

Computing and Investigating Permutation Characters

Saad Owaid Bedaiwi, Maroa Salah AL-shammery

Abstract— This work splits into two parts; in the first we present three actions and formulate them into GAP's codes, then we compute and investigate the associated permutation representations and its characters "permutation characters". In the second part, with the aid of the Meataxe GAP's package, we present a method and algorithm for investigating irreducible and permutation characters for a specific kinds of finite groups.

Index Terms— Permutation Characters, MeatAxe, GAP

I. INTRODUCTION

Recall that a representation of a group G on a set S is a homomorphism $\rho : G \rightarrow \text{Aut}(S)$, when S is a vector space it's called linear representation and if S is just a set, called permutation representation. ρ defines a G -set structure on S by the map $G \times S \rightarrow S, (g,s) \rightarrow \rho(g)(s)$. Permutation characters can be considered as powerful tool to extract information about subgroups of a given group. T. Breuer and G. Pfeiffer [3], describe three deferent methods to compute all those characters of a finite group that have certain properties of transitive permutation characters. In our work, we present three actions mentioned in [2], and formulate them into GAP's codes, then we compute and investigate the associated permutation representations and its permutation characters. Also, with the aid of the Meataxe GAP's package [5], we present a method and algorithm for investigating irreducible and permutation characters for a specific kinds of finite groups.

II. PERMUTATION CHARACTERS

A. Actions under investigation

Let S_n be the symmetric group on n symbols. Throughout this work, we consider the following actions with the associated permutation representations, provided that, they defined over an arbitrary field K , see [2] for more details.

(I) The representation associated to the conjugacy action of S_n on the subset $M \subseteq GL_n(K)$ defined by: $\sigma.M = \sigma M \sigma^{-1}$ where $\sigma \in S_n$, is called the conjugacy representation of S_n and will denoted by ρ_0 .

(II) Let G be subgroup of $S_n \times S_n$, then G act on $M \subseteq GL_n(K)$ by

$(\alpha, \sigma).M = \alpha M \sigma^{-1}$ for $(\alpha, \sigma) \in G$, denote by ρ_1 for the associated permutation representation.

(III) Let G be a subgroup of $(S_k \times S_{n-k}) \times (S_k \times S_{n-k})$ then G is act on

$M \subseteq GL_n(K)$ by $(\alpha, \sigma).M = \alpha M \sigma^{-1}$ where $(\alpha, \sigma) \in G$, denote by ρ_2 the associated permutation representation.

B. Definition [3]: Let G be any group act on a set S and ρ is the corresponding permutation representation. The map $\tau : G \rightarrow K$, defined by:

$$\tau_G(g) = \text{Order}(\text{Fix}_S(g)) = \text{Order}\{s \in S; sg = s\}.$$

is called the permutation character of G on S .

C. Remark:

1. One can embed the symmetric group S_n in the general linear group $GL_n(K)$, by identify any permutation $\sigma \in S_n$ with the matrix:

$$[\sigma]_{i,j} = \begin{cases} 1 & i = \sigma(j), \\ 0 & \text{otherwise.} \end{cases}$$

Which means permuting rows of the identity matrix.

2. All our computations are made utilizing the group algebra system GAP, see [5].

Our aim in this paper to compute the permutation character induced from the permutation representations: conjugacy representation ρ_0, ρ_1 and ρ_2 , for a finite groups carry out properties listed in tables below. Our computations are divided into three cases:

Case I: Find the permutation character of a finite group D with properties as listed in table (1) via the conjugacy action (I) on $S1:=D, S2:=$ derived subgroup of $D. K:= GF(p)$ (the Galois field where $p:=$ prime in particular $p=7$).

Constructing the finite groups with the required properties.

```
gap> alsma:=AllSmallGroups(n,certain properties as
specified);;
gap> iso:=IsomorphismPermGroup(alsma[i]); #for i in
[1..Size(alsma)]
gap> H:=Image(iso);D:=H;;
gap> gen:= GeneratorsOfGroup(D);;
gap> L:=List(gen,
>g->PermutationMat(g,n,GF(K)));;
gap> groumat:=GroupWithGenerators(L);
gap> S1:=groumat;;
gap> S2:=DerivedSubgroup(groumat);
```

Constructing the conjugacy action.

```
gap> action1:=function(a,g)
> local m;
> m:=List(a,x->Permuted(x,g));
> m:=Permuted(m,g^-1);
> m:=ImmutableMatrix(DefaultFieldOfMatrix(a),>m); # for
efficiency make compact
> return m;
> end;
function( a, g ) ... end
```

Saad Owaid Bedaiwi, Department of Mathematics , University of Al-Mustansiryah/College of Science, Baghdad- Iraq.

