Algorithmic determination of isoclinism for 6-generator groups of nilpotency class 2

Ivo Michailov, Ivan Ivanov, Aleksandrina Aleksandrova, Gionul Ilianova

Abstract: *The purpose of this paper is to present a computer algorithm that classifies, with exactness to isoclinism, 6-generator groups of class 2. We prove two theoretical results that are used to replace the slow function 'AreIsoclinic' embedded in the package HAP for GAP [1].*
Key words: *GAP, HAP, nilpotency class 2.*

1. INTRODUCTION

In the paper [3] we described an algorithm for GAP, using the package HAP [1] that distributes all 4-generator groups of nilpotency class 2 into isoclinism classes. It turns out, however, that the function 'AreIsoclinic' embedded in the package HAP is extremely slow and requires a lot of RAM. For example, the verification for a pair of group of order 3^{10}, GAP (under Linux) needed 100 GB RAM, and after a month we still did not get the result. Moreover, the function does not allow the process to be distributed to other computers. Taking into account that even for small orders there are hundreds of groups that have to be analyzed pairwise, it becomes clear that this function will not give the desired results. In section 2 of this paper we find a criteria for isoclinism that can be used for the construction of another algorithm. In section 3 we give the new function based on the results from section 2, and specifically designed for 6-generator groups of nilpotency class 2. The new function is significantly faster, especially if it is run on multiple CPU cores. Finally, we display the algorithm for isoclinic classification of these groups.

2. 6-GENERATOR GROUPS OF NILPOTENCY CLASS 2

We begin by proving a criteria when arbitrary chosen two elements from a 2-generator abelian group generate direct cyclic factors.

Theorem 2.1. Let p be any prime, and let C be any abelian p-group with two generators γ_1, γ_2 , i.e., $C = \langle\gamma_1\rangle \times \langle\gamma_2\rangle \simeq C_{p^{c_1}} \times C_{p^{c_2}}$ for some natural numbers c_1, c_2 . Then for any $\delta_1, \delta_2 \in C$ we have $C = \langle\delta_1\rangle \times \langle\delta_2\rangle$ if and only if $C = \langle\delta_1, \delta_2\rangle$ and $\{|\delta_1|, |\delta_2|\} = \{p^{c_1}, p^{c_2}\}$.

Proof. We need to prove the 'only if' part, because the 'if' part follows from the uniqueness of the decomposition of an abelian group into a direct product of cyclic factors. Now, let $C = \langle\delta_1, \delta_2\rangle$ and $|\delta_1| = p^{c_1}, |\delta_2| = p^{c_2}$.

<u>Case I.</u> Let $c_1 = c_2 = c$. Since $(\delta_1\delta_2^p)^{p^c} = (\delta_1^p\delta_2)^{p^c} = 1$ and $(\delta_1\delta_2^p)^{p^k} \neq 1$, $(\delta_1^p\delta_2)^{p^k} \neq 1$ for $0 < k < c$, we have that $\langle\delta_1\rangle \cap \langle\delta_2\rangle = \{1\}$, so $C = \langle\delta_1\rangle \times \langle\delta_2\rangle$.

<u>Case II.</u> Let $c_1 < c_2$. We can write $\delta_1 = \gamma_1^{a_1}\gamma_2^{a_2}, \delta_2 = \gamma_1^{b_1}\gamma_2^{b_2}$ for some integers a_1, a_2, b_1, b_2. From $|\delta_1| = p^{c_1}$ it follows that $a_2 \equiv 0 \left(\bmod p^{c_2 - c_1}\right)$. We also have $\gcd(b_2, p) = 1$, because $|\delta_2| = p^{c_2}$.

Next, suppose that $a_1 \equiv 0 \left(\bmod p\right)$. It is clear now that the power $\gamma_2^{b_2}$ can not be eliminated by a multiplication of δ_2 with a power of δ_1, i.e., γ_1 is not in $\langle\delta_1, \delta_2\rangle$, a contradiction.

Therefore, we must assume that $\gcd(a_1,p)=1$. Suppose that $\delta_1^{p^c}\in\langle\delta_2\rangle$, i.e., $\delta_1^{p^c}=\gamma_1^{a_1p^c}\gamma_2^{a_2p^c}=\gamma_1^{b_1d}\gamma_2^{b_2d}$ for some integers c and d such that $0\le c<c_1, 0\le d<c_2$. Therefore, $a_1p^c\equiv b_1d\left(\bmod p^{c_1}\right)$ and $a_2p^c\equiv b_2d\left(\bmod p^{c_2}\right)$. Since $\gcd(b_2,p)=1$ and $a_2\equiv 0\left(\bmod p^{c_2-c_1}\right)$, we obtain that $p^{c_2-c_1+c}$ divides d. Hence $p^{c_2-c_1}$ divides a_1, a contradiction.

