The GAP package *Prescribed Automorphism Groups*[*]

Vedran Krčadinac

PMF-MO

4.5.2022.

* This work was fully supported by the Croatian Science Foundation under the project 9752.

The scientific project ACCO

Research objectives:

O1. Development of algorithmic methods for the construction and classification of combinatorial objects with strong algebraic structure. These methods utilise known algebraic and combinatorial properties of the objects to handle larger parameters and problems that have been out of reach with traditional construction methods.

O2. Widening of theoretical knowledge about combinatorial objects that are the topic of research. Interesting theorems are often discovered and proved on the basis of available examples. It is expected that the results of the project will lead to such discoveries.

O3. **Development of a software package, implemented in GAP, for the construction and analysis of combinatorial objects.**

PAG

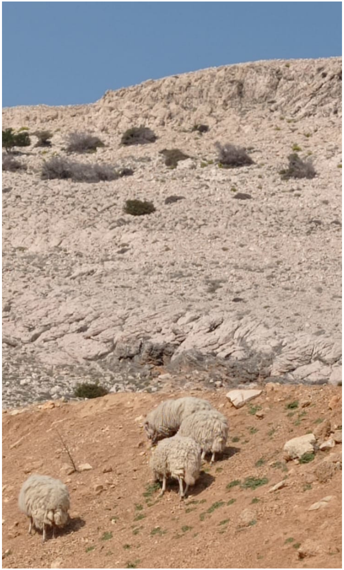
Prescribed Automorphism Groups

0.1

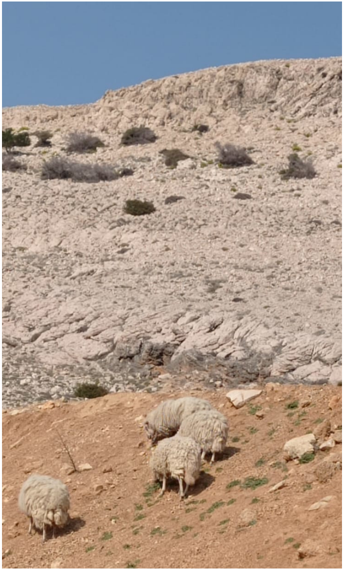
The PAG package

The PAG package

The PAG package

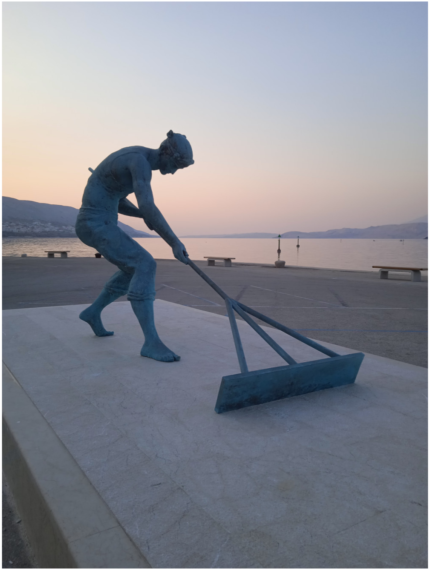


The PAG package

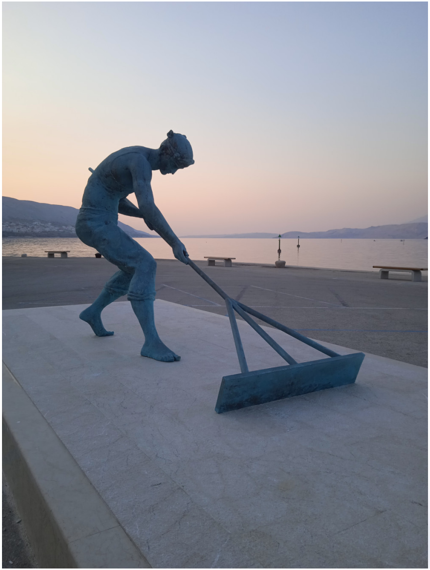


The PAG package

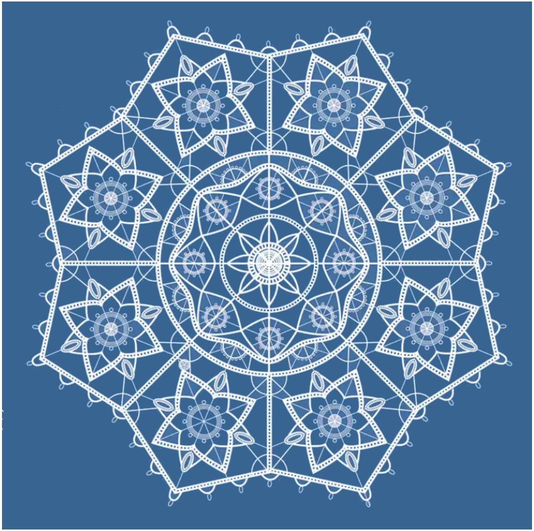
The PAG package



The PAG package



The PAG package



Why prescribe automorphism groups?

Why prescribe automorphism groups?

A t-(v, k, λ) design consists of a v-element set V of points and a collection $\mathcal{B}$ of k-subsets of V called blocks such that every t-subset of V is contained in exactly λ blocks.

Why prescribe automorphism groups?

A t-(v, k, λ) design consists of a v-element set V of points and a collection $\mathcal{B}$ of k-subsets of V called blocks such that every t-subset of V is contained in exactly λ blocks.

The total number of blocks is denoted by $b = |\mathcal{B}|$ and is determined by the other parameters.

Why prescribe automorphism groups?

A t-(v, k, λ) design consists of a v-element set V of points and a collection $\mathcal{B}$ of k-subsets of V called blocks such that every t-subset of V is contained in exactly λ blocks.

The total number of blocks is denoted by $b = |\mathcal{B}|$ and is determined by the other parameters.

An automorphism of the design is a permutation of V preserving $\mathcal{B}$.

Why prescribe automorphism groups?

A t-(v, k, λ) design consists of a v-element set V of points and a collection $\mathcal{B}$ of k-subsets of V called blocks such that every t-subset of V is contained in exactly λ blocks.

The total number of blocks is denoted by $b = |\mathcal{B}|$ and is determined by the other parameters.

An automorphism of the design is a permutation of V preserving $\mathcal{B}$.

Does a 2-$(51, 6, 1)$ design with $b = 85$ blocks exist?

Why prescribe automorphism groups?

A t-(v, k, λ) design consists of a v-element set V of points and a collection $\mathcal{B}$ of k-subsets of V called blocks such that every t-subset of V is contained in exactly λ blocks.

The total number of blocks is denoted by $b = |\mathcal{B}|$ and is determined by the other parameters.

An automorphism of the design is a permutation of V preserving $\mathcal{B}$.

Does a 2-$(51, 6, 1)$ design with $b = 85$ blocks exist?

$$\binom{51}{6} \approx 18 \cdot 10^6$$

Why prescribe automorphism groups?

