







--  
--  
--

PcpGroupByMatGroup( $G$ )

$GGG$ fail  
IsomorphismPcpGroup( $G$ )

$GGG$ fail  
IsomorphismPcpGroupAlmostCrystallographicGroupPOL\_AlmostCrystallographicGroup  
AlmostCrystallographicGroup  
ImagesRepresentative( $map$ ,  $elm$ )  
ImageElm( $map$ ,  $elm$ )  
ImagesSet( $map$ ,  $elms$ )

$mapGH$ IsomorphismPcpGroupelmGImageElmelmmapeelmGImageElmfailpcpelmH  
IsSolvableGroup( $G$ )

$GG$ truefalse  
IsTriangularizableMatGroup( $G$ )

$GG$ truefalse  
IsPolycyclicGroup( $G$ )

$GG$ truefalse  
-

RadicalSeriesSolvableMatGroup( $G$ )

- $G$

HomogeneousSeriesAbelianMatGroup( $G$ )

- $G$

HomogeneousSeriesTriangularizableMatGroup( $G$ )

- $G$

CompositionSeriesAbelianMatGroup( $G$ )

- $G$

CompositionSeriesTriangularizableMatGroup( $G$ )

- $G$

SubgroupsUnipotentByAbelianByFinite( $G$ )

$GG$ -- $GG$ fail

PolExamples( $l$ )

$l$

		Example		
PolExamples	number generators	subgroup of	Hirsch length	
1	3	GL(4,Z)	6	
2	2	GL(5,Z)	6	
3	2	GL(4,Q)	4	
4	2	GL(5,Q)	6	
5	9	GL(16,Z)	3	
6	6	GL(4,Z)	3	
7	6	GL(4,Z)	3	
8	7	GL(4,Z)	3	
9	5	GL(4,Q)	3	
10	4	GL(4,Q)	3	
11	5	GL(4,Q)	3	
12	5	GL(4,Q)	3	
13	5	GL(5,Q)	4	
14	6	GL(5,Q)	4	
15	6	GL(5,Q)	4	
16	5	GL(5,Q)	4	
17	5	GL(5,Q)	4	
18	5	GL(5,Q)	4	
19	5	GL(5,Q)	4	
20	7	GL(16,Z)	3	
21	5	GL(16,Q)	3	
22	4	GL(16,Q)	3	
23	5	GL(16,Q)	3	
24	5	GL(16,Q)	3	

### Example

```

gap> mats :=
[ [ [ 1, 0, -1/2, 0 ], [ 0, 1, 0, 1 ], [ 0, 0, 1, 0 ], [ 0, 0, 0, 1 ] ],
  [ [ 1, 1/2, 0, 0 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 1 ], [ 0, 0, 0, 1 ] ],
  [ [ 1, 0, 0, 1 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 0 ], [ 0, 0, 0, 1 ] ],
  [ [ 1, -1/2, -3, 7/6 ], [ 0, 1, -1, 0 ], [ 0, 1, 0, 0 ], [ 0, 0, 0, 1 ] ],
  [ [ -1, 3, 3, 0 ], [ 0, 0, 1, 0 ], [ 0, 1, 0, 0 ], [ 0, 0, 0, 1 ] ] ];

gap> G := Group( mats );
<matrix group with 5 generators>

# calculate an isomorphism from G to a pcg-group
gap> nat := IsomorphismPcgGroup( G );

gap> H := Image( nat );
Pcg-group with orders [ 2, 2, 3, 5, 5, 5, 0, 0, 0 ]

gap> h := GeneratorsOfGroup( H );
[ g1, g2, g3, g4, g5, g6, g7, g8, g9]

gap> mats2 := List( h, x -> PreImage( nat, x ) );

# take a random element of G
gap> exp := [ 1, 1, 1, 1, 0, 0, 0, 0, 1 ];
gap> g := MappedVector( exp, mats2 );
[ [ -1, 17/2, -1, 233/6 ],
  [ 0, 1, 0, -2 ],
  [ 0, 1, -1, 2 ],
  [ 0, 0, 0, 1 ] ]

# map g into the image of nat
gap> i := ImageElm( nat, g );
g1*g2*g3*g4*g9

# exponent vector
gap> Exponents( i );

