

УДК 512.542

**ДОПОЛНИТЕЛЬНЫЙ МАТЕРИАЛ К СТАТЬЕ
“О ПЕРЕСЕЧЕНИЯХ π -ХОЛЛОВЫХ ПОДГРУПП НЕКОТОРЫХ D_π -ГРУПП”**

И. Н. Белоусов, В. И. Зенков

Приведено в соответствии с просьбой разработчиков GAP (см. главную страницу сайта и <http://www.gap-system.org/Contacts/cite.html>).

1. Функция *OrbHall*

Функция $OrbHall(G, \pi)$ реализует алгоритм для нахождения количества всех регулярных орбит под действием π -холловой подгруппы H группы G на множестве всех π -холловых подгрупп сопряженных с H в группе G таких, что их пересечение с H равно 1. Данная функция и все последующие написаны в системе GAP 4.12.1 с использованием пакета "AtlasRep 2.1.6". Для функции $OrbHall(G, \pi)$ предполагается, что G изоморфна группе автоморфизмов простой спорадической группы.

```
OrbHall:=function(IdG, pi)
  local G,H,Cl_H, Intersect;

  if IdG in ["M11","M23","M24", "Co1", "Co2", "Co3",
    "Th", "Fi23", "Fi24", "B", "M", "J1", "Ru", "J4", "Ly", "B", "M"] then
    G:=SimpleGroup(IdG);
  fi;
  if IdG in ["M12","M22", "HS", "J2", "McL", "Suz", "He", "HN", "Fi22", "ON",
    "J3"] then
    G:=AutomorphismGroup(SimpleGroup(IdG));
  fi;

  H:=HallSubgroup( G, pi );
  Cl_H:=ConjugateSubgroups(G,H);
  Intersect:=Filtered(Cl_H, x -> Order(Intersection(x,H))=1 );;
  return Size(Intersect)/Order(H);
end;
```

Функция $OrbHall(G, \pi)$ выбирает некоторую π -холлову подгруппу H в $Aut(G)$ с помощью функции $HallSubgroup(G, P)$ ¹. Далее находится класс сопряженных с H в G подгрупп Cl_H также с помощью встроенной в GAP функцией $ConjugateSubgroups(G, H)$. Из класса Cl_H выбирается множество $Intersect$ всех подгрупп, пересекающихся с H по единице. Функция выводит результат отношения числа подгрупп во множестве $Intersect$ к порядку подгруппы H .

Результат работы описанной функции для небольших спорадических групп и $\pi = \{2\}$ или $\{3\}$ выписан в следующей таблице.

¹<https://docs.gap-system.org/doc/ref/chap39.html#X7EDBA19E828CD584>
(см. также <https://docs.gap-system.org/doc/ref/chap39.html#X8716635F7951801B>)

G	$OrbHall(G, \{2\})$	$OrbHall(G, \{3\})$
M_{11}	27	6
M_{12}	6	31
M_{22}	5	683
M_{23}	522	7867
M_{24}	163	41956
HS	39	17107
J_1	127	975
J_2	10	102

При запуске функции $OrbHall$ с группами $Co_1, Co_2, Co_3, McL, Suz, He, HN, Th, Fi_{22}, Fi_{23}, Fi'_{24}, B, M, O'N, J_3, Ru, J_4, Ly$ не хватило вычислительных мощностей персонального компьютера с процессором *Core-i7 4770 3.4.GHz* и *16GB* оперативной памяти, используемого авторами.

2. Нахождение нижней границы для $Orb_p(G)$, используя централизатор центра силовской p -подгруппы

Пусть G — группа, изоморфная группе автоморфизмов простой спорадической группы, H — ее силовская p -подгруппа и $C = C_G(Z(H))$. Если найдется элемент g из силовской q -подгруппы A такой, что $C \cap H^g = 1$, то $Orb_p(G) \geq |C : N_C(H^g)|/|H|$.

