The Gap Condition for $S_5$ and GAP Programs

Masaharu MORIMOTO* and Mamoru YANAGIHARA†

(Received November 17, 1995)

Abstract

In transformation groups on manifolds, it has been an interesting problem to ask whether for a given finite group $G$, there exists a real $G$-module $V$ such that $\dim V^P > 2 \dim V^{>P}$ for all subgroups $P$ of prime power order and such that $V^H = 0$ for certain large subgroups $H$ of $G$. This paper provides GAP programs to show that $S_5$ does not admit such a real $S_5$-module $V$.

KEYWORDS: GAP, fixed point, gap condition.


1. Introduction

Let $G$ be a finite group. A real $G$-module $V$ is said to satisfy the gap condition if $\dim V^P > 2 \dim V^{>P}$ for all subgroups $P$ of prime power order and such that $V^H = 0$ for certain large subgroups $H$ of $G$ (precisely to say, for all $H \in \mathcal{L}(G)$ defined below). The existence problem of such modules is closely related to equivariant surgery theory (cf. [PR], [M1]) and construction of exotic actions on closed, smooth manifolds. Our purpose in the present paper is to show that $S_5$ the symmetric group of degree 5 does not admit a real $S_5$-module satisfying the gap condition, employing the computer software GAP (Groups, Algorithms, and Programming) [S]. This result was announced in [MY] (1994) and the present paper includes the details.

Let $G$ be a finite group. Let $S(G)$ denote the set of all subgroups of $G$ and $\mathcal{P}(G)$ the set of all subgroups of $G$ of prime power order. (Particularly, the trivial group $\{1\}$ belongs to $\mathcal{P}(G)$.)

Let $\mathcal{P}(G)$ denote the set of all subgroups of $G$ and $\mathcal{P}(G)$ the set of all subgroups of $G$ of prime power order. (Particularly, the trivial group $\{1\}$ belongs to $\mathcal{P}(G)$.)

For each prime $p$ we define a characteristic subgroup $G^p$ by

$$G^p = \bigcap \{ H \leq G \mid |G/H| \text{ is a power of } p \}.$$  

Then the set $\mathcal{L}(G)$ mentioned above is defined by

$$\mathcal{L}(G) = \{ H \leq G \mid H \supset G^p \text{ for some prime } p \}.$$

Let $\mathcal{M}(G)$ denote the complementary set $S(G) \setminus \mathcal{L}(G)$. If $p$ and $q$ are primes or 1 and $n$ is a positive integer, let $\mathcal{G}^p_n$ denote the family of all finite groups $K$ having a series $P \triangleleft H \triangleleft K$ such that $P$ is of $p$-power order, $H/P$ is a cyclic group of order $n$, and $K/H$ is of $q$-power order. Set

$$\mathcal{G}^p_n = \bigcup_q \mathcal{G}^q_n, \quad \mathcal{G}_p = \bigcup_q \mathcal{G}^q_p, \quad \mathcal{G} = \bigcup_q \mathcal{G}^q, \quad \mathcal{G}^{\text{odd}}[2] = \bigcup_{p, q \text{ odd}} \mathcal{G}^p_p[2].$$

If a finite group $K$ does not belong to $G$ then $K$ is called an Oliver group. By [O, Theorem 7], a finite group $K$ is an Oliver group if and only if $K$ has a smooth fixed-point free action on a disk. These sets $\mathcal{L}(G)$ and

*Department of Environmental and Mathematical Sciences, Faculty of Environmental Science and Technology, Okayama University, Okayama, 700 Japan.
†Department of Industrial and Information Studies, Komatsu College, Komatsu, Ishikawa, 923 Japan.
The organization of the paper is as follows. In Section 2, we give programs to determine the sets $\mathcal{P}(G)$, $\mathcal{L}(G)$ and $\mathcal{P}2\mathcal{S}(G)_{\text{odd}}$. In Section 3, we present programs to compute the fixed point dimensions of real $S_5$-modules, related to the gap condition. In Section 4, we explain how to use the obtained results in Section 3 in order to prove Theorem 1.2.

