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 of 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 provide 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 R. Parker and J. Thackray devised the MeatAxe [Tha81, Par84], to recent progress in the modular Atlas project [WTP+98], the majority of results may be attributed to the application of computers. 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 always resisted a direct computational approach when they were firstly considered. To regain computational tractability, J. Thackray introduced a method called fixed-point reduction in his PhD thesis [Tha81], 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 $\varphi \in \mathrm{Hom}_{FG}(V, W)$ is 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 [Gre80, Section 6] or [Ryb01], for example.

The wide array of different available condensation algorithms for group algebras providing implementations which allow the condensation of, for example, permutation modules [LN00], induced modules [MR99, Noe05] and tensor products of modules [LW98, Noe05], is testimony of the method's usefulness. In this note we want to 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 sheds some new light on the condensation of tensor products of FG-modules.

We fix some further notation which will remain in effect throughout this note: Let V and W be finite dimensional FG-modules. Then $\mathrm{Hom}_F(V, W)$ is also a finite

Key words and phrases. Computational representation theory, condensation.
Partially supported by the DFG grant HI 895/1-1.

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 $K \leq G$ 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 condense with idempotents of the form

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

i.e. 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 by building the theoretical underpinning from which we subsequently construct our algorithm in Sections 4 to 6. Section 5 details the one-off calculations needed to prepare the input for the actual condensation process, which we describe in the following 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. The paper finishes with Section 7 in which we give some runtime examples of our algorithm to illustrate its efficiency.

2. Notation

In general, to compose maps from right to left, we write $\alpha \circ \beta$ to mean the map which first applies 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, i.e. 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, i.e. $\mathcal{B} = \mathcal{C}$, 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.

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

For a subgroup $K \leq G$, 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.

Owing to the limited available computational resources, 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 to construct 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 following 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 $1 \le i \le s$ and $1 \le j \le t$.

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, 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 intend 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, namely 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 in the following.

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 $\mathrm{Hom}_F(V,W)$. □

Remark 3.3. Considering V^λ and e_1 instead of V and e_λ in Lemma 3.2 amounts to a basis change of the algebra FK: the idempotent e_λ is nothing but the trace idempotent for the group $\{\lambda(k^{-1})k \mid k \in K \leq FK \leq FG\} \cong K$ 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 that λ is the trivial character of K, by replacing V by 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 may 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*

$$\mathrm{Hom}_F(S,T)e = \begin{cases} 0 & \text{if } S \not\cong_{FK} T, \\ \alpha \circ \mathrm{End}_{FK}(S) & \text{if } \alpha \text{ is an } FK\text{-isomorphism in } \mathrm{Hom}_{FK}(S,T), \end{cases}$$

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 is an immediate consequence of Lemma 3.2, 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 $S := S_i \leq V{\downarrow_K}$ and $T := T_j \leq W{\downarrow_K}$, 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 am 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 to compute 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

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

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, i.e.*

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

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*, i.e. the subspace* $\mathrm{Hom}_F(S,T)(1-e)$*, is given by* $\mathrm{Hom}_{FK}(S,T)^\perp$*, the orthogonal 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*

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

i.e. 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* $\varphi \in \mathrm{Hom}_F(S,T)$ *we define*

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

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*, i.e. we have* $\pi(\varphi) = \varphi e$.

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

4. The Practise

The aim of this section is to provide the means to address the problems given 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 $V = S_1 \oplus S_2 \oplus \cdots \oplus S_s$ 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 regard to a decomposition $W = T_1 \oplus T_2 \oplus \cdots \oplus T_t$ into a direct sum of simple *FK*-submodules), then we call $\mathcal{B}$ and $\mathcal{C}$ *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 in 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.

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 decompositions 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. □

Plugging together Lemma 4.2 and Lemma 3.8, we arrive at the following theorem, with which we overcome our problems stated at the beginning of Section 3.

