Nilmat

Computing with Nilpotent Matrix Groups

A GAP4 Package

Version 1.3

Alla Detinko

Department of Mathematics

National University of Ireland, Galway

Ireland

alla.detinko@nuigalway.ie

Bettina Eick

Institut Computational Mathematics

TU Braunschweig

38106 Braunschweig

Germany

beick@tu-bs.de

Dane Flannery

Department of Mathematics

National University of Ireland, Galway

Ireland

dane.flannery@nuigalway.ie

September 2017

Contents

1 Introduction

This package is for computing with nilpotent matrix groups over a field $\mathbb{F}$, where $\mathbb{F}$ is a finite field $GF(q)$ or the rational number field $\mathbb{Q}$.

Nilmat contains an implementation of algorithms developed over the past few years, available in theoretical form in the papers [DF04,DF05b,DF06,DF07]. The theory of nilpotent matrix groups is an essential part of linear group theory. Many structural and classification results for nilpotent linear groups are known (see e.g. [Sup76,Weh73]), and specialized methods for handling these groups have been developed. The computational advantages of nilpotent linear groups have been addressed in [DF05a]. For a full description of most of the algorithms of this package, further general information, and historical remarks, see [DF06,DF07].

One purpose of Nilmat is testing nilpotency of a subgroup G of $GL(n,\mathbb{F})$. If $G < GL(n,\mathbb{Q})$ is found to be nilpotent then the package provides a function for deciding whether G is finite. If $G < GL(n,q)$ is found to be nilpotent then the package provides a function that returns the Sylow subgroups of G. Additional functions allow one to test whether a nilpotent subgroup G of $GL(n,\mathbb{F})$ is completely reducible or unipotent, and to compute the order of G if it is finite.

Another feature of Nilmat is a library of nilpotent primitive matrix groups. Specifically, for each integer $n > 1$ and prime power q, this library returns a complete and irredundant list of $GL(n,q)$-conjugacy class representatives of the nilpotent primitive subgroups of $GL(n,q)$.

The problem of constructing nilpotent matrix groups is interesting in its own right. We have included in the package functions concerned with this problem. For example, one such function constructs maximal absolutely irreducible nilpotent subgroups of $GL(n,q)$.

Related research on solvable and polycyclic matrix groups was carried out by Björn Assmann and Bettina Eick in [AE05,AE07]. Most of the algorithms in [AE05] were implemented in the GAP package Polenta, on which Nilmat partially relies.

This work has emanated from research conducted with the financial support of Science Foundation Ireland and the German Academic Exchange Service (DAAD).

2 Computing with nilpotent linear groups

This chapter contains the main functions of this package for computing with nilpotent matrix groups.

2.1 Preliminaries

We first describe some of the basic functions used in Nilmat for nilpotency testing of a group G input by a finite generating set of matrices.

1 ► `JordanSplitting( G )` A

For a subgroup G of $GL(n,\mathbb{F})$, $\mathbb{F} = GF(q)$ or $\mathbb{Q}$, returns a list of two groups $[S,U]$, where S is the semisimple part of G (the group generated by the semisimple parts of the generators of G), and U is the unipotent part of G (the group generated by the unipotent parts of the generators of G). If G is nilpotent, then $G \cong S \times U$, the group S is completely reducible and U is unipotent. This attribute relies on the GAP attribute `JordanDecomposition`.

2 ► `IsUnipotentMatGroup( G )` P

For a subgroup G of $GL(n,\mathbb{F})$, $\mathbb{F} = GF(q)$ or $\mathbb{Q}$, returns `true` if G is unipotent (i.e. conjugate to a group of upper unitriangular matrices) and `false` otherwise.

3 ► `ClassLimit( n , F )` F

returns an upper bound on the nilpotency class of nilpotent subgroups of $GL(n,\mathbb{F})$, $\mathbb{F} = GF(q)$ or $\mathbb{Q}$.

4 ► `AbelianNormalSeries( G, l )` F

Here $G < GL(n,q)$ and l is a positive integer. If G is nilpotent of class at most l and the order of G is coprime to the characteristic of $GF(q)$, then this function determines a normal series with abelian factors for G. Otherwise, the function may still return such a series or it may return `fail`. The function is based on recursively selecting non-central elements from the second centers of terms in the abelian series.

5 ► `PiPrimarySplitting( G )` A

For a subgroup G of $GL(n,q)$, this function returns a list of two subgroups $[B,C]$ with $G = BC$. If G is nilpotent, then $G \cong B \times C$, the group C is the product of all Sylow p-subgroups with $p > n$ and B is the product of all other Sylow subgroups of G.

2.2 Testing nilpotency

The following is one of the main functions of the Nilmat package.