A t-(v,k,λ) design consists of a v-element set V of points and a collection $\mathcal{B}$ of k-subsets of V called blocks such that every t-subset of V is contained in exactly λ blocks.

The total number of blocks is denoted by $b = |\mathcal{B}|$ and is determined by the other parameters.

An automorphism of the design is a permutation of V preserving $\mathcal{B}$.

Does a 2-$(51,6,1)$ design with $b = 85$ blocks exist?

$$\binom{51}{6} \approx 18 \cdot 10^6 \qquad\qquad \binom{\binom{51}{6}}{85} \approx 1.8 \cdot 10^{488}$$

Why prescribe automorphism groups?

A t-(v,k,λ) design consists of a v-element set V of points and a collection $\mathcal{B}$ of k-subsets of V called blocks such that every t-subset of V is contained in exactly λ blocks.

The total number of blocks is denoted by $b = |\mathcal{B}|$ and is determined by the other parameters.

An automorphism of the design is a permutation of V preserving $\mathcal{B}$.

Does a 2-$(51,6,1)$ design with $b = 85$ blocks exist?

$$\binom{51}{6} \approx 18 \cdot 10^6 \qquad\qquad \binom{\binom{51}{6}}{85} \approx 1.8 \cdot 10^{488}$$

$$\binom{\binom{51}{6}/|G|}{85/|G|} \approx 10^{11} \quad \text{for } |G| = 40$$

The old package KMAD 0.7

kmad07.g - Notepad

File Edit Format View Help

```
#
#  KMAD 0.7:  Kramer-Mesner and decompositions
#
#  Vedran Krcadinac (krcko@math.hr) and Renata Vlahovic Kruc (vlahovic@math.hr)
#  Department of Mathematics, University of Zagreb
#  12.05.2021.
#
#  New features:
#
#  12/05/2021 Added replesize4.
#  05/08/2020 Added pgtest.
#  11/01/2020 Added minustriangle (for projective planes)
#  01/01/2020 Added testgama1, testgama2.
#  30/12/2019 Addes incfilter2, aut2, cliquesconf2.
#  25/12/2019 Added makeA5.
#  17/12/2019 Added fastorbrep.
#  24/11/2019 Added cliquesinc1.
#  04/10/2019 Modified incfilter.
#  26/09/2019 Added cliquesconf.
#  11/05/2019 Added forbrepint2, forbrepint4.
#  05/05/2019 Added mycliquer.
#  13/04/2019 Added AGL1.
#  25/03/2019 Repaired restrict.
#  07/01/2019 Enhanced cliquesdes with ordering options.
#  19/12/2018 Added qsindex.
#  10/12/2018 Added cliquesdes.
#  06/12/2018 Added dual, testdes2, blockint3, testdes3.
#  08/10/2018 Repaired forbrep8.
#  30/09/2018 Added tdmcliques.
```

The old package KMAD 0.7

kmad07.g - Notepad

File Edit Format View Help

```
#
# Permutations
#

#moveperm:=function(p,from,to)
#  return Sortex(Concatenation([1..Minimum(to)-1],Permuted(from,p)-Minimum(from)+Minimum(to)))^(-1);
#end;

#moveperm:=function(p,from,to)
#local a;
#a:=Product(List(TransposedMat([from,to]),
#function(x) if x[1]<>x[2] then return (x[1],x[2]); else return (); fi; end));
#p:=a^-1*p*a;
#return Sortex(Permuted(to,p));
#end;

moveperm:=function(p,from,to)
  local c,i;
  c:=();
  for i in [1..Length(from)] do
    c:=c*(from[i],to[i]);
  od;
  return c*p*c;
end;

unionperm:=function(plist,set)
  local p;
```

The old package KMAD 0.7

kmad07.g - Notepad

File Edit Format View Help

```
end;


#
#  Orbit generation
#

addorb := function(g, v, k, o)
  local seeds;
  if IsInt(v) then v:=[1..v]; fi;
  seeds:=AsSet(Concatenation(List(o[k-1],x->Set(Difference(v,x[1]),y->Union(x[1],[y])))));
  o[k]:=Orbits(g,seeds,OnSets);
end;

makeorb := function(g, v, k, o)
  local i;
  if IsInt(v) then v:=[1..v]; fi;
  o[1]:=Orbits(g,Combinations(v,1),OnSets);
  for i in [2..k] do
    addorb(g,v,i,o);
  od;
end;

addrep := function(g, v, k, o)
  local seeds;
  if IsInt(v) then v:=[1..v]; fi;
  seeds:=AsSet(Concatenation(List(o[k-1],x->Set(Difference(v,x),y->Union(x,[y])))));
  o[k]:=OrbitRepresentatives(g,seeds,OnSets);
end;
```

The old package KMAD 0.7

kmad07.g - Notepad

File Edit Format View Help

```
#
#  Tactical decomposition matrices
#

tdmat:=function(g,t,v,k,lambda,beta)
  local bol,b,nu,i,output,input,no,res;
  no:=tmpno();
  b:=lambda*Binomial(v,t)/Binomial(k,t);
  output:=OutputTextFile(Concatenation(tmpath,"tdmat-",no,".par"),false);
  WriteAll(output,Concatenation(String(t)," ",String(v)," ",
           String(k)," ",String(lambda),"\n"));
  if IsGroup(g) then nu:=SortedList(OrbitLengths(g,[1..v]));
    else nu:=g; fi;
  WriteLine(output,Concatenation(String(Size(nu))," ",String(Size(beta))));
  for i in nu do WriteAll(output,Concatenation(String(i)," ")); od;
  WriteAll(output,"\n");
  for i in beta do WriteAll(output,Concatenation(String(i)," ")); od;
  WriteAll(output,"\n");
  WriteLine(output,"?");
  CloseStream(output);
  input:=InputTextUser();
  output:=OutputTextFile(Concatenation(tmpath,"tdmat-",no,".tdm"),false);
  Process(Directory(path),Concatenation(path,"orbmat5qd"),
    input,output,[Concatenation(tmpath,"tdmat-",no,".par"),"-d"]);
  CloseStream(input);
  CloseStream(output);
```

The old package KMAD 0.7

kmad07.g - Notepad

File Edit Format View Help

```
#
# Orbits to Kramer-Mesner system
#

expand:=function(A,lambda)
  return TransposedMat(Concatenation(TransposedMat(A),[List([1..Size(A)],x->lambda)]));
end;

# Add last equation: orbit sizes sum up to b

addeq := function(A,g,bo,b)
 return Concatenation(A,[Concatenation(List(bo,x->Size(Orbit(g,x,OnSets))),[b])]);
end;

KMmat:=function(arg)
# First argument: group
# Second argument: t-orbits/representatives labelling the rows
# Third argument: k-orbits/representatives labelling the columns
# Fourth argument (optional): lambda
# Fifth argument (optional): b
  local mat;
  if Size(arg) in [3,4,5] then
    if IsList(arg[2][1][1]) then
      if IsList(arg[3][1][1]) then
        mat:=List(arg[2],z->List(arg[3],y->Sum(List(y,x->yesno(IsSubset(x,z[1]))))));
      else
        mat:=List(arg[2],z->List(arg[3],y->Sum(List(Orbit(arg[1],y,OnSets),x->yesno(IsSubset(x,z[1])))))
```