2. Structure of subgroups of $G$

We perform computation using the computer software GAP. Let us begin with giving GAP the definition of the group $G$ for which we perform computation. As a usual method in GAP, definition of a group is described by generators being permutations. This is done with the built-in function `Group(-)`. For example, since the symmetric group $S_5$ of degree 5 is generated by the cyclic permutations $(1,2,3,4,5)$ and $(1,2)$, we can make GAP realize the definition of $S_5$ in the form:

$$G := \text{Group}((1,2,3,4,5), (1,2));$$

The set of all conjugacy classes $CCS(G) (= \text{CCSG})$ of subgroups of $G$ is obtained by the built-in function `ConjugacyClassesSubgroups(-)`:

$$\text{CCSG} := \text{ConjugacyClassesSubgroups}(G);$$

and the complete set $RCCS(G) (= \text{RCCSG})$ of representatives of $CCS(G)$ is obtained by the built-in function `List(-,-)`:

$$\text{RCCSG} := \text{List}(\text{CCSG}, h \rightarrow \text{Representative}(h));$$

For example, we obtain the following result in the case $G = S_5$.

**Result 2.1.** There are 19 conjugacy classes of subgroups of $S_5$. They have the following representatives.
RCCSG[1] = Subgroup( G, [ ] ),
RCCSG[2] = Subgroup( G, [ (4,5) ] ),
RCCSG[3] = Subgroup( G, [ (2,3)(4,5) ] ),
RCCSG[4] = Subgroup( G, [ (3,4,5) ] ),
RCCSG[5] = Subgroup( G, [ (2,3)(4,5), (2,4)(3,5) ] ),
RCCSG[6] = Subgroup( G, [ (2,3)(4,5), (2,4,3,5) ] ),
RCCSG[7] = Subgroup( G, [ (4,5), (2,3) ] ),
RCCSG[8] = Subgroup( G, [ (1,2,3,4,5) ] ),
RCCSG[9] = Subgroup( G, [ (3,4,5), (4,5) ] ),
RCCSG[10] = Subgroup( G, [ (3,4,5), (1,2)(4,5) ] ),
RCCSG[11] = Subgroup( G, [ (4,5), (1,3,2) ] ),
RCCSG[12] = Subgroup( G, [ (4,5), (2,3), (2,4)(3,5) ] ),
RCCSG[13] = Subgroup( G, [ (1,2,3,4,5), (2,5)(3,4) ] ),
RCCSG[14] = Subgroup( G, [ (2,3)(4,5), (2,4)(3,5), (3,4,5) ] ),
RCCSG[15] = Subgroup( G, [ (4,5), (1,3,2), (2,3) ] ),
RCCSG[16] = Subgroup( G, [ (1,2,3,4,5), (2,5)(3,4), (2,3,5,4) ] ),
RCCSG[17] = Subgroup( G, [ (2,3)(4,5), (2,4)(3,5), (3,4,5), (4,5) ] ),
RCCSG[18] = Subgroup( G, [ (1,3,2), (2,4,3), (2,3)(4,5) ] ),
RCCSG[19] = Subgroup( G, [ (1,2,3,4,5), (1,2) ] ) = G.

In our computation of $\mathcal{L}(G)$, we use

$$\mathcal{L}(G)_{normal} = \{ H \in \mathcal{L}(G) \mid H < G \} = L_{Gnormal},$$

and the next function makeL_{Gnormal}(-) computes the set $\mathcal{L}(G)_{normal}$.

makeL_{Gnormal} := function()
    local S, H, i, ns, ni;
    S := [];
    ns := Length(RCCSG);
    for i in [1 .. ns] do
        H := RCCSG[i];
        ni := Index(G, H);
        if IsPrimePowerInt(ni) and IsNormal(G, H) then
            Add(S, i);
        elif ni = 1 then
            Add(S, i);
        fi;
    od;
    return S;
end;

Program 2.2.

After making GAP read Program 2.2, we can obtain $\mathcal{L}(G)_{normal}$ by typing

L_{Gnormal} := makeL_{Gnormal}();

in GAP.
Next we give a function $\text{testLG}(-)$ which checks whether a subgroup $H$ lies in $\mathcal{L}(G)$ or not. If $H \in \mathcal{L}(G)$ then $\text{testLG}(-)$ returns true and else false. This $\text{testLG}(-)$ is given by a program including a function $\text{isSubConjugate}(-,-)$ that assigns to subgroups $\text{RCCSG}[h]$ and $\text{RCCSG}[k]$, true if $\text{RCCSG}[h]$ is conjugate to a subgroup of $\text{RCCSG}[k]$ and false otherwise.

\begin{verbatim}
isSubConjugate := function(k, h)
    local size_k, size_h, conj, hh;
    size_k := Size(\text{RCCSG}[k]);
    size_h := Size(\text{RCCSG}[h]);
    if (size_k = Size(G)) or (k = h) then
        return true;
    fi;
    if not (IsInt(size_k/size_h)) then
        return false;
    fi;
    if size_k = size_h then
        return false;
    fi;
    for hh in Elements(\text{CCSG}[h]) do
        if IsSubgroup(\text{RCCSG}[k], hh) then
            return true;
        fi;
    od;
    return false;
end;
\end{verbatim}

Program 2.3.

