they occur), we allow as additional input into the MeatAxe (the program `Chop`, to be precise) a database of simple modules in standard basis. Then an execution of `Chop` will automatically produce bases which are synchronised properly.

In the precondensation step, we now only need to adapt the basis of every module in the database to its splitting field by employing the method of Holt and Rees. The resulting basis change matrix is then applied to all subbases of the whole module's basis which correspond to composition factors isomorphic to the database module.

If the field F is already a splitting field for a composition factor, then we do not need to compute a basis which is adapted to this splitting field, of course; a standard basis is sufficient in this case. Also note that the modules in the database will always be (only) in standard basis form.

Summing up, the following are the preparatory computations constituting the necessary calculations which provide the bases for Theorem 4.3 to be applied.

(i) If Λ is non-trivial, then replace V by V^{λ} and e_{λ} by $e := e_1$.

(ii) Determine the composition factors of $V{\downarrow_K}$ and compute a basis of $V{\downarrow_K}$ which is adapted to the composition series found.

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

(iii) Using the database produced in the previous step, find the composition factors of $W{\downarrow_K}$.

(iv) As in [2], find primitive elements for the splitting fields of the composition factors (that is, their endomorphism rings), and adapt their bases to these elements.

(v) Convert the adapted bases of both $V{\downarrow_K}$ and $W{\downarrow_K}$ to semi-simplicity using Lemmas 5.2 and 5.3.

The final ingredient needed for the application of Theorem 4.3 in the condensation step is knowledge of the Gram matrix B of Lemma 3.8 for every pair (S, T) of composition factors for which $S \cong T$.

In other words, given the primitive element θ of $E := \mathrm{End}_{FK}(S)$, where E is an extension of the ground field F of degree n, we need to determine $B = (\langle \alpha \circ \theta^{i-1}, \alpha \circ \theta^{j-1} \rangle)_{1 \leqslant i,j \leqslant n}$. As we choose θ-adapted synchronised bases for S and T, Definition 3.5 gives

$$\langle \alpha \circ \theta^i, \alpha \circ \theta^j \rangle = \mathrm{trace}({}^{\theta}\mathbf{M}_{\mathcal{B}_S}^{\mathcal{B}_S}(\theta^{i+j})) = \frac{d}{n}\mathrm{trace}(C(\mu_\theta)^{i+j})$$

by using Lemma 4.7, where $C(\mu_\theta)$ is again the companion matrix of the minimal polynomial of θ. Note that $d/n \neq 0$ in F, as S is an absolutely irreducible d/n-dimensional EK-module. Therefore this information is easily determined after calculating a primitive element as outlined above, if we store its minimal polynomial. The inverse of B is recorded as part of its corresponding module in the database.

We close this section by briefly discussing a comparison with the precondensation step in the C-MeatAxe rendition of tensor condensation as described in [5]. In contrast to the peak word method used there, which is highly probabilistic and most likely cannot be analysed rigorously, we use a completely deterministic approach here. Both methods have their merits and disadvantages. Peak words are sometimes difficult to find at all; this happened frequently in cases where our methods succeeded easily. On the other hand, as soon as the orders of some composition factors of the condensation subgroup are divisible by some huge prime, our method is essentially doomed, even though it might still be possible to find peak words. Therefore, it is essentially meaningless to compare the performance of the two methods, which is why we refrain from doing so in this paper. Users have to be aware of both methods and choose the one which is more appropriate for the problem at hand.

6. *Condensation*

After completing the necessary precondensation calculations, we may start the actual condensation of a group element $g \in G$. As we have already seen in Section 1, the group G acts on $\mathrm{Hom}_F(V, W)$ by taking a linear map φ to the homomorphism mapping any $v \in V$ to $\varphi(vg^{-1})g$. Therefore the input to our condensation algorithm consists of two matrices giving, respectively, the action of g^{-1} on V and g on W. Each is, of course, written with respect to synchronised semi-simplicity bases $\mathcal{B}$ and $\mathcal{C}$ which are adapted to the splitting fields of the composition factors. From this we calculate a matrix for the action of the condensed element ege on the condensed homomorphism space $\mathrm{Hom}_F(V, W)e$. The output matrix is constructed by multiple applications of Theorem 4.3. With each call, the matrix product calculated involves submatrices of both $\mathbf{M}_{\mathcal{B}}(g^{-1})$ and $\mathbf{M}_{\mathcal{C}}(g)$.