The old package KMAD 0.7

kmad07.g - Notepad

File Edit Format View Help

```
#
#  Solutions to designs
#

pick:=function ( x, y )
    if x = 1  then
        return y;
    else
        return [   ];
    fi;
    return;
end;

design:=function(g,ksub,sol)
  local tdm;
  tdm:=Size(ksub)=2;
  if tdm then
    tdm:=NestingDepthA(ksub[1])<>NestingDepthA(ksub[2]);
  fi;
  if tdm then
    ksub:=Concatenation(ksub[1]);
  fi;
  if NestingDepthA(ksub)=2 then
    ksub:=List(ksub,x->Orbit(g,x,OnSets));
  fi;
  return Concatenation(List([1..Size(ksub)],x->pick(sol[x],ksub[x])));
end;
```

The old package KMAD 0.7

kmad07.g - Notepad

File Edit Format View Help

```
#
# External programs
#

ssum:=function(a,s)
local input, output,no,res;
no:=tmpno();
output:=OutputTextFile(Concatenation(tmpath,"ssum-",no,".gin"),false);
PrintTo(output,s,"\n");
PrintTo(output,Size(a),"\n");
PrintTo(output,a);
CloseStream(output);
input:=InputTextFile(Concatenation(tmpath,"ssum-",no,".gin"));
output:=OutputTextFile(Concatenation(tmpath,"ssum-",no,".in"),false);
Process(Directory(tmpath),Concatenation(path,"gap2raw"),input,output,[]);
CloseStream(input);
CloseStream(output);
input:=InputTextFile(Concatenation(tmpath,"ssum-",no,".in"));
output:=OutputTextFile(Concatenation(tmpath,"ssum-",no,".out"),false);
Process(Directory(tmpath),Concatenation(path,"ssum"),input,output,[]);
CloseStream(input);
CloseStream(output);
res:=ReadAsFunction(Concatenation(tmpath,"ssum-",no,".out"))();
RemoveFile(Concatenation(tmpath,"ssum-",no,".gin"));
RemoveFile(Concatenation(tmpath,"ssum-",no,".in"));
RemoveFile(Concatenation(tmpath,"ssum-",no,".out"));
```

The old package KMAD 0.7

kmad07.g - Notepad

File Edit Format View Help

```
# Miscellaneous

minustriangle:=function(p,tri)
local a,b,c,A,B,C,pmin;
a:=tri[1];
b:=tri[2];
c:=tri[3];
A:=Filtered(p,x->c in x and b in x)[1];
B:=Filtered(p,x->a in x and c in x)[1];
C:=Filtered(p,x->a in x and b in x)[1];
pmin:=Union(A,B,C);
return List(Filtered(p,x->not a in x and not b in x and not c in x),y->Difference(y,pmin));
end;
```

The old package KMAD 0.7

kmad07.g - Notepad

File Edit Format View Help

```
# Miscellaneous

minustriangle:=function(p,tri)
local a,b,c,A,B,C,pmin;
a:=tri[1];
b:=tri[2];
c:=tri[3];
A:=Filtered(p,x->c in x and b in x)[1];
B:=Filtered(p,x->a in x and c in x)[1];
C:=Filtered(p,x->a in x and b in x)[1];
pmin:=Union(A,B,C);
return List(Filtered(p,x->not a in x and not b in x and not c in x),y->Difference(y,pmin));
end;
```

About 4000 lines of code.

The old package KMAD 0.7

kmad07.g - Notepad

File Edit Format View Help

```
# Miscellaneous

minustriangle:=function(p,tri)
local a,b,c,A,B,C,pmin;
a:=tri[1];
b:=tri[2];
c:=tri[3];
A:=Filtered(p,x->c in x and b in x)[1];
B:=Filtered(p,x->a in x and c in x)[1];
C:=Filtered(p,x->a in x and b in x)[1];
pmin:=Union(A,B,C);
return List(Filtered(p,x->not a in x and not b in x and not c in x),y->Difference(y,pmin));
end;
```

About 4000 lines of code. No documentation! 😱

PAG 0.1 – Manual and documentation

The PAG manual is available at:

https://web.math.pmf.unizg.hr/acco/publications.php

The PAG manual is available at:

https://web.math.pmf.unizg.hr/acco/publications.php

Contents

Chapter 1

The PAG Package

Prescribed Automorphism Groups (PAG) is a GAP package for constructing combinatorial objects with prescribed automorphism groups.

1.1 Getting Started

The package is loaded by

Example
```
gap> LoadPackage("PAG");
```

Let us present a small example from the paper [Krč18]. In Theorem 8.1, simple 5-(16,7,10) designs with the following automorphism group were constructed.

Example
```
gap> g:=Group((2,3,4)(5,6,7,8,9,10)(11,12,13,14,15,16),
> (1,5)(2,12)(3,15)(4,8)(6,14)(7,16)(9,10)(11,13));
```

They can be obtained by typing

Example
```
gap> KramerMesnerSearch(5,16,7,10,g);
```

Chapter 2

The PAG Functions

The following functions are available in the PAG package.

2.1 Working With Permutation Groups

2.1.1 CyclicPermutation

▷ `CyclicPermutation(n)` (function)

Returns the cyclic permutation (1,...,n).

2.1.2 PrimitiveGroupsOfDegree

▷ `PrimitiveGroupsOfDegree(v)` (function)

Returns a list of all primitive permutation gropus on v points.

