Cubefree

Construction Algorithm for Cubefree Groups

A GAP4 Package

by

Heiko Dietrich

School of Mathematical Sciences

Monash University

Clayton VIC 3800
Australia

email: heiko.dietrich@monash.edu

September 2016

Contents

1 Introduction

1.1 Overview

This manual describes the Cubefree package, a GAP 4 package for constructing groups of cubefree order; i.e., groups whose order is not divisible by any third power of a prime.

The groups of squarefree order are known for a long time: Hoelder [Hoe95] investigated them at the end of the 19th century. Taunt [Tau55] has considered solvable groups of cubefree order, since he examined solvable groups with abelian Sylow subgroups. Cubefree groups in general are investigated firstly in [Die05], [DE05], and [DE12], and this package contains the implementation of the algorithms described there.

Some general approaches to construct groups of an arbitrarily given order are described in [BE99a], [BE99b], and [BEO02].

The main function of this package is a method to construct all groups of a given cubefree order up to isomorphism. The algorithm behind this function is described completely in [Die05] and [DE05]. It is a refinement of the methods of the GrpConst package which are described in [BE99c].

This main function needs a method to construct up to conjugacy the solvable cubefree subgroups of $\mathrm{GL}(2,p)$ coprime to p. We split this construction into the construction of reducible and irreducible subgroups of $\mathrm{GL}(2,p)$. To determine the irreducible subgroups we use the method described in [FO05] for which this package also contains an implementation. Alternatively, the IrredSol package [Hoe00] could be used for primes $p \leq 251$.

The algorithm of [FO05] requires a method to rewrite a matrix representation. We use and implement the method of [GH97] for this purpose.

One can modify the construction algorithm for cubefree groups to a very efficient algorithm to construct groups of squarefree order. This is already done in the SmallGroups library. Thus for the construction of groups of squarefree order it is more practical to use `AllSmallGroups` of the SmallGroups library.

A more detailed description of the implemented methods can be found in Chapter 2.

Chapter 3 explains how to install and load the Cubefree package.

1.2 Theoretical background

In this section we give a brief survey about the main algorithm which is used to construct groups of cubefree order: the Frattini extension method. For a by far more detailed description we refer to the above references; e.g. see the online version of [Die05].

Let G be a finite group. The Frattini subgroup $\Phi(G)$ is defined to be the intersection of all maximal subgroups of G. We say a group H is a Frattini extension by G if the Frattini factor $H/\Phi(H)$ is isomorphic to G. The Frattini factor of H is Frattini-free; i.e. it has a trivial Frattini subgroup. It is known that every prime divisor of $|H|$ is also a divisor of $|H/\Phi(H)|$. Thus the Frattini subgroup of a cubefree group has to be squarefree and, as it is nilpotent, it is a direct product of cyclic groups of prime order.

Hence in order to construct all groups of a given cubefree order n, say, one can, firstly, construct all Frattini-free groups of suitable orders and, secondly, compute all corresponding Frattini extensions of order n. A first fundamental result is that a group of cubefree order is either a solvable Frattini extension or a direct product of a $\mathrm{PSL}(2,r)$, $r > 3$

a prime, with a solvable Frattini extension. In particular, the simple groups of cubefree order are the groups $\mathrm{PSL}(2,r)$ with $r > 3$ a prime such that $r \pm 1$ is cubefree. As a nilpotent group is the direct product of its Sylow subgroups, it is straightforward to compute all nilpotent groups of a given cubefree order.

Another important result is that for a cubefree solvable Frattini-free group there is exactly one isomorphism type of suitable Frattini extensions, which restricts the construction of cubefree groups to the determination of cubefree solvable Frattini-free groups. This uniqueness of Frattini extensions is the main reason why the Frattini extension method works so efficiently in the cubefree case.

In other words, there is a one-to-one correspondence between the solvable cubefree groups of order n and *some* Frattini-free groups of order dividing n. This allows to count the number of isomorphism types of cubefree groups of a given order without constructing Frattini extensions.