Maroa Salah AL-shammery, Department of Mathematics , University of Al-Mustansiryah/College of Science,Baghdad-Iraq.

Computing the permutation character.

```
gap> PermutationCharacter(D,S1,action1);
gap> PermutationCharacter(D,S2,action1);
```

D. Example.

A finite group of order 6 which has the properties: non-abelian, solvable and non-perfect.

```
gap> alsma:=AllSmallGroups(6,IsAbelian,
>false,IsSolvable,true,IsPerfect,false);
gap> iso:=IsomorphismPermGroup(alsma[1]);
<action isomorphism>
gap> H:=Image(iso);D:=H;;
gap> gen:= GeneratorsOfGroup(D);
gap> L:=List(gen,
>g->PermutationMat(g,6,GF(7)));
gap> groumat:=GroupWithGenerators(L);
<matrix group with 2 generators>
gap> S1:=groumat;;
gap> S2:=DerivedSubgroup(groumat);
<group of 6x6 matrices over GF(7)>
gap> PermutationCharacter(D,S1,action1);
Character(CharacterTable(Group([(1,2)(3,6)(4,5),(1,3,5)(2,4,6)])),[ 6, 2, 3 ])
gap> PermutationCharacter(D,S2,action1);
Character(CharacterTable(Group([(1,2)(3,6)(4,5),(1,3,5)(2,4,6)])),[ 3, 1, 0 ])
gap> perm1:=Action(D,S1,action1);
gap># check if the action #is transitive, compute its transitivity degree.
gap>IsTransitive(perm1); Transitivity(perm1);
false
0
gap> perm2:=Action(D,S2,action1);
gap>IsTransitive(perm2); Transitivity(perm2);
true
3
```

If we run the above GAP code on different kinds of groups having multiple properties, we have results displayed as shown in table (1). Note that we denote by *tran* and *No.tran* to check whether action1 is transitive, the degree of the transitivity respectively.

Table (1)

[?]	Abelian	Solvable	Perfect	S	Permutation character	Is tran	No.tran
6	NO	YES	NO	??1	[6, 2, 3]	NO	0
6	NO	YES	NO	??2	[3, 1, 0]	YES	3
60	NO	NO	YES	??1	[60,3,4,5,5]	NO	0
60	NO	NO	YES	??2	[60,3 4,5,5]	NO	0
120	NO	NO	NO	??1	[120,12,6,8,4,6,5]	NO	0
120	NO	NO	NO	??2	[60,6,3,4,2,3,5]	NO	0

Case II: Find the permutation character of a finite group $D = H \times H$ with properties as listed in table (2) via the permutation action (II) on $S1:=D, S2:=H. K:= GF(p)$.
Constructing the finite groups with the required properties.

```
gap> alsma:=AllSmallGroups(n,certain properties as
specified);
gap> iso:=IsomorphismPermGroup(alsma[i]); #for i in
[1..Size(alsma)]
gap> H:=Image(iso);
gap> D:=DirectProduct(H,H); Size(D);
gap> gen:= GeneratorsOfGroup(D);
gap> L:=List(gen,
>g->PermutationMat(g,Size(D);,GF(p)));
gap> groumat:=GroupWithGenerators(L);
gap> S1:=groumat;;
gap> genH:= GeneratorsOfGroup(H);
gap> LH:=List(genH,
>g->PermutationMat(g,Size(H),GF(p)));
gap> groumatH:=GroupWithGenerators(LH);
gap> S2:=groumatH;;
gap> p1:=Projection(D,1); # decompose an element in its
parts
gap> p2:=Projection(D,2);
```

Constructing the permutation action (II).

```
gap> action2:=function(a,pair)
> local m,g;
> g:=Image(p1,pair);
> m:=List(a,x->Permuted(x,g));
> m:=Permuted(m,Image(p2,pair)^-1);
> m:=ImmutableMatrix(DefaultFieldOfMatrix(a)
>,m); # for efficiency compact
> return m;
> end;
function( a, pair ) ... end
```

#Computing the permutation character.

```
gap> PermutationCharacter(D,S1,action2);
gap> PermutationCharacter(D,S2,action2);
gap> perm1:=Action(D,Omg1,action2);
gap>IsTransitive(perm1); Transitivity(perm1);
gap> perm2:=Action(D,S2,action2);
gap>IsTransitive(perm2); Transitivity(perm2);
```

If we run the above GAP code on different kinds of groups having multiple properties, we have results displayed as shown in table (2).