PAG 0.1 – Manual and documentation

PAG.xml - Notepad

File Edit Format View Help

```
<!-- %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% -->

<Chapter Label="The PAG Package">
<Heading>The PAG Package</Heading>
<Index>PAG</Index>

<E>Prescribed Automorphism Groups</E> (&PAG;) is a &GAP; package
for constructing combinatorial objects with prescribed automorphism
groups.

<P/>

<Section Label="Getting Started">
<Heading>Getting Started</Heading>

The package is loaded by
<Example><![CDATA[gap> LoadPackage("PAG"); ]]></Example>
Let us present a small example from the paper <Cite Key='VK18'/>.
In Theorem 8.1, simple 5-(16,7,10) designs with the following
automorphism group were constructed.
<Example><![CDATA[gap> g:=Group((2,3,4)(5,6,7,8,9,10)(11,12,13,14,15,16),
> (1,5)(2,12)(3,15)(4,8)(6,14)(7,16)(9,10)(11,13));]]></Example>
They can be obtained by typing
<Example><![CDATA[gap> KramerMesnerSearch(5,16,7,10,g);
Computing t-subset orbit representatives...
28
Computing k-subset orbit representatives...
71
```

PAG 0.1 – Manual and documentation

PAG.gd - Notepad

File Edit Format View Help

```
##############################################################################
##
##
#W  PAG.gd          Prescribed Automorphism Groups (PAG)
#W                  by Vedran Krcadinac
##
##  Declarations and documentation for functions of the PAG package.
##
##


##############################################################################
##
#F  PrimitiveGroupsOfDegree( <v> )
##
##  <#GAPDoc Label="PrimitiveGroupsOfDegree">
##  <ManSection>
##  <Func Name="PrimitiveGroupsOfDegree" Arg="v"/>
##
##  <Description>
##  Returns a list of all primitive permutation gropus on <A>v</A> points.
##  </Description>
##  </ManSection>
##  <#/GAPDoc>
DeclareGlobalFunction( "PrimitiveGroupsOfDegree" );

##############################################################################
##
#F  CyclicPermutation( <n> )
```

GAP (PAG) - Chapter 1: The PAG Pac

file:///C:/Users/Vedran Krčadinac/Desktop/PAG-0.1/doc/chap1.html#X84BC1F5D7BD3CF55

Goto Chapter: Top 1 2 Bib Ind

[Top of Book] [Contents] [Previous Chapter] [Next Chapter]

[MathJax on] [Style]

1 The PAG Package

Prescribed Automorphism Groups (**PAG**) is a **GAP** package for constructing combinatorial objects with prescribed automorphism groups.

1.1 Getting Started

The package is loaded by

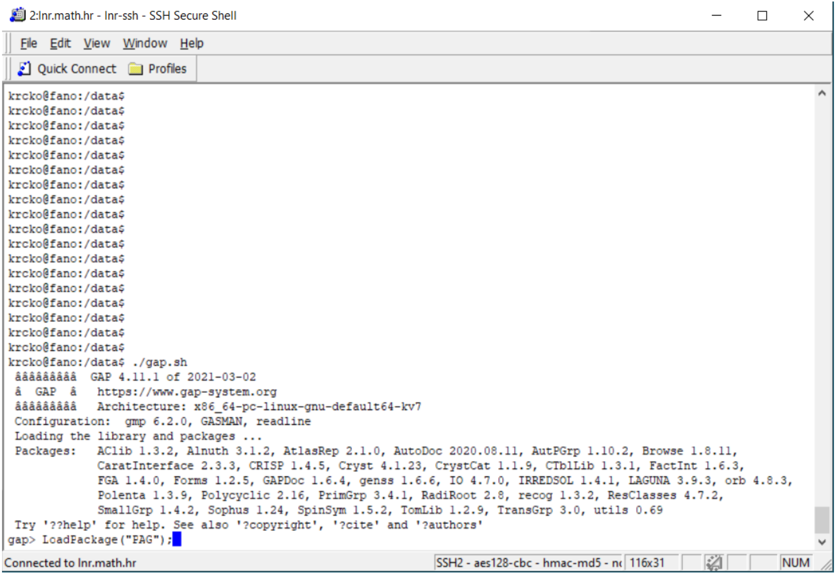
```
gap> LoadPackage("PAG");
```

Let us present a small example from the paper [Kr{č}18]. In Theorem 8.1, simple 5-(16,7,10) designs with the following automorphism group were constructed.

```
gap> g:=Group((2,3,4)(5,6,7,8,9,10)(11,12,13,14,15,16),
> (1,5)(2,12)(3,15)(4,8)(6,14)(7,16)(9,10)(11,13));
```

They can be obtained by typing

PAG 0.1 – Manual and documentation



PAG 0.1 – Manual and documentation

2:lnr.math.hr - lnr-ssh - SSH Secure Shell

File Edit View Window Help

Quick Connect Profiles

```
ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ
Loading  images 1.3.1 (Minimal and Canonical images)
by Christopher Jefferson (http://caj.host.cs.st-andrews.ac.uk/),
   Markus Pfeiffer (http://www.morphism.de/~markusp/),
   Rebecca Waldecker (http://conway1.mathematik.uni-halle.de/~waldecker/), and
   Eliza Jonauskyte (ej31@st-andrews.ac.uk).
maintained by:
   Christopher Jefferson (http://caj.host.cs.st-andrews.ac.uk/).
Homepage: https://gap-packages.github.io/images/
Report issues at https://github.com/gap-packages/images/issues
ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ
ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ
Loading  GRAPE 4.8.3 (GRaph Algorithms using PErmutation groups)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/grape
Report issues at https://github.com/gap-packages/grape/issues
ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ
ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ
Loading  DESIGN 1.7 (The Design Package for GAP)
by Leonard H. Soicher (http://www.maths.qmul.ac.uk/~lsoicher/).
Homepage: https://gap-packages.github.io/design
Report issues at https://github.com/gap-packages/design/issues
ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ
ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ
Loading  PAG 0.1 (Prescribed Automorphism Groups)
by Vedran Krcadinac (https://web.math.pmf.unizg.hr/~krcko/homepage.html).
Homepage: https://gap-packages.github.io/pag
Report issues at https://github.com/gap-packages/pag/issues
ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ
true
gap> ?KramerMesnerSearch
```

Connected to lnr.math.hr | SSH2 - aes128-cbc - hmac-md5 - n(| 116x31 | NUM

PAG 0.1 – Manual and documentation

2:lnr.math.hr - lnr-ssh - SSH Secure Shell

File Edit View Window Help

Quick Connect Profiles

```
2.3-1 KramerMesnerSearch

â£ KramerMesnerSearch( t, v, k, lambda[, opt] ) âââââââââââââââââââââââââââââââââââââââââââââââââââââââ function

Performs  a search for t-(v,k,lambda) designs with presrcribed automorphism group G by the Kramer-Mesner method.
A  record  with  options  can  be  supplied.  By  default,  a list of base blocks for the constructed designs is
returned.  If  opt.Design is defined, the designs are returned in the DESIGN package format 'DESIGN: Design'. If
opt.NonIsomorphic  is  defined,  the  designs are returned in DESIGN format and isomorph-rejection is performed.
Other available options are:

SmallLambda:=true/false.  Perform  the "small lambda filter", i.e. remove k-orbits covering some of the t-orbits
more than lambda times. By default, this is done if lambda<=3.

2.3-2 KramerMesnerMatrix

â£ KramerMesnerMatrix( G, tsub, ksub[, lambda][, b] ) ââââââââââââââââââââââââââââââââââââââââââââââââââ function

Returns the Kramer-Mesner matrix for a permutation group G. The rows are labelled by t-subset orbits represented
by  tsub,  and  the columns by k-subset orbits represented by ksub. A column of constants lambda is added if the
optional  argument  lambda  is  given. Another row is added if the optional argument b is given, repesenting the
constraint that sizes of the chosen k-subset orbits must sum up to the number of blocks b.

2.3-3 SolveKramerMesner

â£ SolveKramerMesner( mat ) ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ function

Solve a Kramer-Mesner system using A.Wassermann's LLL sovler solvediophant.

2.3-4 BaseBlocks

-- <space> page, <n> next line, <b> back, <p> back line, <q> quit --
```

Connected to lnr.math.hr | SSH2 - aes128-cbc - hmac-md5 - nc | 116x31 | NUM

PAG 0.1 – What is implemented so far?