The function $\text{testLG}(-)$ is given by the program:

\begin{verbatim}
testLG := function(h)
    local h1;
    for h1 in LGnormal do
        if isSubConjugate(h, h1) then
            return true;
        fi;
    od;
    return false;
end;
\end{verbatim}

Program 2.4.

The set $\mathcal{L}(G) (= \mathcal{LG})$ is computed by the function $\text{makeLG}()$:

\begin{verbatim}
makeLG := function()
    local S, n, i;
    S := [];
    n := Length(RCCSG);
    for i in [1..n] do
        if testLG(i) then
            S := S [i];
        fi;
    od;
    return S;
end;
\end{verbatim}

Program 2.5.
\begin{verbatim}
Add(S, i);

end;

return S;

end;

Program 2.5.

Result 2.6. If $G = S_5$ then $LG = \{18, 19\}$, i.e. $L(G) = [RCCSG[18], RCCSG[19]]$.

Let $Prime(G) (= PrimeG)$ be the set of primes dividing $|G|$ (the order of $G$). $Prime(G)$ is computed by

$$
PrimeG := \text{Set}(\text{Factors}(\text{Size}(G)));
$$

The next function $\text{coSylow}(-)$ assigns to a prime $p \in Prime(G)$ the normal subgroup $G^p$ (called the $coSylow$ $p$-subgroup of $G$):

\begin{verbatim}
coSylow := function(p)
    local ind, max_ind, Gp, h;
    max_ind := 1;
    Gp := Length(RCCSG);
    for h in LG do
        ind := Index(G, RCCSG[h];
        if IsInt(ind / p) and (max_ind < ind) then
            max_ind := ind;
            Gp := h;
        fi;
    od;
    return Gp;
end;
\end{verbatim}

Program 2.7.

The set $CoSylow(G) = \{(p, G^p) \mid p \in Prime(G)\} (= CoSylowG)$ is obtained by the function $\text{makeCoSylow}()$:

\begin{verbatim}
makeCoSylow := function()
    local S, n, i;
    S := [];
    n := Length(PrimeG);
    for i in [1..n] do
        S[i] := [PrimeG[i], coSylow(PrimeG[i])];
    od;
    return S;
end;

CoSylowG := makeCoSylow();
\end{verbatim}

Program 2.8.

Result 2.9. If $G = S_5$ then $CoSylow(G) = \{(2, RCCSG[18]), (3, RCCSG[19]), (5, RCCSG[19])\}$.

We compute $P2S(G)_{\text{odd}} (= P2SG_{\text{odd}})$ as follows. The function $\text{subgProduct}(-,-)$ defined below assigns a subgroup $HN$ to a subgroup $H$ and a normal subgroup $N$ of $G$.
\end{verbatim}
subgProduct := function(H, N)
    local gen;
    gen := Union(H.generators, N.generators);
    return Subgroup(G, gen);
end;

Program 2.10.

We also use $P(G) (= PG)$ in our computation of $P2S(G)_{\text{odd}}$, and the function `makePG()` computes $P(G)$.

makePG := function()
    local pg, i, ns, size;
    pg := [];
    ns := Length(RCCSG);
    for i in [1 .. ns] do
        size := Size(RCCSG[i]);
        if IsPrimePowerInt(size) or (size = 1) then
            Add(pg, i);
        fi;
    od;
    return pg;
end;
PG := makePG();

Program 2.11.