Finally, suppose that $\delta_1^x\in\langle\delta_2\rangle$ for some integer x such that $\gcd(x,p)=1$. We can write $\delta_1^x=\delta_2^y$ for some integer y. Therefore $a_1x\equiv b_1y\left(\bmod p^{c_1}\right)$ and $a_2x\equiv b_2y\left(\bmod p^{c_2}\right)$, so we must have $\gcd(a_1,p)=\gcd(b_1,p)=\gcd(y,p)=\gcd(b_2,p)=\gcd(a_2,p)=1$. But then $|\delta_1|=p^{c_2}$, a contradiction.

In this way, we have shown that $\langle\delta_1\rangle\cap\langle\delta_2\rangle=\{1\}$, so $C=\langle\delta_1\rangle\times\langle\delta_2\rangle$. □

Now, let G be a p-group of nilpotency class ≤ 2, let B be an abelian normal subgroup with two generators, and let the quotient group G/B be an abelian group with two generators. In a future paper we will show that any such group G is isoclinic to a group generated by elements $\alpha_1,\alpha_2,\beta_1,\beta_2,\gamma_1,\gamma_2$ such that $\alpha_j^{p^{a_j}}=\beta_i^{p^{b_i}}=\gamma_k^{p^{c_k}}=1,[\alpha_1,\alpha_2]=1,$ $[\beta_1,\beta_2]=1,[\beta_i,\alpha_j]=\gamma_1^{u_{ij}}\gamma_2^{v_{ij}}$, where $0\le u_{ij}\le p^{c_1}-1, 0\le v_{ij}\le p^{c_2}-1$ and i,j,k run through the set $\{1,2\}$. Also, we have $\langle\alpha_1,\alpha_2\rangle^p\cap Z(G)=\{1\},\langle\beta_1,\beta_2\rangle^p\cap Z(G)=\{1\}$ and $G'=\langle\gamma_1\rangle\times\langle\gamma_2\rangle\le Z(G)$.

We say that G is of the type $(a_1,a_2,b_1,b_2,c_1,c_2,u_{11},u_{12},u_{21},u_{22},v_{11},v_{12},v_{21},v_{22})$. The conditions for a_i,b_i,c_i so that G is well defined are given in [3]. Of course, the type of G depends on the choice of the generators $\alpha_i,\beta_i,\gamma_i$.

Theorem 2.2. The group G of the type $(a_1,a_2,b_1,b_2,c_1,c_2,u_{11},u_{12},u_{21},u_{22},v_{11},v_{12},v_{21},v_{22})$ is isoclinic to the group H of the type $(a_1',a_2',b_1',b_2',c_1',c_2',u_{11}',u_{12}',u_{21}',u_{22}',v_{11}',v_{12}',v_{21}',v_{22}')$ if and only if there exist $\alpha_i'',\beta_i''\in\langle\alpha_1,\alpha_2,\beta_1,\beta_2\rangle$ and $\gamma_i''\in\langle\gamma_1,\gamma_2\rangle$ such that $G=\langle\alpha_i'',\beta_i'',\gamma_i''\rangle$ for $i=1,2$, and regarding the new generators, G is of the type $(a_1',a_2',b_1',b_2',c_1',c_2',u_{11}',u_{12}',u_{21}',u_{22}',v_{11}', v_{12}',v_{21}',v_{22}')$.

Proof. 'If' part. Clearly H is generated by elements $\alpha_1',\alpha_2',\beta_1',\beta_2',\gamma_1',\gamma_2'$ such that $\alpha_j'^{p^{a_j'}}=\beta_i'^{p^{b_i'}}=\gamma_k'^{p^{c_k'}}=1,[\alpha_1',\alpha_2']=1,[\beta_1',\beta_2']=1,[\beta_i',\alpha_j']=\gamma_1'^{u_{ij}'}\gamma_2'^{v_{ij}'}$. We define the maps $\theta:G/Z(G)\to H/Z(H)$ and $\phi:G'\to H'$ by $\theta(\alpha_i''Z(G))=\alpha_i'Z(H)$, $\theta(\beta_i''Z(G))=\beta_i'Z(H)$ and $\phi(\gamma_i'')=\gamma_i'$. Note that ϕ is an isomorphism, because $|\gamma_i''|=|\gamma_i'|$, $G'=\langle\gamma_1'',\gamma_2''\rangle$ and we can apply theorem 2.1. Since $\langle\alpha_1,\alpha_2\rangle^p\cap Z(G)=\{1\}$, $\langle\beta_1,\beta_2\rangle^p\cap Z(G)=\{1\}$, we obtain that $|\alpha_i''Z(G)|=|\alpha_i'Z(H)|$ and $|\beta_i''Z(G)|=|\beta_i'Z(H)|$. Then according to theorem 2.1, θ is an isomorphism, because $G/Z(G)=\langle\alpha_i''Z(G),\beta_i''Z(G)\rangle$ and $H/Z(H)=\langle\alpha_i'Z(H),\beta_i'Z(H)\rangle$.