The standard Kramer-Mesner method for t-designs:

- Generating G-orbits of k-subsets of V [GAP code]

PAG 0.1 – What is implemented so far?

The standard Kramer-Mesner method for t-designs:

- Generating G-orbits of k-subsets of V [GAP code]
  $\rightsquigarrow$ Orderly algorithm using GAP package **images**

PAG 0.1 – What is implemented so far?

The standard Kramer-Mesner method for t-designs:

- Generating G-orbits of k-subsets of V [GAP code]

  $\rightsquigarrow$ Orderly algorithm using GAP package **images**

  $\rightsquigarrow$ Algorithm for short orbits

PAG 0.1 – What is implemented so far?

The standard Kramer-Mesner method for t-designs:

- Generating G-orbits of k-subsets of V [GAP code]
  - $\rightsquigarrow$ Orderly algorithm using GAP package **images**
  - $\rightsquigarrow$ Algorithm for short orbits
- Computing the Kramer-Mesner matrix [GAP code]

PAG 0.1 – What is implemented so far?

The standard Kramer-Mesner method for t-designs:

- Generating G-orbits of k-subsets of V [GAP code]
  - $\rightsquigarrow$ Orderly algorithm using GAP package **images**
  - $\rightsquigarrow$ Algorithm for short orbits
- Computing the Kramer-Mesner matrix [GAP code]
- Solving 0-1 systems by A. Wassermann's LLL solver [interface to C program]

PAG 0.1 – What is implemented so far?

The standard Kramer-Mesner method for t-designs:

- Generating G-orbits of k-subsets of V [GAP code]
  - $\rightsquigarrow$ Orderly algorithm using GAP package **images**
  - $\rightsquigarrow$ Algorithm for short orbits
- Computing the Kramer-Mesner matrix [GAP code]
- Solving 0-1 systems by A. Wassermann's LLL solver [interface to C program]
- Transforming solutions to GAP package **DESIGN** format [GAP code]

PAG 0.1 – What is implemented so far?

The standard Kramer-Mesner method for t-designs:

- Generating G-orbits of k-subsets of V [GAP code]
  - $\rightsquigarrow$ Orderly algorithm using GAP package **images**
  - $\rightsquigarrow$ Algorithm for short orbits
- Computing the Kramer-Mesner matrix [GAP code]
- Solving 0-1 systems by A. Wassermann's LLL solver [interface to C program]
- Transforming solutions to GAP package **DESIGN** format [GAP code]
- Command *KramerMesnerSearch* that does everything automatically

To do list: from PAG 0.1 to PAG 1.0

Enhancements of the Kramer-Mesner method:

- Tactical decomposition matrices

To do list: from PAG 0.1 to PAG 1.0

Enhancements of the Kramer-Mesner method:

- Tactical decomposition matrices
- Quasi-symmetric designs: good orbits, compatibility matrices

To do list: from PAG 0.1 to PAG 1.0

Enhancements of the Kramer-Mesner method:

- Tactical decomposition matrices
- Quasi-symmetric designs: good orbits, compatibility matrices
- More solvers: Gurobi, Minion. . .

To do list: from PAG 0.1 to PAG 1.0

Enhancements of the Kramer-Mesner method:

- Tactical decomposition matrices
- Quasi-symmetric designs: good orbits, compatibility matrices
- More solvers: Gurobi, Minion...

Other construction methods and types of objects:

- Quasi-symmetric designs by clique search

To do list: from PAG 0.1 to PAG 1.0

Enhancements of the Kramer-Mesner method:

- Tactical decomposition matrices
- Quasi-symmetric designs: good orbits, compatibility matrices
- More solvers: Gurobi, Minion...

Other construction methods and types of objects:

- Quasi-symmetric designs by clique search
- Configurations

To do list: from PAG 0.1 to PAG 1.0

Enhancements of the Kramer-Mesner method:

- Tactical decomposition matrices
- Quasi-symmetric designs: good orbits, compatibility matrices
- More solvers: Gurobi, Minion...

Other construction methods and types of objects:

- Quasi-symmetric designs by clique search
- Configurations
- Strongly regular graphs?

Worked examples in PAG

V. Krčadinac, *Some new designs with prescribed automorphism groups*, J. Combin. Des. **26** (2018), 193–200.

V. Krčadinac, *Some new designs with prescribed automorphism groups*, J. Combin. Des. **26** (2018), 193–200.

8 | DESIGNS WITH PARAMETERS 5-(16, 7, λ)

For $t = 5$, $v = 16$, and $k = 7$, we have $\lambda_{\min} = 5$, $\lambda_{\max} = \binom{11}{2} = 55$, and $M = 5$. By [7, Table 4.46], 5-$(16, 7, 5m)$ designs exist for $m \in \{3, 4, 5\}$. Here we settle the case $m = 2$.

Theorem 8.1. *Simple* 5-$(16, 7, 10)$ *designs exist.*

Proof. Let $G \cong (\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2).A_4$ be the group of order 192 generated by the permutations

$$(2, 3, 4)(5, 6, 7, 8, 9, 10)(11, 12, 13, 14, 15, 16)(1, 5)(2, 12)(3, 15)(4, 8)(6, 14)(7, 16)(9, 10)(11, 13).$$

The Kramer–Mesner system is of size 28×71 and has two solutions for $\lambda = 10$. The two designs are isomorphic and have $\mathrm{Aut}(\mathcal{D}) = G$. Base blocks are listed in Table 5. □

The same group G gives designs for $m = 5$, and for $m \in \{3, 4\}$ a subgroup of index 2 can be used. We did not find any designs for $m = 1$.

Let us present a small example from the paper [Krc18]. In Theorem 8.1, simple 5-(16,7,10) designs with the following automorphism group were constructed.

Example

```
gap> g:=Group((2,3,4)(5,6,7,8,9,10)(11,12,13,14,15,16),
> (1,5)(2,12)(3,15)(4,8)(6,14)(7,16)(9,10)(11,13));
```

They can be obtained by typing

Example

```
gap> KramerMesnerSearch(5,16,7,10,g);
Computing t-subset orbit representatives...
28
Computing k-subset orbit representatives...
71
Computing the Kramer-Mesner matrix...
[ 29, 72 ]
Starting solver...
No BOUNDS
The RHS is fixed !
No upper bounds: 0/1 variables are assumed

Orthogonal defect: 26.953339
First reduction successful
Orthogonal defect: 20.216092
Second reduction successful
.
```

Comments during the calculation can be supressed by setting global options.