```

```

[ 1, 1, 1, 1, 0, 0, 0, 0, 1 ]

# compare the preimage with g
gap> PreImagesRepresentative( nat, i );
[ [ -1, 17/2, -1, 233/6 ],
  [ 0, 1, 0, -2 ],
  [ 0, 1, -1, 2 ],
  [ 0, 0, 0, 1 ] ]

gap> last = g;
true

```

#### Example

```

gap> gens :=
[ [ [ 1746/1405, 524/7025, 418/1405, -77/2810 ],
    [ 815/843, 899/843, -1675/843, 415/281 ],
    [ -3358/4215, -3512/21075, 4631/4215, -629/1405 ],
    [ 258/1405, 792/7025, 1404/1405, 832/1405 ] ],
  [ [ -2389/2810, 3664/21075, 8942/4215, -35851/16860 ],
    [ 395/281, 2498/2529, -5105/5058, 3260/2529 ],
    [ 3539/2810, -13832/63225, -12001/12645, 87053/50580 ],
    [ 5359/1405, -3128/21075, -13984/4215, 40561/8430 ] ] ];

gap> H := Group( gens );
<matrix group with 2 generators>

gap> RadicalSeriesSolvableMatGroup( H );
[ [ [ 1, 0, 0, 0 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 0 ], [ 0, 0, 0, 1 ] ],
  [ [ 1, 0, 0, 79/138 ], [ 0, 1, 0, -275/828 ], [ 0, 0, 1, -197/414 ] ],
  [ [ 1, 0, -3, 2 ], [ 0, 1, 55/4, -55/8 ] ],
  [ [ 1, 4/15, 2/3, 1/6 ] ],
  [ ] ]

```

#### Example

```

gap> G := PolExamples(3);
<matrix group with 2 generators>

gap> GeneratorsOfGroup( G );
[ [ [ 73/10, -35/2, 42/5, 63/2 ],
    [ 27/20, -11/4, 9/5, 27/4 ],
    [ -3/5, 1, -4/5, -9 ],
    [ -11/20, 7/4, -2/5, 1/4 ] ],
  [ [ -42/5, 423/10, 27/5, 479/10 ],
    [ -23/10, 227/20, 13/10, 231/20 ],
    [ 14/5, -63/5, -4/5, -79/5 ],
    [ -1/10, 9/20, 1/10, 37/20 ] ] ]

gap> subgroups := SubgroupsUnipotentByAbelianByFinite( G );
rec( T := <matrix group with 2 generators>,

```



```

      U := <matrix group with 4 generators> )

gap> GeneratorsOfGroup( subgroups.T );
[ [ [ 73/10, -35/2, 42/5, 63/2 ],
    [ 27/20, -11/4, 9/5, 27/4 ],
    [ -3/5, 1, -4/5, -9 ],
    [ -11/20, 7/4, -2/5, 1/4 ] ],
  [ [ -42/5, 423/10, 27/5, 479/10 ],
    [ -23/10, 227/20, 13/10, 231/20 ],
    [ 14/5, -63/5, -4/5, -79/5 ],
    [ -1/10, 9/20, 1/10, 37/20 ] ] ]

# so G is triangularizable!

```

pkgpolenta-

-<https://www.gap-system.org/Packages/packages.html>

LoadPackage

Example

```
gap> ReadPackage( "Polenta", "tst/testall.g" );
```

mhorn@rptu.de

InfoPolenta

SetInfoLevel(InfoPolenta,level)InfoPolenta

InfoLevel( InfoPolenta )

InfoLevel( InfoPolenta )[http://www.icm.tu-bs.de/ag\\_algebra/software/assmann/diploma.pdf](http://www.icm.tu-bs.de/ag_algebra/software/assmann/diploma.pdf)

InfoLevel( InfoPolenta )