If RCCSG[i] is in PG, we check whether $(\text{RCCSG[i]}, \text{RCCSG[j]})$ is in $P2S(G) (= P2SG)$ or not, with the function `testP2SG(-, -)`:

testP2SG := function(i, j)
    local P, H, gsize, pair, p, Gp, K1, K2;
    P := RCCSG[i];
    H := RCCSG[j];
    if not (Size(H) / Size(P) = 2) then
        return false;
    fi;
    if isSubConjugate(j, i) = false then
        return false;
    fi;
    gsize := Size(G);
    for pair in CoSylowG do
        p := pair[1];
        Gp := RCCSG[pair[2]];
        K1 := subgProduct(P, Gp);
        if p = 2 then
            K2 := subgProduct(H, Gp);
            if not (Index(K2, K1) = 2) then
                return false;
            fi;
        fi;
    od;
    return true;
end;
else
    if not (Size(K1) = gsize) then
        return false;
    fi;
fi;
end;

Program 2.12.

We can obtain the list $\mathcal{P}2S(G)$ using the function $\text{makeP2SG}()$:  

\[
\text{makeP2SG} := \text{function}()
\]

local S, np, ns, i, j;
S := [];
np := Length(PG);
ns := Length(RCCSG);
for i in [1..np] do
    for j in [1..ns] do
        if testP2SG(PG[i], j) then
            Add(S, [PG[i], j]);
        fi;
    od;
od;
return S;
end;

Program 2.13.

Result 2.14. If $G = S_5$ then one obtains the result:

\[
\mathcal{P}2S(G) = \{ (RCCSG[1], RCCSG[2]), (RCCSG[3], RCCSG[6]), (RCCSG[3], RCCSG[7]), (RCCSG[4], RCCSG[9]), (RCCSG[4], RCCSG[11]), (RCCSG[5], RCCSG[12]) \}.
\]

The set $\mathcal{P}2S(G)_{\text{odd}}$ is computed by the function $\text{makeP2SGodd}()$:

\[
\text{makeP2SGodd} := \text{function()}
\]

local S, n, i;
S := [];
n := Length(P2SG);
for i in [1..n] do
    if not IsInt(P2SG[i][1] / 2) then
        Add(S, P2SG[i]);
    fi;
od;
return S;
end;

P2SGodd := makeP2SGodd();

Program 2.15.

Result 2.16. If \( G = S_5 \) then one obtains the result:

\[
P2S(G)_{\text{odd}} = \{ (RCCSG[1], RCCSG[2]), (RCCSG[3], RCCSG[6]), (RCCSG[3], RCCSG[7]), (RCCSG[5], RCCSG[12]) \}.
\]

3. H-Fixed point dimensions of irreducible \( G \)-representations

The built-in function \texttt{CharTable(-)} gives the character table of irreducible representations. Before using this function, we must set \texttt{G.conjugacyClasses}.

The character table will be obtained in the order of \texttt{G.conjugacyClasses}. Set

\[
\texttt{G.conjugacyClasses} := \texttt{ConjugacyClasses(G)};
\]

Result 3.1. If \( G = S_5 \) then one obtains the result:

\[
c_1 = \texttt{G.conjugacyClasses[1]} = \texttt{ConjugacyClasses(G, \{\})},
c_2 = \texttt{G.conjugacyClasses[2]} = \texttt{ConjugacyClasses(G, \{4,5\})},
c_3 = \texttt{G.conjugacyClasses[3]} = \texttt{ConjugacyClasses(G, \{3,4,5\})},
c_4 = \texttt{G.conjugacyClasses[4]} = \texttt{ConjugacyClasses(G, \{2,3\}(4,5))},
c_5 = \texttt{G.conjugacyClasses[5]} = \texttt{ConjugacyClasses(G, \{2,3,4,5\})},
c_6 = \texttt{G.conjugacyClasses[6]} = \texttt{ConjugacyClasses(G, \{1,2\}(3,4,5))},
c_7 = \texttt{G.conjugacyClasses[7]} = \texttt{ConjugacyClasses(G, \{1,2,3,4,5\})}.
\]

Next apply the function:

\[
\texttt{CTG := CharTable(G)};
\]

The irreducible character table is tabulated by \texttt{CTG.irreducibles} from the data \texttt{CTG}, and the value of the \( i \)-th irreducible character on the \( j \)-th conjugacy class is given by \texttt{CTG.irreducibles[i][j]}.