1 ▶ `IsNilpotentMatGroup( G )` F

For a subgroup G of $GL(n,\mathbb{F})$, $\mathbb{F} = GF(q)$ or $\mathbb{Q}$, returns `true` if G is nilpotent and `false` otherwise. This function is also installed as method for the property `IsNilpotentGroup`.

We include a brief description of the algorithm behind this function. Let X be a generating set of the given group G. The first stage of testing nilpotency of G is reduction to the semisimple part S of G. The procedure for reducing to the semisimple case is based on the Nilmat functions `JordanSplitting` and `IsUnipotentGroup` described in the previous section. In the following, we assume that all elements of X are semisimple matrices.

If $\mathbb{F} = GF(q)$, then we apply the function `PiPrimarySplitting` to the group S and thus reduce to a smaller group B. Next, we attempt to compute an abelian normal series for B using the function `AbelianNormalSeries`. If no such series exists, then G is not nilpotent. If such a series exists, then we use it to construct the Sylow subgroups of B and check that they commute pairwise. For details on this method, see [DF06].

If $\mathbb{F} = \mathbb{Q}$, then we first use a reduction mod p for a suitable prime p and check that the image of G under the corresponding congruence homomorphism is nilpotent using the finite field method above. If so, then we construct the kernel of the congruence homomorphism and test whether this is central in G. We refer to [DF08] for details. Note that the construction of the congruence homomorphism and its kernel is based on the methods of the Package Polenta; see also [AE05] for background.

The nilpotency testing functions of the package Nilmat have advantages over the standard GAP methods for `IsNilpotentGroup`. When $\mathbb{F}$ is finite, the Nilmat functions have better runtimes for all input groups we tested. When $\mathbb{F}$ is infinite, the standard GAP functions frequently do not terminate at all in sensible time; on the other hand, the Nilmat functions always terminate, with comparatively small runtimes (see the examples in Chapter 3).

2.3 Finiteness, Sylow subgroups, testing complete reducibility

The function `IsNilpotentMatGroup` determines various structural properties of the given group as by-products. The functions in this section have been designed to exploit these by-products.

1 ▶ `IsFiniteNilpotentMatGroup( G )` F

For a nilpotent subgroup G of $GL(n,\mathbb{Q})$, returns `true` if G is finite and `false` otherwise. Note that the function assumes that G is nilpotent and may return an incorrect result if not. The function exploits the by-products of the nilpotency testing functions in Nilmat and hence runs particularly fast (and usually faster than the standard GAP method for testing finiteness) if they have been used to check nilpotency. This function is also installed as method for the property `IsFinite`.

2 ▶ `SylowSubgroupsOfNilpotentFFMatGroup( G )` F

For a nilpotent subgroup G of $GL(n,\mathbb{F})$, $\mathbb{F} = GF(q)$, returns the list of all Sylow subgroups. The advantage of this function over the GAP function `SylowSubgroup` is that the former function returns all Sylow subgroups of G without first computing all prime divisors of the order of G. This function is installed as method for `SylowSystem` for nilpotent matrix groups.

3 ▶ `SizeOfNilpotentMatGroup( G )` F

For a finite nilpotent subgroup G of $GL(n,\mathbb{F})$, $\mathbb{F} = GF(q)$ or $\mathbb{Q}$, this function returns the order of G. The function is based on by-products of the nilpotency testing in Nilmat. Again, in some situations it is more efficient than the similar default GAP function; see the examples in Chapter 3.

4 ▶ `IsCompletelyReducibleNilpotentMatGroup( G )` F

For a nilpotent subgroup G of $GL(n,\mathbb{F})$, $\mathbb{F} = GF(q)$ or $\mathbb{Q}$, returns `true` if G is completely reducible and `false` otherwise.

2.4 A library of primitive nilpotent groups

Another main part of Nilmat is a library of nilpotent primitive matrix groups over finite fields.

1 ► `NilpotentPrimitiveMatGroups( n , p , l )` F

returns a complete and irredundant list L of the conjugacy class representatives of the nilpotent primitive subgroups of $GL(n,p^l)$. The list L contains non-abelian (i.e. non-cyclic) subgroups only if $n = 2m$, m is odd, and $p^l \equiv 3 \bmod 4$. Every non-abelian group in L is given by three generators. Note that the groups in L know their orders i.e. the attribute `Size` has been set for these groups.

2 ► `SizesOfNilpotentPrimitiveMatGroups( n , p , l )` F

returns the list of orders of groups in the list L output by `NilpotentPrimitiveMatGroups( n , p , l )`.

2.5 Further examples of nilpotent matrix groups

In this section we describe various functions designed to produce interesting examples of nilpotent matrix groups.