Example

```
gap> PAGGlobalOptions.Silent:=true;
true
gap> KramerMesnerSearch(5,16,7,10,g);
[ [ [ 1, 2, 3, 4, 5, 6, 13 ], [ 1, 2, 3, 4, 5, 6, 14 ],
        [ 1, 2, 3, 5, 6, 7, 11 ], [ 1, 2, 3, 5, 6, 8, 9 ],
        [ 1, 2, 3, 5, 6, 9, 10 ], [ 1, 2, 3, 5, 6, 9, 12 ],
        [ 1, 2, 3, 5, 6, 10, 15 ], [ 1, 2, 3, 5, 6, 14, 16 ],
        [ 1, 2, 3, 5, 8, 11, 12 ], [ 1, 2, 5, 6, 7, 8, 16 ],
        [ 1, 2, 5, 6, 7, 9, 14 ], [ 1, 2, 5, 6, 7, 12, 13 ],
        [ 1, 2, 5, 6, 7, 14, 15 ] ],
    [ [ 1, 2, 3, 4, 5, 6, 8 ], [ 1, 2, 3, 4, 5, 6, 14 ],
        [ 1, 2, 3, 5, 6, 7, 11 ], [ 1, 2, 3, 5, 6, 9, 12 ],
        [ 1, 2, 3, 5, 6, 10, 12 ], [ 1, 2, 3, 5, 6, 10, 16 ],
        [ 1, 2, 3, 5, 6, 12, 13 ], [ 1, 2, 3, 5, 6, 14, 15 ],
        [ 1, 2, 3, 5, 8, 11, 12 ], [ 1, 2, 5, 6, 7, 8, 9 ],
        [ 1, 2, 5, 6, 7, 9, 14 ], [ 1, 2, 5, 6, 7, 12, 13 ],
        [ 1, 2, 5, 6, 11, 14, 16 ] ] ]
```

The output is a list of base blocks for two designs. There are options to get them in the Design package format (**DESIGN: Design**). Then we can also check that they are really 5-designs.

Worked examples in PAG

The output is a list of base blocks for two designs. There are options to get them in the Design package format (**DESIGN: Design**). Then we can also check that they are really 5-designs.

Example

```
gap> d:=KramerMesnerSearch(5,16,7,10,g,rec(Design:=true));;
gap> List(d,AllTDesignLambdas);
[ [ 2080, 910, 364, 130, 40, 10 ], [ 2080, 910, 364, 130, 40, 10 ] ]
```

The two designs are in fact isomorphic.

Example

```
gap> d:=KramerMesnerSearch(5,16,7,10,g,rec(NonIsomorphic:=true));;
gap> Size(d);
1
```

The option `NonIsomorphic` applies the function `BlockDesignIsomorphismClassRepresentatives` (**DESIGN: BlockDesignIsomorphismClassRepresentatives**) to the constructed designs.

Worked examples in PAG

The output is a list of base blocks for two designs. There are options to get them in the Design package format (**DESIGN: Design**). Then we can also check that they are really 5-designs.

Example
```
gap> d:=KramerMesnerSearch(5,16,7,10,g,rec(Design:=true));;
gap> List(d,AllTDesignLambdas);
[ [ 2080, 910, 364, 130, 40, 10 ], [ 2080, 910, 364, 130, 40, 10 ] ]
```

The two designs are in fact isomorphic.

Example
```
gap> d:=KramerMesnerSearch(5,16,7,10,g,rec(NonIsomorphic:=true));;
gap> Size(d);
1
```

The option `NonIsomorphic` applies the function `BlockDesignIsomorphismClassRepresentatives` (**DESIGN: BlockDesignIsomorphismClassRepresentatives**) to the constructed designs.

B. Schmalz, *The t-designs with prescribed automorphism group, new simple 6-designs*, J. Combin. Des. **1** (1993), 125–170.

The t-Designs with Prescribed Automorphism Group, New Simple 6-Designs

Bernd Schmalz
Universität Bayreuth, Postfach 10 12 51, W-8580 Bayreuth, Germany

ABSTRACT

We introduce an algorithm for the construction of a complete system of representatives of t-designs with given parameters $t-(v, k, \lambda)$ and prescribed full automorphism group A. It is based on the following observation published by Kramer and Mesner in 1976: The $t-(v, k, \lambda)$ designs admitting automorphism group A are exactly the 0-1-solutions $\vec{x}$ of the following system of linear equations

$$M_{t,k}^{A}\,\vec{x} = (\lambda, \ldots, \lambda)^{t}.$$

$M_{t,k}^{A}$ are incidence matrices, which we compute by means of double cosets. Representing

Summary 1.13. *The simple 6-designs known to the author:*

1. There are exactly 1 179 nonisomorphic $6-(33,8,36)$ designs having full automorphism group $\mathrm{P\Gamma L}_2(32)$.
2. There are exactly 2 nonisomorphic $6-(20,9,112)$ designs having full automorphism group $\mathrm{PSL}_2(19)$.
3. There are exactly 3 nonisomorphic $6-(28,8,42)$ designs having full automorphism group $\mathrm{P\Gamma L}_2(27)$.
4. There are exactly 367 nonisomorphic $6-(28,8,63)$ designs having full automorphism group $\mathrm{P\Gamma L}_2(27)$.
5. There are exactly 21 743 nonisomorphic $6-(28,8,84)$ designs having full automorphism group $\mathrm{P\Gamma L}_2(27)$.
6. There are exactly 38 277 nonisomorphic $6-(28,8,105)$ designs having full automorphism group $\mathrm{P\Gamma L}_2(27)$.
7. There are exactly 2 nonisomorphic $6-(14,7,4)$ designs with cyclic derived designs.
8. There are $6-(8m+6,7,4m)$ designs for all positive integers m.
9. There are $6-(22,8,60)$ designs.
10. There are $6-(23+16m,8,4(m+1)(16m+17))$ designs for all integers $m \geq 1$.

1.3.1 6-(14,7,4) Designs

The summary about known 6-designs on page 130 of [Sch93] mentions that there are exactly two 6-(14,7,4) designs with cyclic derived designs. This means that the two 6-designs have automorphisms of order 13. They can be constructed with the following GAP commands.

Example

```
gap> g:=Group(CyclicPermutation(13));
Group([ (1,2,3,4,5,6,7,8,9,10,11,12,13) ])
gap> d:=KramerMesnerSearch(6,14,7,4,g,rec(NonIsomorphic:=true));;
gap> List(d,AllTDesignLambdas);
[ [ 1716, 858, 396, 165, 60, 18, 4 ], [ 1716, 858, 396, 165, 60, 18, 4 ] ]
```

The solver quickly finds 24 solutions of the Kramer-Mesner system. Most of the computation time is used to eliminate isomorphic designs. Both designs have $\mathbb{Z}_{13}$ as their full automorphism group.