Example

```
gap> SetInfoLevel( InfoPolenta, 1 );

gap> PcpGroupByMatGroup( PolExamples(11) );
#I Determine a constructive polycyclic sequence
for the input group ...
#I
#I Chosen admissible prime: 3
#I
#I Determine a constructive polycyclic sequence
for the image under the p-congruence homomorphism ...
#I finished.
#I Finite image has relative orders [ 3, 2, 3, 3, 3 ].
#I
#I Compute normal subgroup generators for the kernel
of the p-congruence homomorphism ...
#I finished.
#I
#I Compute the radical series ...
#I finished.
#I The radical series has length 4.
#I
#I Compute the composition series ...
#I finished.
#I The composition series has length 5.
```

```

#I
#I Compute a constructive polycyclic sequence
#I for the induced action of the kernel to the composition series ...
#I finished.
#I This polycyclic sequence has relative orders [  ].
#I
#I Calculate normal subgroup generators for the
#I unipotent part ...
#I finished.
#I
#I Determine a constructive polycyclic sequence
#I for the unipotent part ...
#I finished.
#I The unipotent part has relative orders
#I [ 0, 0, 0 ].
#I
#I ... computation of a constructive
#I polycyclic sequence for the whole group finished.
#I
#I Compute the relations of the polycyclic
#I presentation of the group ...
#I Compute power relations ...
#I ... finished.
#I Compute conjugation relations ...
#I ... finished.
#I Update polycyclic collector ...
#I ... finished.
#I finished.
#I
#I Construct the polycyclic presented group ...
#I finished.
#I
#I
Pcp-group with orders [ 3, 2, 3, 3, 3, 0, 0, 0 ]

```

```

gap> SetInfoLevel( InfoPolenta, 2 );

```

```

gap> PcpGroupByMatGroup( PolExamples(11) );
#I Determine a constructive polycyclic sequence
#I for the input group ...
#I
#I Chosen admissible prime: 3
#I
#I Determine a constructive polycyclic sequence
#I for the image under the p-congruence homomorphism ...
#I finished.
#I Finite image has relative orders [ 3, 2, 3, 3, 3 ].
#I
#I Compute normal subgroup generators for the kernel
#I of the p-congruence homomorphism ...
#I finished.
#I The normal subgroup generators are

```

```

#I [ [ [ 1, -3/2, 0, 0 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 3 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, 0, 0, 24 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 0 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, 3, 3, 15 ], [ 0, 1, 0, 6 ], [ 0, 0, 1, -6 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, 3, 3, 9 ], [ 0, 1, 0, 6 ], [ 0, 0, 1, -6 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, 3/2, 3/2, 3/2 ], [ 0, 1, 0, 3 ], [ 0, 0, 1, -3 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, -3/2, 9/2, -69/2 ], [ 0, 1, 0, 9 ], [ 0, 0, 1, 3 ], [ 0, 0, 0, 1 ] ],
    , [ [ 1, 0, 0, -24 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 0 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, -3, -3, -9 ], [ 0, 1, 0, -6 ], [ 0, 0, 1, 6 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, -3, -3, -15 ], [ 0, 1, 0, -6 ], [ 0, 0, 1, 6 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, -3, 0, 9 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 6 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, -3, -3, -9 ], [ 0, 1, 0, -6 ], [ 0, 0, 1, 6 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, -3, 0, 9 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 6 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, -3/2, -3/2, -9/2 ], [ 0, 1, 0, -3 ], [ 0, 0, 1, 3 ], [ 0, 0, 0, 1 ] ],
    ],
    [ [ 1, -3, -3, -12 ], [ 0, 1, 0, -6 ], [ 0, 0, 1, 6 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, 3, -3/2, -21 ], [ 0, 1, 0, -3 ], [ 0, 0, 1, -6 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, 3/2, 3/2, 9/2 ], [ 0, 1, 0, 3 ], [ 0, 0, 1, -3 ], [ 0, 0, 0, 1 ] ] ]

#I
#I Compute the radical series ...
#I finished.
#I The radical series has length 4.
#I The radical series is
#I [ [ [ 1, 0, 0, 0 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 0 ], [ 0, 0, 0, 1 ] ],
    [ [ 0, 1, 0, 0 ], [ 0, 0, 1, 0 ], [ 0, 0, 0, 1 ] ], [ [ 0, 0, 0, 1 ] ],
    [ ] ]

#I
#I Compute the composition series ...
#I finished.
#I The composition series has length 5.
#I The composition series is
#I [ [ [ 1, 0, 0, 0 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 0 ], [ 0, 0, 0, 1 ] ],
    [ [ 0, 1, 0, 0 ], [ 0, 0, 1, 0 ], [ 0, 0, 0, 1 ] ],
    [ [ 0, 0, 1, 0 ], [ 0, 0, 0, 1 ] ], [ [ 0, 0, 0, 1 ] ], [ ] ]

#I
#I Compute a constructive polycyclic sequence
    for the induced action of the kernel to the composition series ...
#I finished.
#I This polycyclic sequence has relative orders [ ].
#I
#I Calculate normal subgroup generators for the
    unipotent part ...
#I finished.
#I The normal subgroup generators for the unipotent part are
#I [ [ [ 1, -3/2, 0, 0 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 3 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, 0, 0, 24 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 0 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, 3, 3, 15 ], [ 0, 1, 0, 6 ], [ 0, 0, 1, -6 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, 3, 3, 9 ], [ 0, 1, 0, 6 ], [ 0, 0, 1, -6 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, 3/2, 3/2, 3/2 ], [ 0, 1, 0, 3 ], [ 0, 0, 1, -3 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, -3/2, 9/2, -69/2 ], [ 0, 1, 0, 9 ], [ 0, 0, 1, 3 ], [ 0, 0, 0, 1 ] ],
    , [ [ 1, 0, 0, -24 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 0 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, -3, -3, -9 ], [ 0, 1, 0, -6 ], [ 0, 0, 1, 6 ], [ 0, 0, 0, 1 ] ],
    [ [ 1, -3, -3, -15 ], [ 0, 1, 0, -6 ], [ 0, 0, 1, 6 ], [ 0, 0, 0, 1 ] ],