1 ► `MaximalAbsolutelyIrreducibleNilpotentMatGroup( n , p , l)` F

constructs the unique (up to conjugacy) maximal absolutely irreducible nilpotent subgroup of $GL(n,p^l)$ if such a group exists. Note that such a group exists if and only if each prime divisor of n divides $p^l - 1$. Otherwise the function returns `fail`.

2 ► `MonomialNilpotentMatGroup( n )` F

constructs an example of a finite nilpotent monomial subgroup of $GL(n,\mathbb{Q})$.

3 ► `ReducibleNilpotentMatGroup( m, k, [p, l] )` F

constructs an example of a reducible but not completely reducible nilpotent subgroup of $GL(mk,\mathbb{F})$, where $\mathbb{F} = \mathbb{Q}$ if there are two arguments given and $\mathbb{F} = GF(p^l)$ if there are four arguments given.

3 Examples

In this chapter we give some examples of computing with the Package Nilmat.

3.1 Constructing some nilpotent matrix groups

```
gap> g1 := MaximalAbsolutelyIrreducibleNilpotentMatGroup(52,3,3);
<matrix group with 7 generators>
```

The group g1 is a subgroup of $GL(52, 3^3)$ generated by 7 matrices.

```
gap> g2 := MaximalAbsolutelyIrreducibleNilpotentMatGroup(180,11,2);
<matrix group with 41 generators>
```

The group g2 is a subgroup of $GL(180, 11^2)$ generated by 41 matrices.

```
gap> MaximalAbsolutelyIrreducibleNilpotentMatGroup(210,2,10);
fail
```

In this third example, absolutely irreducible nilpotent subgroups of $GL(210, 2^{10})$ do not exist, because the degree of the matrices and the field size are both even.

```
gap> g3 := MonomialNilpotentMatGroup(450);
<matrix group with 24 generators>
```

Here g3 is a monomial nilpotent subgroup of $GL(450, \mathbb{Q})$.

```
gap> g4 := ReducibleNilpotentReducibleMatGroup(3,180,11,2);
<matrix group with 82 generators>
```

Here g4 $< GL(540, 11^2)$ is the Kronecker product of a unipotent subgroup of $GL(3, 11^2)$ and the group g2.

```
gap> g5 := ReducibleNilpotentMatGroup(7,36);
<matrix group with 72 generators>
```

Here g5 $< GL(252, \mathbb{Q})$ is a reducible nilpotent group constructed as the Kronecker product of a unipotent subgroup of $GL(7, \mathbb{Q})$ with `MonomialNilpotentMatGroup(36)`.

3.2 Testing nilpotency and other functions

We now illustrate use of the functions IsNilpotentMatGroup, SylowSubgroupsOfNilpotentFFMatGroup, IsFiniteNilpotentMatGroup, SizeOfNilpotentMatGroup, and IsCompletelyReducibleNilpotentMatGroup.

```
gap> IsNilpotentMatGroup(GL(200,Rationals));
false

gap> IsNilpotentMatGroup(GL(150,11^3));
false

gap> g6 := MaximalAbsolutelyIrreducibleNilpotentMatGroup(127,2,7);
<matrix group with 3 generators>
gap> IsNilpotentMatGroup(g6);
true

gap> g7 := MonomialNilpotentMatGroup(350);
<matrix group with 6 generators>
gap> IsNilpotentMatGroup(g7);
true
gap> IsFiniteNilpotentMatGroup(g7);
true

gap> g8 := ReducibleNilpotentMatGroup(6,35);
<matrix group with 5 generators>
gap> IsNilpotentMatGroup(g8);
true
gap> IsFiniteNilpotentMatGroup(g8);
false

gap> g9 := ReducibleNilpotentMatGroup(2,36,5,2);
<matrix group with 21 generators>
gap> SylowSubgroupsOfNilpotentFFMatGroup(g9);
[ <matrix group with 5 generators>, <matrix group with 6
generators>, <matrix group with 1 generators> ]
gap> IsCompletelyReducibleNilpotentMatGroup(g9);
false

gap> g10 := MaximalAbsolutelyIrreducibleNilpotentMatGroup(24,5,2);
<matrix group with 17 generators>
gap> SizeOfNilpotentMatGroup(g10);
173946175488
gap> IsCompletelyReducibleNilpotentMatGroup(g10);
true

gap> g11 := MonomialNilpotentMatGroup(96);
<matrix group with 31 generators>
gap> SizeOfNilpotentMatGroup(g11);
6442450944
gap> IsCompletelyReducibleNilpotentMatGroup(g11);
true
```