In the remaining part of this section we consider the construction of the solvable Frattini-free groups of a given cubefree order up to isomorphism. Such a group is a split extension over its socle; i.e. over the product of its minimal normal subgroups. Let F be a solvable Frattini-free group of cubefree order with socle S. Then S is a (cubefree) direct product of cyclic groups of prime order and F can be written as $F = K \ltimes S$ where $K \leq \mathrm{Aut}(S)$ is determined up to conjugacy. In particular, K is a subdirect product of certain cubefree subgroups of groups of the type $\mathrm{GL}(2,p)$ or C_{p-1}. Hence in order to determine all possible subgroups K one can determine all possible projections from such a subgroup into the direct factors of the types $\mathrm{GL}(2,p)$ and C_{p-1}, and then form all subdirect products having these projections. The construction of these subdirect products is one of the most time-consuming parts in the Frattini extension method for cubefree groups.

2 Functionality of the Cubefree package

This chapter describes the methods available from the Cubefree package.

2.1 New methods

This section lists the implemented functions.

1 ► `ConstructAllCFGroups(` *order* `)` F

The input *order* has to be a positive cubefree integer. The output is a complete and irredundant list of isomorphism type representatives of groups of this size. If possible, the groups are given as pc groups and as permutations groups otherwise.

2 ► `ConstructAllCFSolvableGroups(` *order* `)` F

The input *order* has to be a positive cubefree integer. The output is a complete and irredundant list of isomorphism type representatives of solvable groups of this size. The groups are given as pc groups.

3 ► `ConstructAllCFNilpotentGroups(` *order* `)` F

The input *order* has to be a positive cubefree integer. The output is a complete and irredundant list of isomorphism type representatives of nilpotent groups of this size. The groups are given as pc groups.

4 ► `ConstructAllCFSimpleGroups(` *order* `)` F

The input *order* has to be a positive cubefree integer. The output is a complete and irredundant list of isomorphism type representatives of simple groups of this size. In particular, there exists either none or exactly one simple group of the given order.

5 ► `ConstructAllCFFrattiniFreeGroups(` *order* `)` F

The input *order* has to be a positive cubefree integer. The output is a complete and irredundant list of isomorphism type representatives of Frattini-free groups of this size.

6 ► `IsomorphismCubefreeGroups(` G`,` H`)` F

Returns an isomorphism between two cubefree groups G and H, if exists, and fail otherwise. It is assumed that the input groups are permutation groups or pc groups. The algorithm is currently efficient only for solvable input groups due to the lack of a constructive recognition algorithm for the simple factors PSL.

7 ► `IsIsomorphicCubefreeGroups(` G`,` H`)` F

Returns true/false, depending on whether two cubefree groups G and H are isomorphic. It is assumed that the input groups are permutation groups or pc groups.

8 ► `NumberCFGroups(` n`[,` *bool* `]` `)` F

The input n has to be a positive cubefree integer and the output is the number of all cubefree groups of order n. The SmallGroups library is used for squarefree orders, orders of the type p^2 and p^2q, and cubefree orders less than

50000. Only if *bool* is set to false, then only the squarefree orders and orders of the type p^2 and p^2q,are taken from the SmallGroups library.

9 ► `NumberCFSolvableGroups( n[, bool ] )` F

The input n has to be a positive cubefree integer and the output is the number of all cubefree solvable groups of order n. The SmallGroups library is used for squarefree orders, orders of the type p^2 and p^2q, and cubefree orders less than 50000. Only if *bool* is set to false, then only the squarefree orders and orders of the type p^2 and p^2q,are taken from the SmallGroups library.