Thus, as we are mostly dealing with submatrices of larger matrices, the introduction of the following notation is convenient. Let $A \in F^{m \times n}$ be a matrix, and let $r \in \{1, \ldots, m\}$ and $s \in \{1, \ldots, n\}$. For two strictly increasing sequences of integers $1 \leqslant i_1 < i_2 < \ldots < i_r \leqslant m$ and $1 \leqslant j_1 < j_2 < \ldots < j_s \leqslant n$, we set $A_{[i_1,\ldots,i_r]}^{[j_1,\ldots,j_s]}$ to be the $r \times s$ submatrix $(a_{i_k,j_l})_{1 \leqslant k \leqslant r, 1 \leqslant l \leqslant s}$ of A. If $[i_1, \ldots, i_r] = [1, \ldots, m]$ or $[j_1, \ldots, j_s] = [1, \ldots, n]$, then we omit the respective range.

Considering the projection onto $\mathrm{Hom}_F(V, W)e$ of Lemma 3.8, we see that the result computed in Theorem 4.3 does not require knowledge of all entries of the matrix product M. Since, in particular, with respect to the specially constructed bases the matrix giving the

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

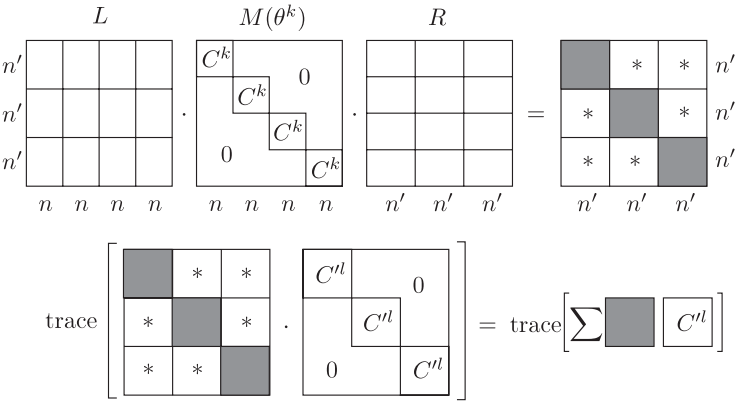

FIGURE 1. *Illustration of the proof of Theorem 6.1.*

primitive element is of the simple block-diagonal form of Lemma 4.7, we only need the very same diagonal blocks of M. Therefore we can reformulate Theorem 4.3 in a more implementation-friendly version, ultimately avoiding unnecessary calculations. Note that for the complexity analysis in the proof we rely on some lemmas presented after the theorem.

THEOREM 6.1. *We use the notation of Theorem 4.3. For brevity we define* $L := \mathbf{M}_{\theta'\mathcal{B}_{S'}}^{\theta\mathcal{B}_S}(g^{-1})$ *and* $R := \mathbf{M}_{\theta\mathcal{C}_T}^{\theta'\mathcal{C}_{T'}}(g)$. *Also, let us denote the companion matrix of* μ_θ *by* $C(\mu_\theta)$. *Then the coefficient vector* vB' *of Theorem 4.3 can be calculated by evaluating the* $n' \times n'$ *matrix*

$$N := \sum_{i=1}^{d'/n'} \sum_{j=1}^{d/n} L_{[(i-1)n'+1,\ldots,in']}^{[(j-1)n+1,\ldots,jn]} \cdot C(\mu_\theta)^k \cdot R_{[(j-1)n+1,\ldots,jn]}^{[(i-1)n'+1,\ldots,in']}$$

and computing $\operatorname{trace}(N \cdot C(\mu_{\theta'})^{l-1})$ *for* $l = 1, 2, \ldots, n'$. *The resulting vector* vB' *then has to be multiplied from the right by* B'^{-1} *to get* v.

Evaluating all this for $k = 0, 1, \ldots, n-1$ *requires, in total, at most*

$$2dd'(n + n' - 1) + n'(2n'^2 + n' - 2)$$

elementary operations in the field F. *Here we count both multiplications and additions as elementary field operations.*

Proof. By the θ-adaptedness of our bases, we can cut L into $n' \times n$ blocks and R into $n \times n'$ blocks as illustrated in Figure 1 (there we have $d/n = 4$ and $d'/n' = 3$).