Функция $IntersectOrders$ находит все порядки пересечений $C \cap H^g$, где g пробегает все элементы подгруппы A , выводит массив этих порядков и, если существует элемент g с указанным в предыдущем абзаце свойством, выводит значение $|C|/|N_G(H)|$. Иначе выдает -1 . Подгруппы H и A выбираются с помощью встроенной в GAP функции $SylowSubgroup(G, p)$.

```
IntersectOrders:=function(IdG, p, q)
  local G,H, A, C, orbH, ListOrderIntersectH;

  if IdG in ["M11","M23","M24", "Co1", "Co2", "Co3",
  "Th", "Fi23", "Fi24", "B", "M", "J1", "Ru", "J4", "Ly", "B", "M"] then
    G:=AtlasGroup(IdG);
  fi;

  if IdG in ["M12","M22", "HS", "J2", "McL", "Suz", "He", "HN", "Fi22",
  "ON", "J3"] then
    G:=AtlasGroup(Concatenation(IdG, ".2"));
  fi;

  H:=SylowSubgroup(G,p);
  A:=SylowSubgroup(G,q);
  C:=Centralizer(G,Center(H));
  orbH:=H^A;
  ListOrderIntersectH:=List(orbH, x-> Order(Intersection(x,C)));
  if 1 in ListOrderIntersectH then
    Print(ListOrderIntersectH, "\n");
    return Order(C)/(Order(Normalizer(C,orbH[Position(
    ListOrderIntersectH,1)]))*Order(H));
  else
    Print(ListOrderIntersectH, "\n");
    return -1;
  fi;
fi;
```

end;

Результат выполнения программы для разных групп G и простых чисел p и q представлен ниже.

```
gap> IntersectOrders("M11",2,11);
[ 16, 2, 1, 2, 1, 1, 1, 1, 2, 1, 2 ]
3
```

```
gap> IntersectOrders("M12",2,11);
[ 128, 4, 1, 2, 2, 2, 4, 2, 2, 2, 2 ]
3
```

```
gap> IntersectOrders("M22",2,11);
[ 256, 2, 8, 4, 2, 1, 1, 4, 4, 16, 2 ]
3
```

```
gap> IntersectOrders("M23",2,23);
[ 128, 1, 1, 1, 1, 16, 8, 1, 1, 1, 8, 1, 1, 1, 1, 1, 1, 4, 1, 1, 1, 2 ]
21
```

```
gap> IntersectOrders("M24",2,23);
[ 1024, 2, 1, 2, 1, 2, 1, 1, 2, 1, 32, 4, 2, 32, 4, 8, 1, 1, 1, 4, 2, 4, 4 ]
21
```

```
gap> IntersectOrders("HS",2,11);
[ 1024, 2, 2, 4, 4, 2, 2, 2, 2, 2, 2 ]
-1
```

```
gap> IntersectOrders("J2",2,7);
[ 256, 8, 2, 2, 2, 2, 8 ]
-1
```

```
gap> IntersectOrders("Co1",2,23);
[ 2097152, 4, 1, 4, 32, 2, 2, 1, 32, 16, 2, 2, 8, 1, 2, 128, 4, 4, 4, 32, 4, 1, 4 ]
42525
```

```
gap> IntersectOrders("Co2",2,23);
[ 262144, 32, 32, 64, 128, 64, 32, 32, 256, 64, 64, 32, 32, 16, 256, 64, 64, 256,
32, 64, 64, 16, 32 ]
-1
```

```
gap> IntersectOrders("Co2",2,11);
[ 262144, 64, 128, 64, 32, 32, 16, 16, 64, 32, 64 ]
-1
```

```
gap> IntersectOrders("Co2",2,7);
[ 262144, 64, 32, 64, 128, 64, 16 ]
-1
```

```
gap> IntersectOrders("Co3",2,23);  
[ 1024, 1, 1, 1, 2, 1, 1, 1, 2, 2, 1, 4, 2, 1, 2, 2, 2, 1, 1, 1, 1, 1, 2 ]  
2835
```

```
gap> IntersectOrders("McL",2,11);  
[ 256, 2, 2, 2, 1, 2, 1, 1, 2, 1, 1 ]  
315
```

```
gap> IntersectOrders("Suz",2,13);  
[ 16384, 1, 8, 1, 1, 4, 1, 2, 16, 2, 1, 8, 2 ]  
405
```

```
gap> IntersectOrders("He",2,17);  
[ 2048, 1, 1, 4, 1, 2, 1, 4, 1, 1, 2, 1, 1, 2, 8, 1, 1 ]  
21
```

```
gap> IntersectOrders("HN",2,19);  
[ 32768, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
225
```

```
gap> IntersectOrders("Fi22",2,13);  
[ 262144, 8, 8, 8, 8, 4, 32, 16, 8, 4, 8, 8, 16 ]  
-1
```

```
gap> IntersectOrders("Fi22",2,11);  
[ 262144, 16, 8, 4, 8, 8, 8, 8, 8, 4, 2 ]  
-1
```

```
gap> IntersectOrders("Fi22",2,7);  
[ 262144, 8, 8, 64, 128, 2, 8 ]  
-1
```

```
gap> IntersectOrders("Fi23",2,17);  
[ 262144, 1, 2, 1, 2, 1, 2, 1, 1, 1, 1, 2, 1, 4, 1, 1, 1 ]  
405
```

```
gap> IntersectOrders("Fi24",2,17);  
[ 4194304, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1 ]  
76545
```

```
gap> IntersectOrders("J1",2,19);  
[ 8, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
1
```

```
gap> IntersectOrders("J1",2,11);  
[ 8, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
1
```

```
gap> IntersectOrders("J1",2,7);  
[ 8, 1, 1, 1, 1, 1, 1 ]  
1
```

```
gap> IntersectOrders("ON",2,31);  
[ 1024, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1 ]  
315
```

```
gap> IntersectOrders("J3",2,19);  
[ 256, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
15
```

```
gap> IntersectOrders("Ru",2,13);  
[ 16384, 2, 2, 1, 1, 2, 2, 1, 2, 1, 1, 2, 1 ]  
15
```

```
gap> IntersectOrders("M11",3,11);  
[ 9, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
1
```

```
gap> IntersectOrders("M12",3,11);  
[ 27, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
4
```

```
gap> IntersectOrders("M22",3,11);  
[ 9, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
1
```

```
gap> IntersectOrders("M23",3,11);  
[ 9, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
1
```

```
gap> IntersectOrders("M24",3,11);  
[ 27, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
20
```

```
gap> IntersectOrders("HS",3,11);  
[ 9, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
4
```

```
gap> IntersectOrders("J2",3,7);  
[ 27, 1, 1, 1, 1, 1, 1 ]  
10
```

```
gap> IntersectOrders("Co3",3,23);  
[ 2187, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
160
```

```
gap> IntersectOrders("Co2",3,23);  
[ 729, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
640
```

```
gap> IntersectOrders("McL",3,11);
```

```
[ 729, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