Moreover, $\phi(\gamma_1''^{u'_{ij}}\gamma_2''^{v'_{ij}}) = \phi([\beta_i'', \alpha_j'']) = [\beta_i', \alpha_j']$.

Therefore, G and H are isoclinic.

<u>'Only if' part.</u> Let G and H be isoclinic and let $\theta : G/Z(G) \to H/Z(H)$ and $\phi : G' \to H'$ be isomorphisms such that $\phi([x,y]) = [x',y']$, where $x'Z(H) = \theta(xZ(G))$ and $y'Z(H) = \theta(yZ(G))$. Then there exist $\alpha_i'', \beta_i'' \in \langle \alpha_1, \alpha_2, \beta_1, \beta_2 \rangle$ such that $\theta(\alpha_i''Z(G)) = \alpha_i'Z(H), \theta(\beta_i''Z(G)) = \beta_i'Z(H)$ and $\phi([\beta_i'', \alpha_j'']) = [\beta_i', \alpha_j']$. Since $\langle \alpha_1, \alpha_2 \rangle^p \cap Z(G) = \{1\}$ and $\langle \beta_1, \beta_2 \rangle^p \cap Z(G) = \{1\}$, we obtain that $\langle \alpha_1'', \alpha_2'' \rangle$ is isomorphic to $\langle \alpha_1', \alpha_2' \rangle$ and $\langle \beta_1'', \beta_2'' \rangle$ is isomorphic to $\langle \beta_1', \beta_2' \rangle$. Moreover, $\langle \alpha_1'', \alpha_2'', \beta_1'', \beta_2'' \rangle = \langle \alpha_1, \alpha_2, \beta_1, \beta_2 \rangle$. From theorem 2.1 now it follows that $\langle \alpha_1'', \alpha_2'' \rangle = \langle \alpha_1'' \rangle \times \langle \alpha_2'' \rangle$ and $\langle \beta_1'', \beta_2'' \rangle = \langle \beta_1'' \rangle \times \langle \beta_2'' \rangle$. Put $\gamma_i'' = \phi^{-1}(\gamma_i') \in \langle \gamma_1, \gamma_2 \rangle$ for $i = 1,2$. Again by theorem 2.1, we get that $\langle \gamma_1'', \gamma_2'' \rangle$ is isomorphic to the direct product $\langle \gamma_1'' \rangle \times \langle \gamma_2'' \rangle$. Since ϕ is an isomorphism, we get that $\gamma_1''^{u'_{ij}}\gamma_2''^{v'_{ij}} = [\beta_i'', \alpha_j'']$ for any i, j. Therefore, regarding the generators $\alpha_i'', \beta_i'', \gamma_i''$, the group G is of the type $(a_1', a_2', b_1', b_2', c_1', c_2', u_{11}', u_{12}', u_{21}', u_{22}', v_{11}', v_{12}', v_{21}', v_{22}')$. □