10 ► `CountAllCFGroupsUpTo( n[, bool ])` F

The input is a positive integer n and the output is a list L of size n such that $L[i]$ contains the number of isomorphism types of groups of order i if i is cubefree and $L[i]$ is not bound, otherwise, $1 \leq i \leq n$. The SmallGroups library is used for squarefree orders, orders of the type p^2 and p^2q, and cubefree orders less than 50000. Only if *bool* is set to false, then only the squarefree orders and orders of the type p^2 and p^2q are taken from the SmallGroups library. This function was implemented only for experimental purposes and its implementation could be improved.

11 ► `CubefreeOrderInfo( n[, bool ] )` F

This function displays some (very vague) information about the complexity of the construction of the groups of (cubefree) order n. It returns the number of possible pairs *(a,b)* where a is the order of a Frattini-free group F with socle S of order b which has to be constructed in order to construct all groups of order n: In fact, for each of these pairs *(a,b)* one would have to construct up to conjugacy all subgroups of order *a/b* of Aut*(S)*. The sum of the numbers of these subgroups for all pairs *(a,b)* as above is the number of groups of order n. Thus the output of `CubefreeOrderInfo` is a trivial lower bound for the number of groups of order n. There is no additional information displayed if *bool* is set to false.

12 ► `CubefreeTestOrder( n )` F

The input has to be a cubefree integer between 1 and 50000. This function tests the functionality of Cubefree, i.e. functions (1)–(7), and compares it with the data of the SmallGroups library. It returns true if everything is okay, otherwise an error message will be displayed.

13 ► `IsCubeFreeInt( n )` P

The output is *true* if n is a cubefree integer and *false* otherwise.

14 ► `IsSquareFreeInt( n )` P

The output is *true* if n is a squarefree integer and *false* otherwise.

15 ► `IrreducibleSubgroupsOfGL( n, q )` O

The current version of this function allows only n=2. The input q has to be a prime-power $q = p^r$ with $p \geq 5$ a prime. The output is a list of all irreducible subgroups of $\mathrm{GL}(2,q)$ up to conjugacy.

16 ► `RewriteAbsolutelyIrreducibleMatrixGroup( G )` F

The input G has to be an absolutely irreducible matrix group over a finite field $\mathrm{GF}(q)$. If possible, the output is G rewritten over the subfield of $\mathrm{GF}(q)$ generated by the traces of the elements of G. If no rewriting is possible, then the input G is returned.

2.2 Comments on the implementation

This section provides some information about the implementations.

ConstructAllCFGroups

The function `ConstructAllCFGroups` constructs all groups of a given cubefree order up to isomorphism using the Frattini Extension Method as described in [Die05], [DE05], [BE99a], and [BE99b]. One step in the Frattini Extension Method is to compute Frattini extensions and for this purpose some already implemented methods of the required GAP package GrpConst are used.

Since `ConstructAllCFGroups` requires only some special types of irreducible subgroups of $\mathrm{GL}(2,p)$ (e.g. of cubefree order), it contains a modified internal version of `IrreducibleSubgroupsOfGL`. This means that the latter is not called explicitely by `ConstructAllCFGroups`.

ConstructAllCFSimpleGroups and ConstructAllCFNilpotentGroups

The construction of simple or nilpotent groups of cubefree order is rather easy, see [Die05] or [DE05]. In particular, the methods used in these cases are independent from the methods used in the general cubefree case.

CountAllCFGroupsUpTo

As described in [Die05] and [DE05], every cubefree group G has the form $G = A \times I$ where A is trivial or non-abelian simple and I is solvable. Further, there is a one-to-one correspondence between the solvable cubefree groups and *some* solvable Frattini-free groups. This one-to-one correspondence allows to count the number of groups of a given cubefree order without computing any Frattini extension. To reduce runtime, the computed irreducible and reducible subgroups of the general linear groups $\mathrm{GL}(2,p)$ and also the number of the computed solvable Frattini-free groups are stored during the whole computation. This is very memory consuming but reduces the runtime significantly. The alternative is to run a loop over `NumberCFGroups`. This function was implemented only for experimental purposes and its implementation could be improved.