The first idea for avoiding unnecessary computations is that to evaluate the traces of the big $d' \times d'$ matrices $L \cdot \mathbf{M}_{\mathcal{B}_S}^{\mathcal{B}_T}(\alpha \circ \theta^k) \cdot R \cdot \mathbf{M}_{\mathcal{B}_{S'}}^{\mathcal{B}_{T'}}(\alpha \circ \theta'^l)$, we only have to compute the grey diagonal $n' \times n'$ blocks of $L \cdot \mathbf{M}_{\mathcal{B}_S}^{\mathcal{B}_S}(\theta^k) \cdot R$, owing to the nice block-diagonal structure of $\mathbf{M}_{\mathcal{B}_{S'}}^{\mathcal{B}_{S'}}(\theta'^l)$ and Remark 4.4. Note that the blocks marked with a star in Figure 1 are not necessarily equal to 0, but computing them is not needed!

The second idea is to use the sparseness of the companion matrices that occur, together with caching of intermediate results.

Let ${}^kA := L \cdot \mathbf{M}_{\mathcal{B}_S}^{\mathcal{B}_S}(\theta^k) \cdot R$. To compute the ith grey block ${}^kA_{[(i-1)n'+1,\ldots,in']}^{[(i-1)n'+1,\ldots,in']}$ of ${}^kA$, we have to add the products

$$L_{[(i-1)n'+1,\ldots,in']}^{[(j-1)n+1,\ldots,jn]} \cdot C(\mu_\theta)^k \cdot R_{[(j-1)n+1,\ldots,jn]}^{[(i-1)n'+1,\ldots,in']}$$

for $j = 1, \ldots, n$. Because we need these products for all $k = 0, \ldots, n-1$, we can compute the products $L_{[(i-1)n'+1,\ldots,in']}^{[(j-1)n+1,\ldots,jn]} \cdot C(\mu_\theta)^k$ inductively by just multiplying some previously known

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

matrix by a companion matrix from the right. From the complexity results in Lemma 6.5, it follows that we need at most $d/n \cdot (n-1) \cdot 2n'n$ elementary field operations to compute all products $L^{[(j-1)n+1,\ldots,jn]}_{[(i-1)n'+1,\ldots,in']} \cdot C(\mu_\theta)^k$ for fixed i and all j, k. Given those products, evaluating the ith grey block ${}^kA^{[(i-1)n'+1,\ldots,in']}_{[(i-1)n'+1,\ldots,in']}$ of ${}^kA$ for all k needs at most $d/n \cdot n'^2(2n-1)$ elementary field operations for the matrix multiplications with the R-parts, plus another d/n additions of $n' \times n'$ matrices to the final result, requiring $d/n \cdot n'^2$ elementary field operations.

All these numbers of elementary field operations have to be multiplied by d'/n', as we have to compute all grey $n' \times n'$ blocks of all ${}^kA$. However, owing to the fact that all $n' \times n'$ diagonal blocks of $\mathbf{M}^{\mathcal{B}_{S'}}_{\mathcal{B}_{S'}}(\theta'^l)$ are equal, we do not have to process these blocks separately but only need to compute their sum. This is illustrated in the second row of Figure 1.

Summing up these numbers results in

$$\frac{d'}{n'}\left(\frac{d}{n}(n-1)\cdot 2n'n + \frac{d}{n}\cdot n'^2(2n-1) + \frac{d}{n}\cdot n'^2\right) = 2dd'\cdot(n+n-1'),$$

which is the first summand of the number of operations in the theorem.

It remains to evaluate the traces of $N \cdot C(\mu_{\theta'})^l$ for $l = 0, \ldots, n'-1$. Here we can use the same trick as above and compute these products by inductively multiplying previously computed matrices by the companion matrix $C(\mu_{\theta'})$ from the right. Thus, to compute all these matrices needs at most $(n'-1)\cdot 2n'^2$ elementary field operations, again by Lemma 6.5. To evaluate the traces then requires another $n'\cdot(n'-1)$ additions.

Since these traces form the vector vB' from Theorem 4.3, we still have to multiply this vector from the right by the stored $n' \times n'$ matrix B'^{-1}, which needs another $n'(2n'-1)$ elementary field operations.

Summing up these numbers results in

$$(n'-1)\cdot 2n'^2 + n'\cdot(n'-1) + n'(2n'-1) = n'(2n'^2+n'-2)$$

elementary field operations, which is the second summand of the number given in the theorem.