Theorem 4.3. *Let S, S', T and T' be simple FK-modules together 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 $\mathcal{C}_T$, and $\mathcal{B}_{S'}$ and $\mathcal{C}_{T'}$, respectively, are synchronised, i.e. $\alpha(\mathcal{B}_S) = \mathcal{C}_T$ 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 $\{\alpha \circ \theta^k \mid k = 1, \ldots, n\}$ 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

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

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

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

Thus 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

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

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 $k\in\{0,\dots,n-1\}$. 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 $\mathrm{Hom}_{FK}(S',T')$, i.e. by applying Lemma 3.8. Also note that

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

□

Remark 4.4. In the above formulae we have

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

Theorem 4.3 illustrates that a large portion of the computational effort to calculate 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 involved vector spaces 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 $\{b_1,\dots,b_{d/n}\}\subseteq\mathcal{B}_S$ to be a subset which is an E-basis. Using these elements we define the sequence

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

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 chosen elements of $\mathcal{B}_S$ 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 $\mathrm{Hom}_F(S,T)$ with which it is particularly easy to work.

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*

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

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

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

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

Proof. By our chosen bases $\mathbf{M}_{\theta}^{{}^{\theta}\mathcal{B}_S}{}_{\mathcal{B}_S}$ maps the primitive element θ to the block-diagonal matrix having $C(\mu_\theta)$ along the diagonal and $\mathbf{M}_{\theta}^{{}^{\theta}\mathcal{C}_T}{}_{\mathcal{B}_S}(\alpha)$ 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 $\alpha \circ \theta$. 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 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 may identify $\mathrm{Hom}_F(V,W)e_\lambda$ by means of Lemma 3.2 with $\mathrm{Hom}_F(V^\lambda,W)e_1$, 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 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 only relevant at the very beginning of the computation: as the special basis of the condensed homomorphism space only relies 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 which form the framework of an algorithmic implementation:

Step 1: Determine the composition factors $\{S_1,\dots,S_s\}$ and $\{T_1,\dots,T_t\}$ 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 $eg_1e,\dots,eg_ke \in eFGe$. As 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 – applying synchronised and adapted bases – does not only allow for 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 fulfilled by a run of the MeatAxe [Tha81, Par84]. But as a key element of precondensation is the calculation of the synchronised semi simplicity bases, the output of current MeatAxe implementations is insufficient for our purposes. For this task we employ our own GAP implementation of the MeatAxe, which will be published separately in the form of the forthcoming GAP package chop. It features an augmented decomposition algorithm `Chop` whose added functionality 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 $0 = V_0 \lneqq V_1 \lneqq \cdots \lneqq V_l = V$ a composition series of V. Then an F-basis $\mathcal{B}$ of V is called *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 $i = 1, \ldots, l$.

Upon the user's request the chop package's main program `Chop` computes a basis adapted to the composition series found.

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 $\mathrm{Tr} : \mathrm{Hom}_F(V, W) \to \mathrm{Hom}_{FK}(V, W)$, i.e. 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.

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 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 $S \leq V$ 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 of $\mathcal{B} = \mathcal{B}_S \sqcup \mathcal{B}_Q$ 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 $K' \leq K$ 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 $\mathbf{M}_{\mathcal{B}}(k'k) = \mathbf{M}_{\mathcal{B}}(k')\mathbf{M}_{\mathcal{B}}(k)$ the claim follows. □

To use Lemma 5.3 to its full extent, we have to apply it several times: After choosing a subgroup chain $\{1\} = K_0 \lneqq K_1 \lneqq \cdots \lneqq K_l = K$ for K, we may iterate Lemma 5.3, and therefore 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 $|K| = \ell^m$, then 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 $2\ell^m$ a straightforward implementation would take.

Having solved the problem of how to quickly calculate semi simplicity bases, we now turn to the second open problem: 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 both tasks allows a simultaneous treatment.

To compare two simple modules, i.e. to test whether they are isomorphic or not, the MeatAxe uses Parker's standard basis technique (confer [Par84]). 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. However in general, 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 Holt and Rees introduce in [HR94, Section 3], which computes a primitive element.

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 result wanted. In particular, the basis change required needs only 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 its degree of 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 the light of Theorem 4.3, to ensure that the bases for both restricted modules $V{\downarrow_K}$ and $W{\downarrow_K}$ are mutually synchronised, i.e. any two isomorphic composition factors afford the same matrix representation, irrespective of the fact in which module 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.