IrreducibleSubgroupsOfGL

If the input is a matrix group over $\mathrm{GF}(q)$, then the algorithm needs to construct $\mathrm{GF}(q^3)$ or $\mathrm{GF}(q^6)$ internally.

RewriteAbsolutelyIrreducibleMatrixGroup

The function `RewriteAbsolutelyIrreducibleMatrixGroup` as described algorithmically in [GH97] is a probabilistic Las Vegas algorithm; it retries until a correct answer is returned. If the input is $G \leq \mathrm{GL}(d,p^r)$, then the expected runtime is $O(rd^3)$.

2.3 Comments on the efficiency

The package GrpConst contains several implementations of algorithms to construct groups of a given order. One of these algorithms is the Frattini extension method, see Chapter 1. The algorithm used in Cubefree is a modification of the Frattini extension method to the case of cubefree orders.

The advantage of this modification is that the isomorphism problem at the construction of Frattini extensions is solved completely on a theoretic level. Also, the construction of the Frattini-free groups up to isomorphism is reduced to the determination of certain subgroups of groups of the type $\mathrm{GL}(2,p)$ and C_{p-1}, p a prime, and to the construction of subdirect products of these subgroups. As this is exponential, this is a main bottleneck of the current implementation.

A modification of the Frattini extension method to squarefree orders yields a powerful construction algorithm for squarefree groups which is based on number theory only. An implementation of this algorithm can be found in the SmallGroups library. Thus for squarefree groups one should definitely use `AllSmallGroups` and `NumberSmallGroups` instead of the functions of Cubefree. The same holds for groups of order p^2 or p^2q.

Moreover, using the functionality of Cubefree, the SmallGroups library now contains all groups of cubefree order at most 50000. Hence, also in this case, one should prefer `AllSmallGroups` and `NumberSmallGroups` to access the data of the library directly.

For all other cubefree orders n one can try to use Cubefree to construct or count the corresponding groups. Note, that the success of these computations depends basically on the complexity and number theory of the prime-power factorization of n. For each prime p with $p^2 \mid n$ one might have to construct subgroups of $\mathrm{GL}(2,p)$ and subdirect products involving these subgroups. One can use the info class `InfoCF` to get some information during the computation. In order to construct subdirect products, we need a permutation representation of these matrix groups. To rewrite them at once, we compute a permutation representation of $\mathrm{GL}(2,p)$ and apply this isomorphism to the constructed subgroups. Unfortunately, this is quite time and memory consuming for bigger primes.

In other words, Cubefree can note handle *unreasonable* cubefree orders. To get a rough idea of the complexity of the computation of groups of order n and to get a trivial lower bound for the number of groups, one can use `CubefreeOrderInfo(n)`.

At the end of this section we consider the quotient $q(n)$ of `NumberSmallGroups(n)` and `CubefreeOrderInfo(n)` for cubefree $1 \leq n \leq 50000$. Although for most of these integers we have a small quotient, note that $q(n)$ seems to be unbounded in general. There are 41597 cubefree integers between 1 and 50000 and 26414 of these integers fulfill $q(n) = 1$. Moreover, 13065 of these integers fulfill $1 < q(n) < 5$ and the remaining 2118 integers have $5 \leq q(n) \leq 54$; e.g. $n = 2^2 \cdot 3 \cdot 5 \cdot 7^2 \cdot 13$ has $q(n) = 1221/23$.