Note that in an actual implementation, some optimisation will be done if zeros are encountered, and thus our numbers are upper bounds. □

Under certain circumstances Theorem 6.1 allows a further simplification.

COROLLARY 6.2. *In the case where F is a splitting field for S and T as well as for S' and T', the corresponding coefficient vector of Theorem 4.3 is in fact only a scalar and is given by*

$$\frac{1}{d'}\sum_{i=1}^{d'} L_{[i]}\cdot R^{[i]};$$

that is, it is obtained by adding the standard scalar products of the vectors $L_{[i]}$ and $R^{[i]}$ for all i. In this case, it only takes $2dd'+1$ elementary field operations to compute the result.

Proof. We observe that if $n = 1$, then we have $C(\mu_\theta) = 1$ in Theorem 6.1, and therefore we may omit this matrix from the product. If n' is also equal to 1, then we only have to compute the diagonal of $L \cdot R$ in Theorem 6.1 and sum up all entries. The theorem directly specialises to the claim. Note that the additional 1 in the expression $2dd'+1$ of field operations is due to the division by d'. □

It is now easy to formulate the condensation algorithm. However, to prepare for this endeavour we need to release a barrage of notation. As we have detailed in Section 3, a basis for the condensed space $\mathrm{Hom}_F(V,W)e$ may be obtained by embedding and concatenating bases for the FK-homomorphism spaces $\mathrm{Hom}_{FK}(S,T)$ whenever $S \cong T$ for composition factors S and T. Thus, let $\{S_1, \ldots, S_r\}$ and $\{T_1, \ldots, T_r\}$ denote, respectively, a complete set of

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

isomorphism types of composition factors occurring simultaneously in $V{\downarrow_K}$ and $W{\downarrow_K}$ such that $S_i \cong T_i$ for $i = 1, \ldots, r$. Let s_i be the multiplicity of S_i in $V{\downarrow_K}$ and, analogously, let t_i be the multiplicity of T_i in $W{\downarrow_K}$. Denote the different direct summands of isomorphism type S_i occurring in $V{\downarrow_K}$ by $S_i^{(1)}, \ldots, S_i^{(s_i)}$ and those of isomorphism type T_i occurring in $W{\downarrow_K}$ by $T_i^{(1)}, \ldots, T_i^{(t_i)}$. Furthermore, we use the notation of Theorems 4.3 and 6.1; in other words, d_i gives the dimension of S_i and T_i, and n_i gives the degree of the corresponding splitting field. Let θ_i denote a primitive element of this splitting field, considered as an element of $\mathrm{End}_{FK}(S_i)$.

THEOREM 6.3. *Using the notation from above, we obtain the following.*

(i) *Algorithm 2 computes the representing matrix of ege on* $\mathrm{Hom}_F(V, W)e$ *with respect to a basis adapted to the decomposition in equation (3).*

(ii) *Execution of Algorithm 2 needs at most*

$$\sum_{l=1}^{r}\sum_{l'=1}^{r}\sum_{s=1}^{s_l}\sum_{s'=1}^{s_{l'}}\sum_{t=1}^{t_l}\sum_{t'=1}^{t_{l'}} 2d_l d_{l'}(n_l + n_{l'} - 1) + n_{l'}(2n_{l'} + n_{l'} - 2)$$
$$= \sum_{l=1}^{r}\sum_{l'=1}^{r} s_l s_{l'} t_l t_{l'} \left[2d_l d_{l'}(n_l + n_{l'} - 1) + n_{l'}(2n_{l'} + n_{l'} - 2)\right]$$

elementary field operations.

Algorithm 2 HomCond — Condensation Algorithm

Input: matrices $\mathbf{M}_{\mathcal{B}}(g^{-1})$ and $\mathbf{M}_{\mathcal{C}}(g)$ for some $g \in G$ with synchronised, splitting field adapted semi-simplicity bases $\mathcal{B}$ and $\mathcal{C}$.

Output: the matrix m gives the action of ege on $\mathrm{Hom}_F(V, W)e$.