```
80
```

```
gap> IntersectOrders("Suz",3,13);
```

```
[ 2187, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

```
16
```

```
gap> IntersectOrders("He",3,17);
```

```
[ 27, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

```
560
```

```
gap> IntersectOrders("Fi23",3,23);
```

```
[ 1594323, 9, 1, 1, 1, 1, 3, 3, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 81 ]
```

```
1024
```

```
gap> IntersectOrders("HN",3,19);
```

```
[ 729, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

```
80
```

```
gap> IntersectOrders("Fi22",3,13);
```

```
[ 19683, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

```
256
```

```
gap> IntersectOrders("Fi23",3,13);
```

```
[ 1594323, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

```
1024
```

```
gap> IntersectOrders("Fi24",3,13);
```

```
[ 43046721, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 3, 1 ]
```

```
56320
```

```
gap> IntersectOrders("J1",3,19);
```

```
[ 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

```
10
```

```
gap> IntersectOrders("ON",3,31);
```

```
[ 81, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

```
1
```

```
gap> IntersectOrders("J3",3,19);
```

```
[ 243, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

```
1
```

```
gap> IntersectOrders("J3",3,17);
```

```
[ 243, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

```
1
```

```
gap> IntersectOrders("J3",3,5);
```

```
[ 243, 1, 1, 1, 1 ]
```

```
1
```

```
gap> IntersectOrders("Ru",3,29);  
[ 27, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1, 1 ]  
80
```