Table (2)

[?]	Abelian	Solvable	Perfect	S	Permutation character	Is tran	No. tran
6	NO	YES	NO	??1	[36,0,0,0,12,0,0,0,18]	NO	0
14	NO	NO	YES	??1	[196,0,0,0,0,28,0,0,0,0,98,0,0,0,0,98]	NO	0
14	NO	NO	NO	??2	[14,0,0,0,0,2,0,0,0,0,0,7,0,0,0,0,7]	YES	1

Case III: Find a permutation character of $D = M \times N$, where M and N are finite groups such that: $|M| = 4, |N| = 4 - m$, abelian, solvable and non-perfect. The permutation action (III) on $S:=D, K:= GF(13)$.

Constructing the finite group with the required properties.
gap> alsma:=AllSmallGroups(4,IsAbelian,true,IsSolvable,tru
e,

```
>IsPerfect,false);
[ <pc group of size 4 with 2 generators>, <pc group of size 4
with 2 generators> ]
gap> iso:=IsomorphismPermGroup(alsma[2]);
[ f1, f2 ] -> [ (1,2), (3,4) ]
gap> H:=Image(iso);;
gap> rows:=Tuples([0,1],4);;zerooone:=Tuples(rows,4);;
gap> m:=3;;stb:=Stabilizer(H,[1..m],OnSets);
Group([ (1,2) ])
gap> D:=DirectProduct(stb,stb);
Group([ (1,2), (3,4) ])
gap> p1:=Projection(G,1);
1st projection of Group([ (1,2), (3,4) ])
gap> p2:=Projection(G,2);
2nd projection of Group([ (1,2), (3,4) ])
```

Constructing the permutation action (III).

```
gap> gen:= GeneratorsOfGroup(D);;
gap> L:=List(gen,g->PermutationMat(g,4,GF(13)));;
gap> genter:=GroupWithGenerators(L);
<matrix group with 2 generators>
gap> S:=genter;;
gap> perm:=Action(G,S,action2);
Group([ (1,3)(2,4), (1,3)(2,4) ])
gap> IsTransitive(perm); Transitivity(perm);
false
0
gap> PermutationCharacter(G,S,action2);
Character( CharacterTable( Group([ (1,2), (3,4) ])),[ 4, 0, 0, 4
] )
```

III. Computing and investigating permutation characters

The permutation character of a transitive action is called a transitive permutation character. With the aid of the GAP's Package MeatAxe [5] we investigate the permutation characters of some finite groups which are listed in table (3) with their sizes and properties.

A. Lemma [6]: Let G be any group, if τ is a transitive permutation character then:

- τ is a character of G .
- $\tau(g) \geq 0, \forall g \in G$.
- $d_i = \tau(1)$ divides $|G|$.
- $\tau(g^n) \geq \tau(g)$, for $g \in G$ and integers n .
- $\langle \tau, 1_G \rangle = 1$.
- the multiplicity of $\psi \in \text{Rat}(G)$ as a constituent of π is at most $\psi(1) / \langle \psi, \psi \rangle$.
- $\tau(g) = 0$ if the order of g does not divide $|G|$.
- $\tau(1) \cdot |N_G(g)|$ divides $\tau(g) \cdot |G|$ for all $g \in G$.
- $\tau(g) \leq \frac{|G| - \pi(1)}{|g^G| \cdot |Gal_G(g)|}$ for all $g \in G \setminus \{1\}$.

The character τ with the properties of Lemma (2.1) is called a possible permutation character of G . The following is a procedure to list possible permutation characters.

B. The Combinatorial Approach Method [7]:

This method builds up a reasonably small set of possible permutation characters. The obtained set satisfies some of the conditions stated in Lemma (2.1) which use the bounds on coefficients of rationally irreducible characters given by branch (f).

C. Algorithm [4]:

Let $\text{Irr}(G) = \{1_G = \chi_1, \chi_2, \dots, \chi_n\}$ be the set of all irreducible characters of a given group G , fix a degree d that divides $|G|$. By successively choosing coefficients a_i for $i = 1, 2, \dots, n$, all coefficients vectors (a_1, a_2, \dots, a_n) with $a_1 = 1$ are generated that satisfy

$$0 \leq a_i \leq \min \left\{ \frac{\chi_i(1)}{\langle \chi_i, \chi_i \rangle}, \frac{d - \sum_{j=1}^{i-1} a_j \chi_j(1)}{\chi_i(1)} \right\}$$

for $i = 2, \dots, n$.

and $\sum_{j=1}^n a_j \chi_j(1) = d$.