$m \leftarrow 0 \in F^{D \times D}$ $\{D = \dim_F \mathrm{Hom}_F(V, W)e\}$

for $1 \leqslant l \leqslant r$ **do**
  for $1 \leqslant l' \leqslant r$ **do**
    for $1 \leqslant s \leqslant s_l$ **do**
      for $1 \leqslant s' \leqslant s_{l'}$ **do**
        for $1 \leqslant t \leqslant t_l$ **do**
          for $1 \leqslant t' \leqslant t_{l'}$ **do**
            Evaluate an expression N as in Theorem 6.1 for
            $L \leftarrow \mathbf{M}_{\theta_{l'}\mathcal{B}_{S_{l'}^{(s')}}}(g^{-1})$ and $R \leftarrow \mathbf{M}_{\theta_l\mathcal{C}_{T_l^{(t)}}}(g)$
            giving rise to the $n_l \times n_{l'}$ matrix that describes the action of ege on $\mathrm{Hom}_{FK}(S_l^{(s)}, T_l^{(t)})$ projected onto $\mathrm{Hom}_{FK}(S_{l'}^{(s')}, T_{l'}^{(t')})$, and put the result in the correct place in m.
          end for
        end for
      end for
    end for
  end for
end for

Proof. This is all evident from Theorem 6.1 and upon putting everything together as described in Theorem 3.1. □

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

REMARK 6.4. The six nested loops in Algorithm 2 and the corresponding six nested sums in the analysis in Theorem 6.3(ii) render it particularly important to optimise whatever is happening within these loops. We reap such optimisations from our special choice of bases, which allow for easy projection using traces. The resulting improvements in computational complexity over the tensor condensation implementation in the C-MEATAXE 2.4 explain well the performance improvements exhibited in the next section.

In the splitting field case (all $n_l = 1$) this can be seen especially easily. The dominant term within the six sums is $2d_l d_{l'}$. In the description of tensor condensation in [5], the same six nested loops are employed as here; however, in the innermost loop two matrix multiplications $A \cdot B \cdot C$ with $A \in F^{d_{l'} \times d_l}$, $B \in F^{d_l \times d_l}$ and $C \in F^{d_l \times d_{l'}}$, plus a multiplication of the resulting $d_{l'} \times d_{l'}$ matrix with a vector of length $d_{l'}$, are done. This amounts to

$$d_{l'}(2d_l - 1)d_l + d_{l'}(2d_l - 1)d_{l'} + 2d_{l'} = d_{l'}((2d_l - 1)d_l + (2d_l + 1)d_{l'} + 2)$$

elementary field operations. The dominant term in this expression is $(d_l + d_{l'})2d_l d_{l'}$, which is greater by a factor of $(d_l + d_{l'})$, the sum of the dimensions of the two K-composition factors. Note that, of course, this factor varies greatly as we go through the different pairs of isomorphism types of K-composition factors (the different summands in the l and l' loops). Furthermore, these different contributions are weighted by the multiplicities s_l, $s_{l'}$, t_l and $t_{l'}$. Therefore it is impossible to give a nice expression for the overall speed-up factor, since it will always be a weighted average over the individual factors $(d_l + d_{l'})$ for the different summands. In any case, we improve the critical part of the computation in the innermost loop by a factor of the sum of the dimensions of the two composition factors at hand. In the non-splitting field situation the same is true, but the weights of the contributions are changed by the values of n_l and $n_{l'}$. This analysis explains the observed speed-ups well.

We conclude this section with a lemma about numbers of elementary field operations for basic vector and matrix arithmetic.

LEMMA 6.5 (Complexity of basic matrix arithmetic). *Let F be a field, and let $M \in F^{a \times b}$ and $N \in F^{b \times c}$ be matrices over F. Furthermore, let $v \in F^{1 \times b}$ be a row vector. In all the following statements, we count additions as well as multiplications of elements of F as 'elementary field operations'.*

Then, the matrix product $M \cdot N$ can be computed using at most $a \cdot (2b - 1) \cdot c$ elementary field operations. The product $v \cdot N$ of the vector v and the matrix N can be computed using at most $(2b - 1) \cdot c$ elementary field operations.

For $a = c$, the $\mathrm{trace}(MN)$ of the product MN can be computed using at most $2ab - 1$ elementary field operations.

If $b = c$ and N is a companion matrix, the product $M \cdot N$ can be computed using at most $2ab$ elementary field operations, and the product $v \cdot N$ using at most $2b$ elementary field operations.

Proof. The product $v \cdot N$ can be computed by multiplying row i of N with the ith entry of v for $i = 1, 2, \ldots, b$ and summing up all the results. The scalar multiplications need bc elementary field operations, and then we have to do $b - 1$ additions of vectors of length c, resulting in a total of $(2b - 1) \cdot c$ operations.