Example

```
gap> List(d,AutomorphismGroup);
[ Group([ (1,13,12,11,10,9,8,7,6,5,4,3,2) ]),
  Group([ (1,13,12,11,10,9,8,7,6,5,4,3,2) ]) ]
```

Worked examples in PAG

$M_{t,k}^A$ are incidence matrices, which we compute by means of double cosets. Representing the set of all solutions of the above system of equations implicitly by a graph gives us the possibility either to extract the solutions explicitly or to compute their precise numbers, which often are very big. We use the lattice of overgroups of A in the full symmetric group S_v for the construction or enumeration of the isomorphism types of the t-designs with full automorphism group A from these solutions. To the best of our knowledge our approach for the first time allows one to compute the precise number of isomorphism types or even these designs themselves for substantial numbers. We determined the (number of) isomorphism types for many known parameter sets and found new simple 6-designs with parameters

$$6 - (28, 8, \lambda), \lambda = 42, 63, 84, 105,$$

and full automorphism group $\mathrm{P\Gamma L}_2(27)$. We constructed all isomorphism types of these designs; their precise numbers are 3, 367, 21 743, 38 277, respectively.

1.3.2 6-(28,8,λ) Designs

In [Sch93], the existence of 6-(28,8,λ) designs was established for $\lambda = 42$, 63, 84, and 105. The exact numbers of these designs with automorphism group $P\Gamma L(2,27)$ were computed. While the projective general linear groups are readily available in GAP through the PGL command, there seemst to be no equivalent command for semilinear groups. Using the FinInG package, we can get $P\Gamma L(2,27)$ as the collineation group of the projective line over $GF(27)$.

Example
```
gap> LoadPackage("FinInG");
gap> g1:=CollineationGroup(ProjectiveSpace(1,27));
The FinInG collineation group PGammaL(2,27)
```

We need a permutation representation of this group on 28 points.

Example
```
gap> g:=Image(ActionOnAllProjPoints(g1));
Group([ (3,28,27,26,25,24,23,22,21,20,19,18,17,4,16,15,14,13,12,11,10,9,8,7,6,5),
  (1,2,4)(5,8,24)(6,21,10)(7,16,15)(9,25,28)(11,13,14)(12,27,23)(17,26,18)
  (19,20,22), (5,7,13)(6,10,21)(8,16,14)(9,18,22)(11,24,15)(12,27,23)(17,19,25)
  (20,28,26) ])
```

Alternatively, we can get the group from the library of small primitive permutation groups.

Example
```
gap> PrimitiveGroupsOfDegree(28);
[ PGL(2, 7), PSL(2, 8), PGammaL(2, 8), PSU(3, 3), PGammaU(3, 3), PSp(6, 2), A(8),
  S(8), PSL(2, 27), PGL(2, 27), PSL(2, 27):3, PGammaL(2, 27), A(28), S(28) ]
```

Worked examples in PAG

Now we can construct the designs with $\lambda = 42$.

Example

```
gap> d:=KramerMesnerSearch(6,28,8,42,g);;
Computing t-subset orbit representatives...
14
Computing k-subset orbit representatives...
72
Computing the Kramer-Mesner matrix...
.
.
.
Loops: 27732
Total number of solutions: 3

total enumeration time: 0:00:00
gap> Size(d);
4
```

Notice that A. Wassermann's LLL solver [Was98] reports finding 3 solutions, but we get 4 sets of base blocks. That's because the solver may return the same solution more than once. Here is how to get rid of multiple solutions.

Example

```
gap> Size(AsSet(d));
3
```

Worked examples in PAG

Most of the CPU time in the example above was used to compute the Kramer-Mesner matrix. The left-hand side of the Kramer-Mesner system is the same matrix for all λ, so we can compute it once and reuse it to save time.

Example

```
gap> tsub:=SubsetOrbitRepresentatives(g,28,6);;
gap> ksub:=SubsetOrbitRepresentatives(g,28,8);;
gap> m:=KramerMesnerMat(g,tsub,ksub);;
```

Now we can quickly get the exact numbers of designs from the paper [Sch93].

Example

```
gap> PAGGlobalOptions.Silent:=true;
true
gap> Size(AsSet(SolveKramerMesner(ExpandMatRHS(m,42))));
3
gap> Size(AsSet(SolveKramerMesner(ExpandMatRHS(m,63))));
367
gap> Size(AsSet(SolveKramerMesner(ExpandMatRHS(m,84))));
21743
gap> Size(AsSet(SolveKramerMesner(ExpandMatRHS(m,105))));
38277
```

Worked examples in PAG

A. Nakić, *The first example of a simple* 2-$(81, 6, 2)$ *design*, Examples and Counterexamples **1** (2021), 100005.

A B S T R A C T

We give the very first example of a simple $2-(81, 6, 2)$ design. Its points are the elements of the elementary abelian group of order 81 and each block is the union of two parallel lines of the 4-dimensional geometry over the field of order 3. Hence it is also additive.

Worked examples in PAG

A. Nakić, *The first example of a simple* 2-$(81,6,2)$ *design*, Examples and Counterexamples **1** (2021), 100005.

ABSTRACT

We give the very first example of a simple $2-(81,6,2)$ design. Its points are the elements of the elementary abelian group of order 81 and each block is the union of two parallel lines of the 4-dimensional geometry over the field of order 3. Hence it is also additive.

Let $G=\mathbb{Z}_3^4$ be the elementary abelian group of order 81. Given two elements $x \in G \setminus \{0\}$ and $y \in G \setminus \{0,x,2x\}$, let $B(x,y)$ be the union of the two parallel lines $\{0,x,2x\}$ and $\{y,x+y,2x+y\}$ of AG$(4,3)$. The G-stabilizer of $B(x,y)$ (under the natural action of G on itself) is clearly given by $\{0,x,2x\}$, hence its G-orbit has size $|G|/3=27$. Also, from the divisibility conditions we infer that a $2-(81,6,2)$ design has $432=27\cdot 16$ blocks. Thus it makes sense to look for a design with these parameters whose collection of blocks is the union of the G-orbits of 16 suitable blocks of the form $B(x,y)$. Such a 16-tuple of blocks has been found with a computer and it is given below.