For each such vector, the character satisfies conditions (a), (c), (e), and (f) of Lemma (2.1), and we can check the other conditions. Moreover, every true permutation character of G of degree d occurs among these.

In the table (3) the following notations are used: the numbers of conjugacy classes of elements

$No. C_G(g)$, the number of rational classes $No. \text{Rat}(G)$, the number of possible permutation characters $No. \tau(g)$, the number of conjugacy classes of subgroups $No. C_G(H)$ where H subgroup of G , the numbers of transitive permutation characters $No. \text{tra } \tau(g)$, the structure description of the group G $stru(G)$, the numbers of absolutely irreducible characters of G over a field K $No. \text{absol Irr}_K(G)$, the numbers of irreducible characters of G over K $No. \text{Irr}_K(G)$.

Let us started with an example of a group of size 6, which is non-abelian, solvable and non-perfect.

D. Example

```
gap> alsma:=AllSmallGroups(6,IsAbelian,
>false,IsSolvable,true,IsPerfect,false);
[ <pc group of size 6 with 2 generators> ]
gap> iso:=IsomorphismPermGroup(alsma[1]);;
gap> G:=Image(iso);;
gap> NrConjugacyClasses(G);
3
gap> RationalClasses( G );;Size(RationalClasses( G ));
3
gap> tbl:=CharacterTable(G);;
gap> Display(tbl);
CT1
2 1 1 .
3 1 . 1
1a 2a 3a
2P 1a 1a 3a
3P 1a 2a 1a
```

```
X.1  1 1 1
X.2  1 -1 1
X.3  2 . -1
gap> PermChars( tbl);Size(PermChars( tbl));
4
gap> ConjugacyClassesSubgroups(G);
gap> Size(ConjugacyClassesSubgroups( G ));
4
gap> tom:=TableOfMarks(G);Display(tom);
1: 6
2: 3 1
3: 2 . 2
4: 1 1 1 1
```

```
gap> PermCharsTom( tbl,
>tom );Size(PermCharsTom( tbl, tom ));
4
gap> StructureDescription( G );
"S3"
```

Construct the permutation module of G over $GF(p)$ where $p = 7$.

```
gap> K:=GF(7);module := PermutationGModule(G,K);
rec( dimension := 6, field := GF(7), generators := [ <
immutable compressed matrix 6x6 over GF(7) >,
< immutable compressed matrix 6x6 over GF(7) > ],
isMTXModule := true )
gap> COF := MTX.CollectedFactors( module );
[ [ rec( IsAbsolutelyIrreducible := true, IsIrreducible:=
true,dimension := 1,field := GF(7),generators := [ [ [ Z(7)^0 ]
], [ [ Z(7)^0 ] ] ], isMTXModule := true,smashMeataxe := rec(
algebraElement := [ [ [ 1, 2 ], [ 1, 2 ] ], [ Z(7)^3, Z(7)^5,
Z(7)^2, Z(7)^2 ] ],algebraElementMatrix:=[[Z(7)^0]
],characteristicPolynomial := x_1-Z(7)^0,charpolFactors :=
x_1-Z(7)^0, degreeFieldExt := 1, ndimFlag := 1,
nullspaceVector := [ Z(7)^0 ] ) ), 1 ],[ rec( IsIrreducible :=
true, dimension := 1, field := GF(7), generators := [ [ [ Z(7)^3
] ], [ [ Z(7)^0 ] ] ],isMTXModule := true,smashMeataxe :=
rec( algebraElement := [ [ [ 1, 2 ], [ 3, 1 ] ], [ Z(7)^3, Z(7),
0*Z(7), Z(7)^3 ] ],algebraElementMatrix := [ [ Z(7) ] ],
characteristicPolynomial := x_1+Z(7)^4,charpolFactors :=
x_1+Z(7)^4, ndimFlag := 1, nullspaceVector := [ Z(7)^0 ] ) ),
1 ],[ rec( IsAbsolutelyIrreducible := true,IsIrreducible := true,
dimension := 2, field := GF(7),generators := [ [ [ Z(7)^0,
0*Z(7) ], [ Z(7)^3, Z(7)^3 ] ], [ [ 0*Z(7), Z(7)^0 ], [ Z(7)^3,
Z(7)^3 ] ] ],isMTXModule := true, smashMeataxe := rec(
algebraElement := [ [ [ 2, 1 ] ], [ Z(7)^0, Z(7)^0, Z(7)^3 ]
],algebraElementMatrix := [ [ Z(7)^2, Z(7)^2 ], [ Z(7)^5,
Z(7)^4 ] ],characteristicPolynomial := x_1^2+x_1+Z(7)^5,
charpolFactors := x_1+Z(7)^2, degreeFieldExt := 1,ndimFlag
:= 1, nullspaceVector := [ Z(7)^4, Z(7)^0 ] ) ), 2 ] ]
gap> List( cf,
>x -> MTX.IsAbsolutelyIrreducible(x[1]) );
[ true,true, true ]
gap> List( cf, x -> MTX.IsIrreducible(x[1]) );
[ true, true, true ]
```