Result 3.2. If \( G = S_5 \) then one obtains the result:

\[
\begin{array}{cccccccc}
\chi_1 &= \texttt{CTG.irreducibles[1]} & 1 & 1 & 1 & 1 & 1 & 1 \\
\chi_2 &= \texttt{CTG.irreducibles[2]} & 1 & -1 & 1 & -1 & 1 & 1 \\
\chi_3 &= \texttt{CTG.irreducibles[3]} & 4 & -2 & 1 & 0 & 1 & -1 \\
\chi_4 &= \texttt{CTG.irreducibles[4]} & 4 & 2 & 1 & 0 & -1 & -1 \\
\chi_5 &= \texttt{CTG.irreducibles[5]} & 5 & 1 & -1 & 1 & 1 & 0 \\
\chi_6 &= \texttt{CTG.irreducibles[6]} & 5 & -1 & -1 & 1 & 1 & 0 \\
\chi_7 &= \texttt{CTG.irreducibles[7]} & 6 & 0 & 0 & -2 & 0 & 0 & 1 \\
\end{array}
\]

<table>
<thead>
<tr>
<th>conjugacy classes</th>
<th>( c_1 )</th>
<th>( c_2 )</th>
<th>( c_3 )</th>
<th>( c_4 )</th>
<th>( c_5 )</th>
<th>( c_6 )</th>
<th>( c_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \chi_1 )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( \chi_2 )</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( \chi_3 )</td>
<td>4</td>
<td>-2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>( \chi_4 )</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>( \chi_5 )</td>
<td>5</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \chi_6 )</td>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \chi_7 )</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.2 : Irreducible Characters of \( S_5 \)

In order to regard the data in \texttt{CTG} of a irreducible character as a function from \( G \) to the complex number field, we prepare the function \texttt{irrCharacter(-, -)}. This function assigns \( \chi_j(x) \) to the \( j \)-th irreducible character \( \chi_j \) and \( x \in G \).
irrCharacter := function(j, x)
        local k, n;
        n := Length(CTG.irreducibles);
        for k in [1 .. n] do
            if x in G.conjugacyClasses[k] then
                return CTG.irreducibles[j][k];
            fi;
        od;
    end;

Program 3.3.

Let $V$ be a complex $G$-representation. The dimension $\dim_{\mathbb{C}} V^H$ of $H$-fixed point set $V^H$ is calculated with the formula

$$\dim_{\mathbb{C}} V^H = \frac{1}{|H|} \sum_{h \in H} \chi_V(h),$$

where $\chi_V$ is the character of $G$, canonically identified with $V$. We give the function `fixedDim(-, -)` assigning $\dim_{\mathbb{C}} V^H$ to the $i$-th subgroup $H = \text{RCCSG}[i]$ in $	ext{RCCSG}$ and the $j$-th irreducible character $V = \text{CTG.irreducibles}[j]$ of $G$ by

`fixedDim := function(i, j)
    local h, x, s, d;
    if (i = Length(RCCSG)) then
        if (j = 1) then
            return 1;
        else
            return 0;
        fi;
    fi;
    h := RCCSG[i];
    s := Size(h);
    d := Sum(Elements(h), x -> irrCharacter(j, x)) / s;
    return d;
end;`

Program 3.4.

Now we make the table FDT of the fixed dimensions. For a subgroup RCCSG[i], FDT[i] is a list of the fixed dimension of the $j$-th irreducible representation by the $i$-th subgroup.

`FDT[i] := List([1 .. n], j -> fixedDim(i, j));`

**Result 3.5.** If $G = S_5$ then one obtains the result:
In order to obtain such FDT, we give the function `makeFixedDimTable()`: 

```plaintext
makeFixedDimTable := function()
    local S, nr, ni, i, j;
    S := [];
    nr := Length(RCCSG);
    ni := Length(CTG.irreducibles);
    for i in [1 .. nr] do
        S[i] := List([1 .. ni], j -> fixedDim(i, j))
    od;
    return S;
end;

FDT := makeFixedDimTable();
```

Program 3.7.