3. ALGORITHMIC DETERMINATION OF ISOCLINISM

As we mentioned In the introduction, the isoclinism function in the package HAP is very slow, so we suggest the following function "AreIsoclinic" that can replace the same function in the file "bogomolov.gi", which can be found in the directory "C:\gap4r7\pkg\Hap1.10\lib\NonabelianTensor". This function is based on theorem 2.2.

```
#####################################################################
InstallGlobalFunction(AreIsoclinic, function(arg)
local G,H,bool, GhomGZ, HhomHZ, GZ, gensGZ, HZ, DG, DH, p,  a, b, c, i, aa, bb, cc,
aaa, bbb, ah, bh, ch, gensG, gensH, G1, C, C1, i1, i2, i3, i4, i5, i6, i7, i8, j1, j2, j3, j4,
j5, j6, j7, j8, k1, k2, k3, k4, a1, a2, b1, b2, c1, c2, ah1, ah2, bh1, bh2, ch1, ch2,
u11, u12, u21, u22, v11, v12, v21, v22, B, B1, i9, i10, i11, i12, i13, i14, i15, i16;
#This function can return true, false or fail
G:=arg[1];H:=arg[2];
if Length(arg)>2 then bool:=arg[3]; else bool:=true; fi;

##########################Quick tests##########################
if G=H then return true; fi;
DG:=DerivedSubgroup(G); DH:=DerivedSubgroup(H);
if not AbelianInvariants(DG)=AbelianInvariants(DH) then return false; fi;
if Order(DG)=1 then return true; fi;
GhomGZ:=NaturalHomomorphismByNormalSubgroup(G,Center(G));
GZ:=Image(GhomGZ);
HhomHZ:=NaturalHomomorphismByNormalSubgroup(H,Center(H));
HZ:=Image(HhomHZ);
if not AbelianInvariants(GZ)=AbelianInvariants(HZ) then return false; fi;

##########################Quick tests finished################
gensG:=GeneratorsOfGroup(G); gensH:=GeneratorsOfGroup(H);
a:=[];b:=[];c:=[];aa:=[];bb:=[];cc:=[];aaa:=[];bbb:=[]; ah:=[];bh:=[];ch:=[];
for i in [1..2] do a[i]:=gensG[i]; ah[i]:=gensH[i]; od;
```

```
for i in [1..2] do b[i]:=gensG[i+2]; bh[i]:=gensH[i+2]; od;
for i in [1..2] do c[i]:=gensG[i+4]; ch[i]:=gensH[i+4]; od;
a1:=Order(a[1]);a2:=Order(a[2]);b1:=Order(b[1]);b2:=Order(b[2]);c1:=Order(c[1]);c2:=Order
(c[2]);
ah1:=Order(ah[1]);ah2:=Order(ah[2]);bh1:=Order(bh[1]);bh2:=Order(bh[2]);ch1:=Order(ch[
1]);ch2:=Order(ch[2]);
for i1 in [0..ch1-1] do for i2 in [0..ch2-1] do
if Comm(bh[1],ah[1])=ch[1]^i1*ch[2]^i2 then u11:=i1;v11:=i2;fi;
if Comm(bh[1],ah[2])=ch[1]^i1*ch[2]^i2 then u12:=i1;v12:=i2;fi;
if Comm(bh[2],ah[1])=ch[1]^i1*ch[2]^i2 then u21:=i1;v21:=i2;fi;
if Comm(bh[2],ah[2])=ch[1]^i1*ch[2]^i2 then u22:=i1;v22:=i2;fi;
od;od;
C:=Subgroup(G, [c[1],c[2]]);
for i1 in [0..a1-1] do   for i2 in [0..a2-1] do   for i3 in [0..a1-1] do   for i4 in [0..a2-1] do
for i5 in [0..b1-1] do   for i6 in [0..b2-1] do   for i7 in [0..b1-1] do   for i8 in [0..b2-1] do
for i9 in [0..b1-1] do   for i10 in [0..b2-1] do   for i11 in [0..b1-1] do   for i12 in [0..b2-1] do
for i13 in [0..a1-1] do   for i14 in [0..a2-1] do   for i15 in [0..a1-1] do   for i16 in [0..a2-1] do
aa[1]:=a[1]^i1*a[2]^i2*b[1]^i9*b[2]^i10; aa[2]:=a[1]^i3*a[2]^i4*b[1]^i11*b[2]^i12;
bb[1]:=b[1]^i5*b[2]^i6*a[1]^i13*a[2]^i14; bb[2]:=b[1]^i7*b[2]^i8*a[1]^i15*a[2]^i16;
G1:=Subgroup(G, [aa[1],aa[2],bb[1],bb[2],c[1],c[2]]);
B:=Subgroup(G, [b[1],b[2],c[1],c[2]]); B1:=Subgroup(G, [bb[1],bb[2],c[1],c[2]]);
if Size(G1)=Size(G) and Size(B1)=Size(B) and IsAbelian(B1)
and Order(aa[1])=ah1 and Order(aa[2])=ah2 and Order(bb[1])=bh1 and Order(bb[2])=bh2
then
for k1 in [0..c1-1] do   for k2 in [0..c2-1] do   for k3 in [0..c1-1] do   for k4 in [0..c2-1] do
cc[1]:=c[1]^k1*c[2]^k2; cc[2]:=c[1]^k3*c[2]^k4; C1:=Subgroup(C, [cc[1],cc[2]]);
if Size(C1)=Size(C) and Order(cc[1])=ch1 and Order(cc[2])=ch2 and
Comm(bb[1],aa[1])=cc[1]^u11*cc[2]^v11 and Comm(bb[1],aa[2])=cc[1]^u12*cc[2]^v12 and
Comm(bb[2],aa[1])=cc[1]^u21*cc[2]^v21 and Comm(bb[2],aa[2])=cc[1]^u22*cc[2]^v22 then
return true;fi;
od;od;od;od;fi;od;od;od;od;od;od;od;od;od;od;od;od;od;od;od;od;
return false;
end);
##################################################################
```