Note that if the field F is 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 preparatory computations constituting the necessary calculations which provide the bases for Theorem 4.3 to be applied are:

(1) If Λ is non-trivial then replace V by V^λ and e_λ by $e := e_1$.
(2) Determine the composition factors of $V{\downarrow_K}$ and compute a basis of $V{\downarrow_K}$ which is adapted to the composition series found.

(3) Using the database produced in the previous step, find the composition factors of $W{\downarrow_K}$.
(4) As in [HR94], find primitive elements for the splitting fields of the composition factors, i.e. their endomorphism rings, and adapt their bases to these elements.
(5) 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 the 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 $B = (\langle \alpha \circ \theta^{i-1}, \alpha \circ \theta^{j-1} \rangle)_{1 \le i,j \le n}$. As we choose θ-adapted synchronised bases for S and T, Definition 3.5 gives

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

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.

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 respectively giving 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, $r \in \{1, \dots, m\}$, and $s \in \{1, \dots, n\}$. For two strictly increasing sequences of integers $1 \le i_1 < i_2 < \cdots < i_r \le m$ and $1 \le j_1 < j_2 < \cdots < j_s \le n$ we set $A^{[j_1,\dots,j_s]}_{[i_1,\dots,i_r]}$ to be the $(r \times s)$-submatrix $(a_{i_k,j_l})_{1 \le k \le r, 1 \le l \le s}$ of A. If $[i_1, \dots, i_r] = [1, \dots, m]$ or $[j_1, \dots, j_s] = [1, \dots, n]$ 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 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* $L := \mathbf{M}_{\theta'}^{{}^{\theta}\mathcal{B}_S}{}_{\mathcal{B}_{S'}}(g^{-1})$ *and* $R := \mathbf{M}_{\theta}^{{}^{\theta'}\mathcal{C}_{T'}}{}_{\mathcal{C}_T}(g)$. *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*

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

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 $k = 0, 1, \ldots, n-1$ *needs altogether 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 is illustrated in Figure 1 (there $d/n = 4$ and $d'/n' = 3$).

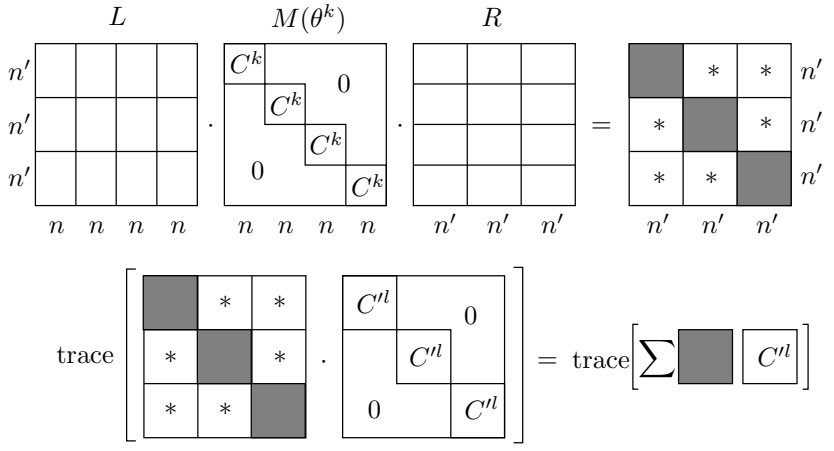

FIGURE 1. Illustration for the proof of Theorem 6.1

The first idea to avoid 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 $L \cdot \mathbf{M}_{\mathcal{B}_S}^{\mathcal{B}_S}(\theta^k) \cdot R$, due 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 are not necessary to compute!