2.4 An example session

In this section we outline some examples of applications of the methods described above. We included runtimes for all examples, but omitted the output in some cases, since it would be too long to be printed. The runtimes have been obtained on an Intel(R) Pentium(R) 4 CPU 3.00GHz PC running under Linux.

```
gap> n:=5^2*7*13^2*67^2*97*107;
1377938614325
gap> CubefreeOrderInfo(n,false);
12
gap> Length(ConstructAllCFGroups(n));time;
12
53111

gap> n:=19^2*23^2*29*37*73^2*107^2;
12501895704027377
gap> CubefreeOrderInfo(n,false);
24
gap> NumberCFGroups(n);time;
24
190536
gap> Length(ConstructAllCFGroups(n));time;
24
948319

gap> n:=5^2*13*23^2*43^2*191;
60716861075
gap> CubefreeOrderInfo(n,false);
16
gap> Length(ConstructAllCFGroups(n)); time;
16
29146
```

Now we compute some more data.

```
gap>  n:=2*2*3*11*17*67;
150348
gap> CubefreeOrderInfo(n,false);
20
gap> NumberCFGroups(n);time;
145
12073
gap> Length(ConstructAllCFGroups(n)); time;
145
20757
gap> NumberCFSolvableGroups(n);time;
144
11925
gap> Length(ConstructAllCFSolvableGroups(n)); time;
144
18893
gap> Length(ConstructAllCFFrattiniFreeGroups(n)); time;
109
14421
gap> Length(ConstructAllCFNilpotentGroups(n));time;
2
12
gap> Length(ConstructAllCFSimpleGroups(n));time;
1
8
```

We consider another example with some info class output.

```
gap> SetInfoLevel(InfoCF,1);
gap> ConstructAllCFGroups(4620);;time;
#I  Construct all groups of order 4620.
#I    Compute solvable Frattini-free groups of order 2310.
#I    Compute solvable Frattini-free groups of order 4620.
#I  Construct 138 Frattini extensions.
#I    Compute solvable Frattini-free groups of order 77.
#I  Construct 1 Frattini extensions.
#I    Compute solvable Frattini-free groups of order 7.
#I  Construct 1 Frattini extensions.
15501

gap> n:=101^2*97*37^2*29^2;
1139236591513
gap> CubefreeOrderInfo(n,false);
8
gap> NumberCFGroups(n);time;
8
36
gap> SetInfoLevel(InfoCF,1);
gap> ConstructAllCFGroups(n);time;
#I  Construct all groups of order 1139236591513.
#I    Compute solvable Frattini-free groups of order 10512181.
```

```
#I    Compute solvable Frattini-free groups of order 304853249.
#I    Compute solvable Frattini-free groups of order 388950697.
#I    Compute solvable Frattini-free groups of order 1061730281.
#I    Compute solvable Frattini-free groups of order 11279570213.
#I    Compute solvable Frattini-free groups of order 30790178149.
#I    Compute solvable Frattini-free groups of order 39284020397.
#I    Compute solvable Frattini-free groups of order 1139236591513.
#I  Construct 8 Frattini extensions.
[ <pc group of size 1139236591513 with 7 generators>,
  <pc group of size 1139236591513 with 7 generators>,
  <pc group of size 1139236591513 with 7 generators>,
  <pc group of size 1139236591513 with 7 generators>,
  <pc group of size 1139236591513 with 7 generators>,
  <pc group of size 1139236591513 with 7 generators>,
  <pc group of size 1139236591513 with 7 generators>,
  <pc group of size 1139236591513 with 7 generators> ]
1848
```

The last example considers the cubefree order less than 50000 for which the number of groups with this order is maximal: there are 3093 groups of order 44100.

```
gap> n:=2*2*3*3*5*5*7*7;
44100
gap> CubefreeOrderInfo(n,false);
100
gap> NumberCFSolvableGroups(n,false);time;
3087
572639
gap> Length(ConstructAllCFSolvableGroups(n)); time;
3087
843085
gap> NumberCFGroups(n,false);time;
3093
719245
gap> Length(ConstructAllCFGroups(n)); time;
3093
1016763
gap> Length(ConstructAllCFFrattiniFreeGroups(n)); time;
1305
504451
gap> Length(ConstructAllCFNilpotentGroups(n));time;
16
180
```

2.5 Accuracy check

We have compared the results of `ConstructAllCFGroups` with the library of cubefree groups of SmallGroups. Further, we compared the solvable groups constructed by `IrreducibleSubgroupsOfGL` with the library of IrredSol. We have also done random isomorphism tests to verify that the list of groups we computed is not redundant.