3.3 Using the library of primitive nilpotent groups

This section gives examples of applying the functions from the Nilmat library of primitive nilpotent subgroups of $GL(n,q)$.

```
gap> L0 := NilpotentPrimitiveMatGroups(2,3,1);
[ Group([ [ [ 0*Z(3), Z(3)^0 ], [ Z(3)^0, Z(3)^0 ] ] ]),
  Group([ [ [ Z(3)^0, 0*Z(3) ], [ 0*Z(3), Z(3)^0 ] ],
      [ [ Z(3), Z(3)^0 ], [ Z(3), Z(3) ] ],
      [ [ Z(3)^0, 0*Z(3) ], [ 0*Z(3), Z(3) ] ] ]),
  Group([ [ [ Z(3)^0, 0*Z(3) ], [ 0*Z(3), Z(3)^0 ] ],
      [ [ 0*Z(3), Z(3)^0 ], [ Z(3), 0*Z(3) ] ],
      [ [ Z(3), Z(3) ], [ Z(3), Z(3)^0 ] ] ]) ]
gap> SizesOfNilpotentPrimitiveMatGroups(2,3,1);
[ 8, 8, 16 ]
gap> List(L0,Size);
[ 8, 8, 16 ]

gap> L1 := NilpotentPrimitiveMatGroups(2,2,10);;
gap> Length(L1);
40
gap> Size(L1[38]);
209715
gap> s := SizesOfNilpotentPrimitiveMatGroups(2,2,10);;
[ 5, 15, 25, 41, 55, 75, 123, 155,
165, 205, 275, 451, 465, 615, 775, 825, 1025, 1271, 1353, 1705,
2255, 2325, 3075, 3813, 5115, 6355, 6765, 8525, 11275, 13981,
19065, 25575, 31775, 33825, 41943, 69905, 95325, 209715,
349525, 1048575 ]

gap> L2 := NilpotentPrimitiveMatGroups(55,3,1);;
gap> Length(L2);
114

gap> L3 := NilpotentPrimitiveMatGroups(6,3,3);;
gap> Length(L3);
110

gap> L4 := NilpotentPrimitiveMatGroups(22,11,1);;
gap> Length(L3);
1002
```

The lists `L1` and `L2` contain only abelian groups, while `L3` and `L4` contain non-abelian nilpotent groups.

4 Installation

The Package Nilmat is a GAP code only package and requires no external binaries.

Once Nilmat is loaded, calls to the GAP functions `IsNilpotent`, `IsNilpotentGroup`, `SylowSubgroup`, and `SylowSystem` for subgroups of $GL(n,q)$, and calls to `IsNilpotent`, `IsNilpotentGroup`, and `IsFinite` for subgroups of $GL(n,\mathbb{Q})$, automatically switch to corresponding functions from Nilmat. Thus Nilmat should be disabled if one wishes to use the former GAP functions for matrix groups over $GF(q)$ or $\mathbb{Q}$.

For testing nilpotency and finiteness over $\mathbb{Q}$, the GAP package Polenta is also required. Note that `Nilmat` does not use functions from Polenta which depend on `KASH`. Hence to use Nilmat, `KASH` installation is not required, and all Nilmat functions run under both Windows and Linux.

Bibliography

[AE05] B. Assmann and B. Eick. Computing polycyclic presentations for polycyclic rational matrix groups. *J. Symbolic Comput.*, 40(6):1269–1284, 2005.

[AE07] B. Assmann and B. Eick. Testing polycyclicity of finitely generated rational matrix groups. *Math. Comp.*, 76:1669–1682, 2007.

[DF04] A. S. Detinko and D. L. Flannery. Classification of nilpotent primitive linear groups over finite fields. *Glasgow Math. J.*, 46:585–594, 2004.

[DF05a] A. S. Detinko and D. L. Flannery. Locally nilpotent linear groups. *Irish Math. Soc. Bull.*, (56):37–51, 2005.

[DF05b] A. S. Detinko and D. L. Flannery. Nilpotent primitive linear groups over finite fields. *Comm. Algebra*, 33:1–9, 2005.

[DF06] A. S. Detinko and D. L. Flannery. Computing in nilpotent matrix groups. *LMS J. Comput. Math.*, 9:104–134 (electronic), 2006.

[DF08] A. S. Detinko and D. L. Flannery. Algorithms for computing with nilpotent matrix groups over infinite domains. *J. Symbolic Comput.*, 43:43 (2008) 8–26, 2008.

[Sup76] D. A. Suprunenko. *Matrix Groups*. Transl. Math. Monogr., vol. 45. American Mathematical Society, Providence, RI, 1976.

[Weh73] B. A. F. Wehrfritz. *Infinite Linear Groups*. Springer-Verlag, Berlin, Heidelberg, New York, 1973.

Index

This index covers only this manual. A page number in *italics* refers to a whole section which is devoted to the indexed subject. Keywords are sorted with case and spaces ignored, e.g., "`PermutationCharacter`" comes before "permutation group".