To compute the matrix product $M \cdot N$, we have to multiply each row of M from the right by N. Thus this can be done in $a \cdot (2b - 1) \cdot c$ elementary field operations, by the results in the previous paragraph.

If $a = c$, then evaluating $\mathrm{trace}(MN)$ amounts to forming all scalar products of the ith row of M with the ith column of N and adding up all these scalars. Since such a scalar product costs $2b - 1$ elementary field operations and summing up needs another $a - 1$ additions, the total number of operations needed is $a(2b - 1) + a - 1 = 2ab - 1$.

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

Now let $b = c$ and let N be a companion matrix. Then a multiplication of a vector v by N amounts to shifting the vector v one entry to the right, multiplying the rightmost entry of v by the last row of N and adding the two resulting vectors. Neglecting the shift, this needs $2b$ elementary field operations. The multiplication of M by N can thus be done using at most $2ab$ elementary field operations. □

REMARK 6.6. In the preceding lemma we always give upper bounds, since in practical applications the number of necessary operations can be reduced by using zeros that occur in the matrices.

7. *Performance*

In this section we present empirical evidence for the performance of our new algorithm.

For two FG-modules V and W, the space of homomorphisms $\mathrm{Hom}_F(V, W)$, viewed as an FG-module by the action from formula (1), is isomorphic to the tensor product $V^* \otimes W$, where V^* denotes the contragredient module of V. Thus we can compare the result of the condensation of $\mathrm{Hom}_F(V, W)$ with that of $V^* \otimes_F W$ and show the difference in performance between our algorithm and the tensor condensation algorithm in the C-MEATAXE.

In Table 1 we present timings of computations, which were all done on a machine with a Pentium Core2 Quad Q6600 processor running at 2.4 GHz. The first column, marked G, shows the isomorphism type of G; the second column, marked q, shows the number of elements of the base field F; the third and fourth columns contain the dimensions of the two modules, and the product of those dimensions is the dimension of both $\mathrm{Hom}_F(V, W)$ and $V^* \otimes_F W$. The next two columns show the order $|K|$ of the condensation subgroup and the dimension of the condensed module. The columns marked HC and TC contain runtimes in seconds for the condensation of one element using HOMCOND (HC) and for the condensation of one element using TCOND (TC). Finally, the last column, marked Mem, contains the main memory requirement for a GAP session performing only the HOMCOND condensation without the precomputations. Note that an empty GAP session alone needs already about 100 MB just to load the library and packages on a 64-bit machine.

For the group Fi_{22}, we used as condensation subgroup a Sylow 3-subgroup of the 12th maximal subgroup, which is isomorphic to the symmetric group S_{10}. For HN, we used the extraspecial normal subgroup of order 2^{1+8} in the 4th maximal subgroup, which is of isomorphism type $2^{1+8} \cdot (A_5 \times A_5) \cdot 2$. For Fi_{23}, we used the extraspecial normal subgroup of order 3^{1+8} in the 7th maximal subgroup, which is of isomorphism type $3^{1+8} \cdot 2^{1+6} \cdot 3^{1+2} \cdot 2S_4$. For Ly, we used a non-normal subgroup of order 3125 in the 5th maximal subgroup $5^{1+4} : 4S_6$.

The modules for the group Fi_{23} in characteristic 2 have non-absolutely irreducible constituents when restricted to the condensation subgroup, whereas all the other examples demonstrate the splitting field case.

One should not expect this comparison to entirely exhibit the improved complexity of our algorithm as mentioned in Remark 6.4. The two implementations are substantially distinct:

TABLE 1. *Performance of* HOMCOND *and* TCOND (*with times in seconds*).

G	q	dim V	dim W	$\lvert K\rvert$	CDim	HC	TC	Mem
Fi_{22}	7	429	78	81	436	0.40	0.54	105
HN	5	626	626	512	1096	0.208	9.98	116
HN	5	8152	626	512	11096	308	2145	599
Fi_{23}	2	1494	1494	19683	684	10.4	26.0	175
Fi_{23}	2	19940	19940	19683	25542	61200	227591	6911
Ly	3	651	651	3125	185	2.357	5.33	104

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press

the C-MeatAxe is implemented completely in the C programming language, whereas our programs are implemented in the GAP language, with only the low-level finite field arithmetic implemented in C. Also, the implementations of the finite field arithmetic are quite different: the C-MeatAxe uses table lookup; the arithmetic in the cvec package (see [7]) used in our programs employs machine word operations and no tables. Furthermore, in fine details such as cache-awareness, already the nature of how the C-MeatAxe and GAP organise their memory accesses leads to a significant variance.