$\{(0,0,0,0),(0,0,0,1),(0,0,0,2),(0,1,0,0),(0,1,0,1),(0,1,0,2)\}$

$\{(0,0,0,0),(0,0,1,1),(0,0,2,2),(2,1,0,0),(2,1,1,1),(2,1,2,2)\}$

$\{(0,0,0,0),(0,1,1,1),(0,2,2,2),(0,0,1,0),(0,1,2,1),(0,2,0,2)\}$

$\{(0,0,0,0),(0,1,2,0),(0,2,1,0),(2,0,2,1),(2,1,1,1),(2,2,0,1)\}$

$\{(0,0,0,0),(1,0,0,0),(2,0,0,0),(0,2,2,1),(1,2,2,1),(2,2,2,1)\}$

$\{(0,0,0,0),(1,0,1,0),(2,0,2,0),(0,1,0,0),(1,1,1,0),(2,1,2,0)\}$

$\{(0,0,0,0),(1,0,1,1),(2,0,2,2),(0,0,2,0),(1,0,0,1),(2,0,1,2)\}$

$\{(0,0,0,0),(1,0,2,0),(2,0,1,0),(0,2,1,1),(1,2,0,1),(2,2,2,1)\}$

$\{(0,0,0,0),(1,0,2,2),(2,0,1,1),(0,1,2,1),(1,1,1,0),(2,1,0,2)\}$

$\{(0,0,0,0),(1,1,0,0),(2,2,0,0),(0,2,0,1),(1,0,0,1),(2,1,0,1)\}$

$\{(0,0,0,0),(1,1,0,1),(2,2,0,2),(0,2,2,0),(1,0,2,1),(2,1,2,2)\}$

$\{(0,0,0,0),(1,1,2,0),(2,2,1,0),(0,0,2,1),(1,1,1,1),(2,2,0,1)\}$

$\{(0,0,0,0),(1,1,2,1),(2,2,1,2),(0,2,1,1),(1,0,0,2),(2,1,2,0)\}$

$\{(0,0,0,0),(1,1,2,2),(2,2,1,1),(0,2,2,0),(1,0,1,2),(2,1,0,1)\}$

$\{(0,0,0,0),(1,2,1,2),(2,1,2,1),(0,0,2,1),(1,2,0,0),(2,1,1,2)\}$

$\{(0,0,0,0),(1,2,2,0),(2,1,1,0),(0,2,2,1),(1,1,1,1),(2,0,0,1)\}$

1.3.3 2-(81,6,2) Designs

The first simple 2-(81,6,2) design was recently found by A. Nakic [Nak21]. Here are the base blocks of this design copy-pasted from the paper.

Example

```
gap> bb:=[[[0,0,0,0],[0,0,0,1],[0,0,0,2],[0,1,0,0],[0,1,0,1],[0,1,0,2]],
> [[0,0,0,0],[0,0,1,1],[0,0,2,2],[2,1,0,0],[2,1,1,1],[2,1,2,2]],
> [[0,0,0,0],[0,1,1,1],[0,2,2,2],[0,0,1,0],[0,1,2,1],[0,2,0,2]],
> [[0,0,0,0],[0,1,2,0],[0,2,1,0],[2,0,2,1],[2,1,1,1],[2,2,0,1]],
> [[0,0,0,0],[1,0,0,0],[2,0,0,0],[0,2,2,1],[1,2,2,1],[2,2,2,1]],
> [[0,0,0,0],[1,0,1,0],[2,0,2,0],[0,1,0,0],[1,1,1,0],[2,1,2,0]],
> [[0,0,0,0],[1,0,1,1],[2,0,2,2],[0,0,2,0],[1,0,0,1],[2,0,1,2]],
> [[0,0,0,0],[1,0,2,0],[2,0,1,0],[0,2,1,1],[1,2,0,1],[2,2,2,1]],
> [[0,0,0,0],[1,0,2,2],[2,0,1,1],[0,1,2,1],[1,1,1,0],[2,1,0,2]],
> [[0,0,0,0],[1,1,0,0],[2,2,0,0],[0,2,0,1],[1,0,0,1],[2,1,0,1]],
> [[0,0,0,0],[1,1,0,1],[2,2,0,2],[0,2,2,0],[1,0,2,1],[2,1,2,2]],
> [[0,0,0,0],[1,1,2,0],[2,2,1,0],[0,0,2,1],[1,1,1,1],[2,2,0,1]],
> [[0,0,0,0],[1,1,2,1],[2,2,1,2],[0,2,1,1],[1,0,0,2],[2,1,2,0]],
> [[0,0,0,0],[1,1,2,2],[2,2,1,1],[0,2,2,0],[1,0,1,2],[2,1,0,1]],
> [[0,0,0,0],[1,2,1,2],[2,1,2,1],[0,0,2,1],[1,2,0,0],[2,1,1,2]],
> [[0,0,0,0],[1,2,2,0],[2,1,1,0],[0,2,2,1],[1,1,1,1],[2,0,0,1]]]*Z(3)^0;;
```

Worked examples in PAG

The points of this design are elements of the 4-dimensional vector space V over $GF(3)$. Here is how to get the desing in the Design package format.

Example

```
gap> V:=Tuples([0,1,2],4)*Z(3)^0;;
gap> d1:=Union(List(bb,y->List(V,x->AsSet(x+y))));;
gap> d:=BlockDesign(81,List(d1,y->List(y,x->Position(V,x))));;
gap> AllTDesignLambdas(d);
[ 432, 32, 2 ]
```

The full automorphism group of the design is of order 2592. It's a semidirect product of the additive group of V and a group of order 32.

Example

```
gap> aut:=AutomorphismGroup(d);
<permutation group with 4 generators>
gap> Size(aut);
2592
gap> StructureDescription(aut);
"(C3 x C3 x C3 x C3) : (C16 : C2)"
```

Worked examples in PAG

This group has three subgroups of order 648 up to conjugation. We can use the second subgroup to construct four more simple 2-(81,6,2) designs.

Example
```
gap> g:=Filtered(AllSubgroupsConjugation(aut),x->Size(x)=648);
[ <permutation group of size 648 with 7 generators>,
  <permutation group of size 648 with 7 generators>,
  <permutation group of size 648 with 7 generators> ]
gap> dd:=KramerMesnerSearch(2,81,6,2,g[2],rec(NonIsomorphic:=true));;
gap> List(dd,x->Size(AutomorphismGroup(x)));
[ 1944, 15552, 1296, 2592, 3888 ]
```

Two of the new designs have larger full automorphism groups than design from [Nak21]. Using their subgroups, more simple 2-(81,6,2) designs can be constructed.

This group has three subgroups of order 648 up to conjugation. We can use the second subgroup to construct four more simple 2-(81,6,2) designs.

Example
```
gap> g:=Filtered(AllSubgroupsConjugation(aut),x->Size(x)=648);
[ <permutation group of size 648 with 7 generators>,
  <permutation group of size 648 with 7 generators>,
  <permutation group of size 648 with 7 generators> ]
gap> dd:=KramerMesnerSearch(2,81,6,2,g[2],rec(NonIsomorphic:=true));;
gap> List(dd,x->Size(AutomorphismGroup(x)));
[ 1944, 15552, 1296, 2592, 3888 ]
```

Two of the new designs have larger full automorphism groups than design from [Nak21]. Using their subgroups, more simple 2-(81,6,2) designs can be constructed.

Homework: construct more examples of simple $2\text{-}(81,6,2)$ designs!

The End

Thanks for your attention!