For example, if G is of the type $(1,1,1,1,1,1,u_{11},u_{12},u_{21},u_{22},v_{11},v_{12},v_{21},v_{22})$, we can construct the following algorithm which is similar to the one given in [3]:

```
################################################################
p:=3;; a1:=1;; a2:=1;;  b1:=1;; b2:=1;;  c1:=1;; c2:=1;;  L:=[];; M:=[];; m:=0;; ID:=[];;
    for u11 in [0..p^c1-1] do for u12 in [0..p^c1-1] do
      for u21 in [0..p^c1-1] do   for u22 in [0..p^c1-1] do
        for v11 in [0..p^c2-1] do    for v12 in [0..p^c2-1] do
          for v21 in [0..p^c2-1] do   for v22 in [0..p^c2-1] do
            m:=m+1;
            F := FreeGroup(IsSyllableWordsFamily,"x1","x2","y1","y2","z1","z2");;
            x1 := F.1;; x2 := F.2;; y1 := F.3;; y2 := F.4;; z1 := F.5;; z2 := F.6;;
            rels := [x1^(p^a1), x2^(p^a2), y1^(p^b1), y2^(p^b2), z1^(p^c1), z2^(p^c2),
            Comm(y1,x1)/(z1^u11*z2^v11), Comm(y1,x2)/(z1^u12*z2^v12),
            Comm(y2,x1)/(z1^u21*z2^v21), Comm(y2,x2)/(z1^u22*z2^v22)];
            H := F / rels;
            G := PcGroupFpGroup(H);
```

```
            if IsPcGroup(G) then G := PcGroupToPcpGroup(G); fi;
            Add(L,G); Add(M, [a1,a2,b1,b2,c1,c2,u11,u12,u21,u22,v11,v12,v21,v22]);
            GG:=Pcgs(G); Code:=CodePcgs(GG); ID[m] := Code;
    od;  od;  od;  od; od;  od; od; od;
C:=IsoclinismClasses(L);;
for c in C do pc:=Pcgs(c[1]); for i in [1..m] do if ID[i]=CodePcgs(pc) then Print(M[i]);fi;
od; od;
###############################################################
```

Although we have 3^8 groups that need to be classified, by splitting the cycles in the latter algorithm, the work can be distributed to multiple CPU cores (and computers). In the end, we obtained the following.

Theorem 3.1. Let G be a 3-group of nilpotency class ≤ 2, and let G be of the type $(1,1,1,1,1,1,u_{11},u_{12},u_{21},u_{22},v_{11},v_{12},v_{21},v_{22})$. Then G is isoclinic to precisely one of the following groups (according to James [2] notations):

$$\{1\},\Phi_2(111),\Phi_4(1^5),\Phi_5(1^5),\Phi_{12}(1^6),\Phi_{13}(1^6),\Phi_{15}(1^6).$$

4. CONCLUSION

The algorithms we developed in this paper and in [3] allow us to classify effectively the groups of nilpotency class 2 having up to 6 generators. Moreover, the function 'AreIsoclinic' can be modified easily for a group with more than 6 generators. In most cases that we encountered, the number of isoclinism classes is the same for any prime p. Taking into account the well-known fact that two isoclinic groups have equal Bogomolov multipliers, we will be able to prove more general results concerning the Bogomolov multipliers.

ACKNOWLEDGEMENTS. This paper is partially supported by a project RD-08-287/12.03.2015 of Shumen University.

REFERENCES

[1]. Ellis, G. ''GAP package HAP – homological algebra programming, Version 1.10.15'', available online (http://www.gap-system.org/Packages/hap.html), 2014.

[2]. James, R. The groups of order p^6 (p an odd prime), Math. Comp. 34 No. 150 (1980), 613-637.

[3]. Michailov, I, Ivan Ivanov, Nikola Ziapkov, Algorithmic generation of isoclinism classes for 4-generator groups of nilpotency class 2, Proceedings of University of Ruse, 2015 (to appear).

ABOUT THE AUTHORS:

Prof. D. Sc. Ivo Michailov Michailov, e-mail: ivo@fmi.shu-bg.net