Table (3)

[n]	abelian	Solvable	Perfect	$\frac{ \text{Aut}(G) }{ G }$							
6	NO	YES	NO	3	3	4	4	4	$?_3$	2	3
24	NO	YES	NO	12	9	16	16	16	$?_4 \times ?_3$	6	9
80	NO	YES	NO	26	11	20	20	20	$?_{40}: ?_2$	10	16
60	NO	YES	NO	30	10	12	12	12	$?_3 \times (?_3: ?_4)$	4	3
120	NO	NO	NO	7	7	19	19	19	$?_5$	2	2
144	NO	YES	NO	48	16	33	33	33	$?_8 \times ?_{18}$	12	24
168	NO	NO	YES	6	5	12	15	15	$????(3,2)$	2	2
240	NO	NO	NO	18	14	31	31	31	$?_2 \times ???(2,5)$	4	5
360	NO	NO	NO	21	14	41	41	41	$?_3 \times ?_5$	3	3
360	NO	NO	YES	7	6	22	22	22	$?_6$	2	2
720	NO	NO	NO	24	19	78	79	79	$??? 2,4 : ?_4$	6	6
1500	NO	NO	NO	125	14	31	13	13	$?_{25} \times ?_5$	2	4

APPENDIX

• The Meataxe algorithm

The Meat-axe is a fundamental tool in computational representation theory, most often used to test irreducibility of a finite matrix group or algebra, and in the case of reducibility to construct an invariant subspace. A number of versions have been described in the literature, first by R. Parker [7] in 1984 and later by other [8,9] The implementations of the Meat-axe in the computer algebra systems GAP and MAGMA [11] are based on the version of D. F. Holt and S. Rees [10].

Algorithm:

We assume that K is a field and $\delta: A \rightarrow K^{n \times n}$ is a matrix representation with representation module $V := K^n$.

Input: An n -dimensional matrix representation δ of A in terms of matrices

$(a_1 := \delta(a_1), \dots, a_k := \delta(a_k))$ for a generating system (a_1, \dots, a_k) of A

Output: Either the information that δ is irreducible, or matrix representations of A on an A -submodule W and on the factor module V/W given by matrices for the generating system (a_1, \dots, a_k) of A .

REFERENCES

- [1]Alperin J. L., Bell R. B. Groups and representations. – Springer Science & Business Media, 2012. – T. 162.
- [2]Cherniavsky Y., Bagno E. Permutation representations on invertible matrices // Linear Algebra and its Applications. – 2004. – C. 494 –518.
- [3]Breuer T., Pfeiffer G. Finding possible permutation characters //Journal of Symbolic Computation. – 1998. – T. 26. – №. 3. – C. 343-354.
- [4]Lux K., Pahlings H. Representations of groups a computational approach. – Cambridge University Press, 2010. – T. 124.
- [5]The GAP Group, GAP -- Groups, Algorithms, and Programming, Version 4.7.9; 2015. (<http://www.gap-system.org>).
- [6]Isaacs I. M. Character theory of finite groups. – Courier Corporation, 2013.
- [7]Parker R. A. The computer calculation of modular characters (the meat-axe) //Computational Group Theory, Academic Press, London. – 1984. – C. 267-274.
- [8]Ivanyos G., Lux K. Treating the exceptional cases of the MeatAxe //Experimental Mathematics. – 2000. – T. 9. – №. 3. – C. 373-381.
- [9]Neumann P. M., Praeger C. E. Cyclic matrices and the Meataxe //Groups and Computations. – 2001. – T. 3. – C. 291-300.
- [10]Holt D. F., Rees S. Testing modules for irreducibility //Journal of the Australian Mathematical Society (Series A). – 1994. – T. 57. – №. 01. – C. 1-16.
- [11]Bosma W., Cannon J., Playoust C. The Magma algebra system I: The user language //Journal of Symbolic Computation. – 1997. – T. 24. – №. 3. – C. 235-265.