The second idea is to use the sparseness of the occurring companion matrices 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 i-th grey block ${}^kA_{[(i-1)n'+1,\dots,in']}^{[(i-1)n'+1,\dots,in']}$ of ${}^kA$, we have to add the products

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

for $j = 1, \dots, n$. Because we need these products for all $k = 0, \dots, n-1$, we can compute the products $L_{[(i-1)n'+1,\dots,in']}^{[(j-1)n+1,\dots,jn]} \cdot C(\mu_\theta)^k$ inductively by just multiplying some previously known matrix by a companion matrix from the right. From the complexity results in Lemma 6.5 it follows that we need at most $\frac{d}{n}(n-1) \cdot 2n'n$ elementary field operations to compute all products $L_{[(i-1)n'+1,\dots,in']}^{[(j-1)n+1,\dots,jn]} \cdot C(\mu_\theta)^k$ for fixed i and all j and k. Given those products, evaluating the i-th grey block ${}^kA_{[(i-1)n'+1,\dots,in']}^{[(i-1)n'+1,\dots,in']}$ of ${}^kA$ for all k needs at most $\frac{d}{n} \cdot n'^2(2n-1)$ elementary field operations for the matrix multiplications with the R-parts, and another $\frac{d}{n}$ additions of $n' \times n'$ matrices to the final result needing $\frac{d}{n} \cdot n'^2$ 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, due 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 we only have 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 $l = 0, \dots, n'-1$. 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 needs 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 in the number in the theorem.

Note that in an actual implementation some optimisation is done if zeroes 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 that 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*

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

i.e. 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 $\{T_1, \ldots, T_r\}$ respectively denote a complete set of 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 sums 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:*

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.

Its execution needs at most

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

elementary field operations.

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

Remark 6.4. The six nested loops in Algorithm 2 and the corresponding six nested sums in the analysis in Theorem 6.3 render it particularly important to optimise things happening within these loops. We reap such optimisations from our special choice of bases allowing for easy projection using traces. The resulting improvements in computational complexity over previous implementations explain well the performance improvements exhibited in the following section.

In the splitting field case this can be seen especially easily: The dominant term within the 6 sums is $2dd'$. In previous implementations of tensor condensation, the same 6 nested loops were employed as here, however, in the innermost loop two matrix multiplications $A \cdot B \cdot C$ with $A \in F^{d' \times d}$, $B \in F^{d \times d}$, and $C \in F^{d \times d'}$ plus a multiplication of the resulting $d' \times d'$-matrix with a vector of length d' were done. This amounts to

$$d'(2d-1)d + d'(2d-1)d' + 2d' = d'((2d-1)d + (2d+1)d' + 2)$$

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 $1 \le l \le r$ **do**
  for $1 \le l' \le r$ **do**
    for $1 \le s \le s_l$ **do**
      for $1 \le s' \le s_{l'}$ **do**
        for $1 \le t \le t_l$ **do**
          for $1 \le t' \le t_{l'}$ **do**
            Evaluate an expression N as in Theorem 6.1 for
            $L \leftarrow \mathbf{M}_{{}^{\theta_l}\mathcal{B}_{S_l^{(s)}},\ {}^{\theta'_{l'}}\mathcal{B}_{S_{l'}^{(s')}}}(g^{-1})$ and $R \leftarrow \mathbf{M}_{{}^{\theta'_{l'}}\mathcal{C}_{T_{l'}^{(t')}},\ {}^{\theta_l}\mathcal{C}_{T_l^{(t)}}}(g)$
            giving rise to the $n_l \times n_{l'}$ matrix describing 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

elementary field operations, which is one order higher in the dimension of the K-composition factors.

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 $M \in F^{a\times b}$ and $N \in F^{b\times c}$ 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 with 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 i-th entry of v for $i = 1, 2, \ldots, b$ and summing up all 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 trace(MN) amounts to forming all scalar products of the i-th row of M with the i-th column of N and adding up all 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 $a(2b-1)+a-1 = 2ab-1$.

Let now $b = c$ and 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 both resulting vectors. Neglecting the shift this needs $2b$ elementary field operations. The multiplication of M by N thus can 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 zeroes 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 the one of $V^* \otimes_F W$ and show the difference in performance between our algorithm and the tensor condensation algorithm in the C-MeatAxe.