Let $\text{Irr}(G)$ denote the set of all isomorphism classes of irreducible complex $G$-representations. A complete set of representatives of $\text{Irr}(G)$ is denoted by $R\text{Irr}(G)$. The set $R\text{Irr}(G)$ is identified with the set of all irreducible characters of $G$. Let $\text{Irr}(G, \mathcal{M}(G))$ be the set of all isomorphism classes of irreducible complex $G$-representations $V$ such that $V^H = 0$ for all $H \in \mathcal{L}(G)$. Let $R\text{Irr}(G, \mathcal{M}(G))$ be a complete set of representatives of $\text{Irr}(G, \mathcal{M}(G))$. The next function `testIrrMG(-)` tells whether an irreducible $G$-representation belongs to $\text{Irr}(G, \mathcal{M}(G))$ or not.

```plaintext
testIrrMG := function(i)
    local j;
    for j in LG do
        if not (FDT[j][i] = 0) then
            return true;
        fi;
    od;
    return false;
end;
```

In order to obtain such FDT, we give the function `makeFixedDimTable()`: 

Table 3.6 : $S_5$-Fixed Dimensions

<table>
<thead>
<tr>
<th>$V_1$</th>
<th>$V_2$</th>
<th>$V_3$</th>
<th>$V_4$</th>
<th>$V_5$</th>
<th>$V_6$</th>
<th>$V_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCCSG[1]</td>
<td>1 1 4 4 5 5 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[2]</td>
<td>1 0 1 3 3 2 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[3]</td>
<td>1 1 2 2 3 3 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[4]</td>
<td>1 1 2 2 1 1 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[5]</td>
<td>1 1 1 1 2 2 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[6]</td>
<td>1 0 1 1 1 2 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[7]</td>
<td>1 0 0 2 2 1 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[8]</td>
<td>1 1 0 0 1 1 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[9]</td>
<td>1 0 0 2 1 0 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[10]</td>
<td>1 1 1 1 1 1 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[11]</td>
<td>1 0 1 1 1 0 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[12]</td>
<td>1 0 0 1 1 1 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[13]</td>
<td>1 1 0 0 1 1 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[14]</td>
<td>1 1 1 1 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[15]</td>
<td>1 0 0 1 1 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[16]</td>
<td>1 0 0 0 0 1 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[17]</td>
<td>1 0 0 1 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[18]</td>
<td>1 1 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCCSG[19]</td>
<td>1 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Let $\text{Irr}(G)$ denote the set of all isomorphism classes of irreducible complex $G$-representations. A complete set of representatives of $\text{Irr}(G)$ is denoted by $R\text{Irr}(G)$. The set $R\text{Irr}(G)$ is identified with the set of all irreducible characters of $G$. Let $\text{Irr}(G, \mathcal{M}(G))$ be the set of all isomorphism classes of irreducible complex $G$-representations $V$ such that $V^H = 0$ for all $H \in \mathcal{L}(G)$. Let $R\text{Irr}(G, \mathcal{M}(G))$ be a complete set of representatives of $\text{Irr}(G, \mathcal{M}(G))$. The next function `testIrrMG(-)` tells whether an irreducible $G$-representation belongs to $\text{Irr}(G, \mathcal{M}(G))$ or not.
return false;
fi;
ods;
return true;
end;

Program 3.8.

A set \( R_{Irr}(G, M(G)) \) is obtained by the function:

\[
\text{makeR}_{Irr}MG := \text{function}()
\]
\[
\quad \text{local } S, i, n; \\
\quad S := []; \\
\quad n := \text{Length}(\text{CTG.irreducibles}); \\
\quad \text{for } i \text{ in } [1..n] \text{ do} \\
\quad \quad \text{if } \text{testIr}rMG(i) \text{ then} \\
\quad \quad \quad \text{Add}(S, i); \\
\quad \quad \fi; \\
\quad \ods;
\]

Program 3.9.

\textbf{Result 3.10.} If \( G = S_5 \) then one obtains the result: \( R_{Irr}(G, M(G)) = \{ V_3, V_4, V_5, V_6, V_7 \} \).

Two irreducible complex \( G \)-representations \( V \) and \( W \) are said to be \textit{Galois conjugate} if \( \dim_C V^H = \dim_C W^H \) for all subgroups \( H \) of \( G \). Let \( G\text{CCIIrr}(G, M(G)) \) be the set of all Galois conjugate classes of representations in \( \text{Irr}(G, M(G)) \), and let \( R\text{GCCIIrr}(G, M(G)) \) be a complete set of representatives of \( G\text{CCIIrr}(G, M(G)) \). The next function \text{testGaloisConjugate}(\cdot, \cdot) \) checks whether, given a set \( S \) of irreducible representations, a irreducible representation is Galois conjugate to an element in \( S \) or not.

\[
\text{testGaloisConjugate} := \text{function}(\text{Ir}rs, i) \\
\quad \text{local } n, j, k, s; \\
\quad n := \text{Length}((\text{RCCSG}); \\
\quad \text{for } j \text{ in } \text{Ir}rs \text{ do} \\
\quad \quad s := \text{Sum}([1..n], k -> \text{AbsInt}(\text{FDT}[k][i] - \text{FDT}[k][j])); \\
\quad \quad \text{if } (s = 0) \text{ then} \\
\quad \quad \quad \text{return true}; \\
\quad \quad \fi; \\
\quad \ods; \\
\quad \text{return false};
\]

Program 3.11.