Note that we do not show the precomputation times, as the methods for obtaining a K-semi-simple basis are incomparable: whereas the C-MeatAxe uses a probabilistic approach based on peak words (see [4]) throughout, our implementation uses the deterministic techniques described in Algorithm 1.

The two techniques can behave completely differently in different situations. The major part of the precomputation in our algorithm is computing a composition series and semi-simplicity bases of the modules V and W restricted to the condensation subgroup. In particular, for the bigger modules, such as the one for Fi_{23} with dimension 19940 or the one for HN with dimension 8152, this takes a substantial amount of time. In the case of the C-MeatAxe, we found that in some examples its peak word search does not finish in any reasonable amount of time, forcing us to use *ad hoc* methods to come up with the input data for TCOND. The remaining precomputation to compute synchronised bases is basically negligible for our programs; the corresponding precomputations computing P- and Q-matrices in the tensor condensation programs of the C-MeatAxe are also negligible.

Acknowledgements. We are indebted to Jon Thackray for many interesting discussions on the subject of his own optimisations of the tensor condensation programs. This research was partially supported by the DFG grant HI 895/1-1.

References

1. J. A. Green, *Polynomial representations of* GL_n, Lecture Notes in Mathematics 830 (Springer, Berlin, 1980).
2. D. F. Holt and S. Rees, 'Testing modules for irreducibility', *J. Aust. Math. Soc. Ser. A* 57 (1994) 1–16.
3. F. Lübeck and M. Neunhöffer, 'Direct condense 2', 2000, http://www.math.rwth-aachen.de/∼DC/.
4. K. Lux, J. Müller and M. Ringe, 'Peakword condensation and submodule lattices: an application of the Meat-Axe', *J. Symbolic Comput.* 17 (1994) 529–544.
5. K. Lux and M. Wiegelmann, 'Condensing tensor product modules', *The atlas of finite groups: ten years on (Birmingham, 1995)*, London Mathematical Society Lecture Note Series 249 (Cambridge University Press, Cambridge, 1998) 174–190.
6. J. Müller and J. Rosenboom, 'Condensation of induced representations and an application: the 2-modular decomposition numbers of Co_2', *Computational methods for representations of groups and algebras (Essen, 1997)*, Progress in Mathematics 173 (Birkhäuser, Basel, 1999) 309–321.
7. M. Neunhöffer, 'cvec: a GAP-package implementing compressed vectors and matrices', 2006, http://www-groups.mcs.st-and.ac.uk/∼neunhoef/Computer/Software/GAP/cvec.html.
8. F. Noeske, 'Morita-Äquivalenzen in der algorithmischen Darstellungstheorie', PhD Thesis, RWTH Aachen, 2005.
9. R. A. Parker, 'The computer calculation of modular characters (the meat-axe)', *Computational group theory (Durham, 1982)* (Academic Press, London, 1984) 267–274.
10. M. Ringe, 'The MeatAxe – computing with modular representations', 2009, http://www.math.rwth-aachen.de/homes/MTX/.
11. A. J. E. Ryba, 'Condensation of symmetrized tensor powers', *J. Symbolic Comput.* 32 (2001) 273–289.
12. J. G. Thackray, 'Modular representations of some finite groups', PhD Thesis, University of Cambridge, 1981.
13. R. Wilson, J. Thackray, R. Parker, F. Noeske, J. Müller, K. Lux, F. Lübeck, C. Jansen, G. Hiss and T. Breuer, 'The modular Atlas project', 1998, http://www.math.rwth-aachen.de/∼MOC/.

Klaus Lux
Department of Mathematics
The University of Arizona
Tucson, AZ 85721-0089
USA

klux@math.arizona.edu

Max Neunhöffer
School of Mathematics and Statistics
Mathematical Institute
University of St Andrews
North Haugh, St Andrews
Fife KY16 9SS
United Kingdom

neunhoef@mcs.st-and.ac.uk

Felix Noeske
Lehrstuhl D für Mathematik
RWTH Aachen University
52056 Aachen
Germany

felix.noeske@math.rwth-aachen.de

https://doi.org/10.1112/S1461157012000071 Published online by Cambridge University Press