In Figure 2 we present timings, which were all done on a machine with 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. 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 respectively. 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 the 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 $2^{1+8}.(A_5 \times A_5).2$. For Fi_{23} we used the extraspecial normal subgroup of order 3^{1+8} in the 7th maximal subgroup, which is of isomorphism type $3^{1+8}.2^{1+6}.3^{1+2}.2S_4$. For Ly we used a non-normal subgroup of order 3125 in the 5th maximal subgroup $5^{1+4} : 4S_6$.

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	1 096	0.208	9.98	116
HN	5	8 152	626	512	11 096	308	2 145	599
Fi_{23}	2	1 494	1 494	19 683	684	10.4	26.0	175
Fi_{23}	2	19 940	19 940	19 683	25 542	61 200	227 591	6911
Ly	3	651	651	3125	185	2.357	5.33	104

FIGURE 2. Performance of HOMCOND and TCOND (times in seconds)

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. Due to the highly optimised code on both sides, the actual performance measured does not always follow the predictions of the complexity analysis, simply because in real-world examples the break-even point where complexity arguments come into effect is not reached. Also, both implementations are substantially distinct: The C-MEATAXE is implemented completely in the C programming language, whereas our programs are implemented in the GAP language, and only the low level finite field arithmetic is 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 [Neu06]) we are using in our programs uses machine word operations and no tables. Furthermore, in fine details like 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 to obtain a K-semisimple basis are incomparable: Whereas the C-MEATAXE uses peak words (see [LMR94]) throughout, our implementation uses the techniques described in Algorithm 1. Both 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 like 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 it's 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, and also the corresponding precomputation 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.

References

[Gre80] James A. Green. *Polynomial representations of* GL_n, volume 830 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1980.

[HR94] Derek F. Holt and Sarah Rees. Testing modules for irreducibility. *J. Austral. Math. Soc. Ser. A*, 57(1):1–16, 1994.

[LMR94] Klaus Lux, Jürgen Müller, and Michael Ringe. Peakword condensation and submodule lattices: an application of the MEAT-AXE. *J. Symbolic Comput.*, 17(6):529–544, 1994.

[LN00] Frank Lübeck and Max Neunhöffer. Direct condense 2, 2000. `http://www.math.rwth-aachen.de/~DC/`.

[LW98] Klaus Lux and Markus Wiegelmann. Condensing tensor product modules. In *The atlas of finite groups: ten years on (Birmingham, 1995)*, volume 249 of *London Math. Soc. Lecture Note Ser.*, pages 174–190. Cambridge Univ. Press, Cambridge, 1998.

[MR99] Jürgen Müller and Jens Rosenboom. Condensation of induced representations and an application: the 2-modular decomposition numbers of Co_2. In *Computational methods for representations of groups and algebras (Essen, 1997)*, volume 173 of *Progr. Math.*, pages 309–321. Birkhäuser, Basel, 1999.

[Neu06] Max 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`.

[Noe05] Felix Noeske. *Morita-Äquivalenzen in der algorithmischen Darstellungstheorie*. PhD thesis, RWTH Aachen, 2005.

[Par84] R. A. Parker. The computer calculation of modular characters (the meat-axe). In *Computational group theory (Durham, 1982)*, pages 267–274. Academic Press, London, 1984.

[Ryb01] A. J. E. Ryba. Condensation of symmetrized tensor powers. *J. Symbolic Comput.*, 32(3):273–289, 2001.

[Tha81] Jon G. Thackray. *Modular Representations of Some Finite Groups*. PhD thesis, University of Cambridge, 1981.

[WTP+98] Robert Wilson, Jon Thackray, Richard Parker, Felix Noeske, Jürgen Müller, Klaus Lux, Frank Lübeck, Christoph Jansen, Gerhard Hiss, and Thomas Breuer. The modular Atlas project, 1998. `http://www.math.rwth-aachen.de/~MOC/`.

E-mail address: `klux@math.arizona.edu`

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

E-mail address: `neunhoef@mcs.st-and.ac.uk`

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

E-mail address: `felix.noeske@math.rwth-aachen.de`

Lehrstuhl D für Mathematik, RWTH Aachen University, 52056 Aachen, Germany