One can call the following test files. The first one constructs some groups of order at most 2000 and compares the results with the SmallGroups library:

```
RereadPackage("cubefree","tst/testQuick.g");
```

The command

```
RereadPackage("cubefree","tst/testBig.g");
```

constructs the solvable groups of a random cubefree (but not squarefree) order at most $2^{28} - 1$ and does a random isomorphism test. Depending on the chosen number, the computation might not terminate due to memory problems.

The following constructs the groups of three random cubefree orders less than 50000 compares the result with the SmallGroups library. Depending on the chosen orders, this may take a while:

```
RereadPackage("cubefree","tst/testSG.g");
```

The test file *testSGlong.g* constructs all cubefree groups of order at most 50000 compares the results with the SmallGroups library. There will be a positive progress report every 50th order so that you can abort the test whenever you want.

```
RereadPackage("cubefree","tst/testSGlong.g");
```

Three of these four test files use the function `CubefreeTestOrder`, see Section 2.1.

The last test file compares some results of `IrreducibleSubgroupsOfGL` with the database of IrredSol. This may take a while:

```
RereadPackage("cubefree","tst/testMat.g");
```

3 Installing and loading the Cubefree package

3.1 Installing the Cubefree package

The installation of the Cubefree package follows standard GAP rules, see also Chapter 76.1 in the GAP reference manual. So the standard method is to unpack the package into the `pkg` directory of your GAP distribution. This will create an `cubefree` subdirectory.

3.2 Loading the Cubefree package

To use the Cubefree Package you have to request it explicitly. This is done by calling `LoadPackage` like this:

```
gap> LoadPackage("Cubefree");
Loading Cubefree 1.16 ...

   - Construction Algorithm for Cubefree Groups, 1.16 -
   ---- Heiko Dietrich, heiko.dietrich@monash.edu -----
true
```

Bibliography

[BE99a] H. U. Besche and B. Eick. Construction of finite groups. *J. Symb. Comput.*, 27:387–404, 1999.

[BE99b] H. U. Besche and B. Eick. The groups of order at most 1000 except 512 and 768. *J. Symb. Comput.*, 27:405–413, 1999.

[BE99c] H. U. Besche and B. Eick. *GrpConst*, 1999. A GAP 4 package, see [GAP05].

[BEO02] H. U. Besche, B. Eick, and E. A. O'Brien. A Millennium Project: Constructing small groups. *Intern. J. Alg. and Comput.*, 12:623–644, 2002.

[DE05] H. Dietrich and B. Eick. On the groups of cube-free order. *J. Algebra*, 292:122–137, 2005.

[DE12] H. Dietrich and B. Eick. Addendum to "cubefree:on the groups of cube-free order". *J. Algebra*, 367:247–248, 2012.

[Die05] H. Dietrich. On the groups of cube-free order. Diploma thesis, TU Braunschweig, 2005.

[FO05] D. L. Flannery and E. A. O'Brien. Linear groups of small degree over finite fields. *Intern. J. Alg. Comput.*, 15:467–502, 2005.

[GAP05] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.4*, 2005.

`http://www.gap-system.org`.

[GH97] S. P. Glasby and R. B. Howlett. Writing representations over minimal fields. *Comm. Alg.*, 25:1703–1711, 1997.

[Hoe95] O. Hoelder. Die Gruppen mit quadratfreier Ordnung. *Nachr. Koenigl. Ges. Wiss. Goettingen Math.-Phys. K*, 1:211–229, 1895.

[Hoe00] B. Hoefling. *Irredsol*, 2000. A GAP 4 package, see [GAP05].

[Tau55] D. R. Taunt. Remarks on the Isomorphism Problem in Theories of Construction of finite Groups. *Proc. Cambridge Philos. Soc.*, 51:16–24, 1955.