We can find a set \( R\text{GCCIIrr}(G, M(G)) \) by the next function:

\[
\text{makeRGaloisCCIIrrMG} := \text{function}() \\
\quad \text{local } a, i, j, k, gcc; \\
\quad gcc := [\text{R}\text{IrrMG}[1]]; \\
\quad \text{for } i \text{ in } \text{R}\text{IrrMG} \text{ do }
\]
if not testGaloisConjugate(gcc, i) then
    Add(gcc, i);
fi;

od;

return gcc;
end;

RGCCIrrMG := makeRGaloisCCirrMG();

Program 3.12.

Result 3.13. If \( G = S_5 \) then one obtains \( RGCCIrr(G, M(G)) = \{ V_3, V_4, V_5, V_6, V_7 \} \).

The function \( fixedDimDiff(\cdot, \cdot) \) below assigns to Pairs (a set consisting of pairs \((H, K)\) of subgroups of \( G \)) and a set \( Irrs \) of irreducible representations, the list of \( \dim_C V^H - 2 \dim_C V^K \), where \((H, K)\) runs over Pairs and \( V \) does over \( Irrs \).

\[
\text{fixedDimDiff} := \text{function(Pairs, Irrs)} \\
\text{local S, pair, h, k, i, b;} \\
S := [] ; \\
\text{for pair in Pairs do} \\
    h := pair[1] ; \\
    k := pair[2] ; \\
    b := \text{List(Irrs, i -> FDT[h][i] - 2 * FDT[k][i])}; \\
    \text{Add(S, b);} \\
\text{od;} \\
\text{return S;} \\
end;
\]

Program 3.14.

Result 3.15. If \( G = S_5 \) then by \( fixedDimDiff(P2SG, RGCCIrrMG) \), one obtains the result:

<table>
<thead>
<tr>
<th>irreducible modules</th>
<th>( V_3 )</th>
<th>( V_4 )</th>
<th>( V_5 )</th>
<th>( V_6 )</th>
<th>( V_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>((RCCSG[1], RCCSG[2]))</td>
<td>2</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>((RCCSG[3], RCCSG[6]))</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>((RCCSG[3], RCCSG[7]))</td>
<td>2</td>
<td>-2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>((RCCSG[4], RCCSG[9]))</td>
<td>2</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>((RCCSG[4], RCCSG[11]))</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>((RCCSG[5], RCCSG[12]))</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.16 : Differences of \( S_5 \)-Fixed Dimensions

4. Proof of Theorem 1.2

Let \( G = S_5 \). If \( V \) is a real \( G \)-module satisfying the gap condition then the complex module \( C \otimes_R V \) satisfies the gap condition with respect to complex dimension. That is, the function

\[
f_V(P, H) = \dim_C V^P - 2 \dim_C V^H
\]
is positive for all $P \in \mathcal{P}(G)$ and $H > P$, and in addition, $\dim_{G} V^{K} = 0$ for all $K \in \mathcal{L}(G)$. Suppose that there exists a complex $G$-module satisfying the gap condition. Replacing each irreducible summand by a Galois conjugate module in $RGCC Irr(G, M(G))$, we obtain a complex $G$-module

$$V = a_{3}V_{3} \oplus a_{4}V_{4} \oplus a_{5}V_{5} \oplus a_{6}V_{6} \oplus a_{7}V_{7},$$

where $a_{i}$ are nonnegative integers, satisfying the gap condition. Since

$$f_{V}(P, H) > 0 \quad \text{for} \quad (P, H) = (RCCS[G][3], RCCS[G][6]) \quad \text{and} \quad (RCCS[G][4], RCCS[G][11]),$$

it follows from Table 3.16 that $a_{5} - a_{6} > 0$ and $-a_{5} + a_{6} > 0$. This is a contradiction. Thus there never exists a real $G$-module satisfying the gap condition if $G = S_{5}$.

REFERENCES