```

```

[ [ 1, -3, 0, 9 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 6 ], [ 0, 0, 0, 1 ] ],
[ [ 1, -3, -3, -9 ], [ 0, 1, 0, -6 ], [ 0, 0, 1, 6 ], [ 0, 0, 0, 1 ] ],
[ [ 1, -3, 0, 9 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 6 ], [ 0, 0, 0, 1 ] ],
[ [ 1, -3/2, -3/2, -9/2 ], [ 0, 1, 0, -3 ], [ 0, 0, 1, 3 ], [ 0, 0, 0, 1 ]
],
[ [ 1, -3, -3, -12 ], [ 0, 1, 0, -6 ], [ 0, 0, 1, 6 ], [ 0, 0, 0, 1 ] ],
[ [ 1, 3, -3/2, -21 ], [ 0, 1, 0, -3 ], [ 0, 0, 1, -6 ], [ 0, 0, 0, 1 ] ],
[ [ 1, 3/2, 3/2, 9/2 ], [ 0, 1, 0, 3 ], [ 0, 0, 1, -3 ], [ 0, 0, 0, 1 ] ] ]
#I
#I Determine a constructive polycyclic sequence
for the unipotent part ...
#I finished.
#I The unipotent part has relative orders
#I [ 0, 0, 0 ].
#I
#I ... computation of a constructive
polycyclic sequence for the whole group finished.
#I
#I Compute the relations of the polycyclic
presentation of the group ...
#I Compute power relations ...
.....
#I ... finished.
#I Compute conjugation relations ...
.....
#I ... finished.
#I Update polycyclic collector ...
#I ... finished.
#I finished.
#I
#I Construct the polycyclic presented group ...
#I finished.
#I
Pcp-group with orders [ 3, 2, 3, 3, 3, 0, 0, 0 ]

```

--

[http://www.icm.tu-bs.de/ag\\_algebra/software/assmann](http://www.icm.tu-bs.de/ag_algebra/software/assmann)

--

CompositionSeriesAbelianMatGroup  
CompositionSeriesTriangularizableMat  
Group

HomogeneousSeriesAbelianMatGroup  
HomogeneousSeriesTriangularizableMat  
Group

ImageElm  
ImagesRepresentative  
ImagesSet  
InfoPolenta

IsomorphismPcpGroup  
IsPolycyclicGroup  
IsSolvableGroup  
IsTriangularizableMatGroup

PcpGroupByMatGroup

PolExamples

RadicalSeriesSolvableMatGroup

SubgroupsUnipotentByAbelianByFinite