SmallGrp
The GAP Small Groups Library
1.5.4

4 July 2024

Hans Ulrich Besche

Bettina Eick

Eamonn O'Brien

Bettina Eick

Email: beick@tu-bs.de

Homepage: http://www.iaa.tu-bs.de/beick

Address: Institut Analysis und Algebra
       TU Braunschweig
       Universitätsplatz 2
       D–38106 Braunschweig
       Germany

Eamonn O'Brien

Email: obrien@math.auckland.ac.nz

Homepage: https://www.math.auckland.ac.nz/~obrien

Address: Department of Mathematics
       University of Auckland
       Private Bag 92019
       Auckland
       New Zealand

Contents

Chapter 1

The Small Groups Library

1.1 Overview

The Small Groups library gives access to all groups of certain "small" orders. The groups are sorted by their orders and they are listed up to isomorphism; that is, for each of the available orders a complete and irredundant list of isomorphism type representatives of groups is given. Currently, the library contains the following groups:

- those of order at most 2000 except 1024 (423 164 062 groups);

- those of cubefree order at most 50 000 (395 703 groups);

- those of order p^7 for the primes $p = 3, 5, 7, 11$ (907 489 groups);

- those of order p^n for $n \leq 6$ and all primes p

- those of order $q^n \cdot p$ for q^n dividing 2^8, 3^6, 5^5 or 7^4 and all primes p with $p \neq q$;

- those of squarefree order;

- those whose order factorises into at most 3 primes.

The first three items in this list cover an explicit range of orders; the last four provide access to infinite families of groups having orders of certain types.

The library also has an identification function: it returns the library number of a given group. This function determines library numbers using invariants of groups. The function is available for all orders in the library except for the orders 512 and 1536 and except for the orders p^5, p^6 and p^7 above 2000.

The library is organised in 11 layers. Each layer contains the groups of certain orders and their corresponding group identification routines. It is possible to install the first n layers of the group library and the first m layers of the group identification for each $1 \leq m \leq n \leq 11$. This might be useful to save disk space. There is an extensive README file for the Small Groups library available in the `small` directory of the GAP distribution containing detailed information on the layers. A brief description of the layers is given here:

(1) the groups whose order factorises into at most 3 primes.

(2) the remaining groups of order at most 1000 except 512 and 768.

(3) the remaining groups of order $2^n \cdot p$ with $n \leq 8$ and p an odd prime.

(4) the remaining groups of order 5^5, 7^4 and of order $q^n \cdot p$ for q^n dividing 3^6, 5^5 or 7^4 and $p \neq q$ a prime.

(5) the remaining groups of order at most 2000 except 1024, 1152, 1536 and 1920.

(6) the groups of orders 1152 and 1920.

(7) the groups of order 512.

(8) the groups of order 1536.

(9) the remaining groups of order p^n for $4 \leq n \leq 6$.

(10) the remaining groups of cubefree order at most 50 000 and of squarefree order.

(11) the remaining groups of order p^7 for $p = 3, 5, 7, 11$.

The data in this library has been carefully checked and cross-checked. It is believed to be reliable. However, no absolute guarantees are given and users should, as always, make their own checks in critical cases.

The data occupies about 30 MB (storing over 400 million groups in about 200 megabits). The group identification occupies about 47 MB of which 18 MB is used for the groups in layer (6). More information on the Small Groups library can be found on http://www.icm.tu-bs.de/ag_algebra/software/small/

This library has been constructed by Hans Ulrich Besche, Bettina Eick and E. A. O'Brien. A survey on this topic and an account of the history of group constructions can be found in [BEO02]. Further detailed information on the construction of this library is available in [New77], [O'B90], [O'B91], [BE99a], [BE99b], [BE01], [BEO01], [EO99a], [EO99b], [NOV04], [Gir03], [DE05], [OV05]. The Small Groups library incorporates the GAP 3 libraries `TwoGroup` and `ThreeGroup`. The data from these libraries was directly included into the Small Groups library, and the ordering there was preserved. The Small Groups library replaces the Gap 3 library of solvable groups of order at most 100. However, both the organisation and data descriptions of these groups has changed in the Small Groups library.

As of version 1.4 of this library, the arrangement of groups is the same as in Magma, Version 2.23. In earlier releases of this library, the arrangement in orders p^7, $p = 3, 5, 7, 11$ disagreed. An attempt to fix this was instated on version 1.1 of this library, but a wrong permutation was used. If you would like to refer to index numbers for these orders in older versions of the library, see `SMALL_GROUPS_OLD_ORDER` (1.2.14)). The arrangement of all other orders has always agreed and has remained stable.

In version 1.5, the number of groups of order 1024 was corrected. For more information, refer to [Bur21].

1.2 Function Reference

1.2.1 SmallGroup (for group order and index)

▷ SmallGroup(*order, i*)   (function)

▷ SmallGroup(*pair*)   (function)

returns the `i`-th group of order `order` in the catalogue. If the group is solvable, it will be given as a PcGroup; otherwise it will be given as a permutation group. If the groups of order `order` are not installed, the function reports an error and enters a break loop.

```
——————————————— Example ———————————————
gap> G := SmallGroup( 60, 4 );
<pc group of size 60 with 4 generators>
gap> StructureDescription( G );
"C60"
gap> G := SmallGroup( 60, 5 );
Group([ (1,2,3,4,5), (1,2,3) ])
gap> StructureDescription( G );
"A5"
gap> G := SmallGroup( 768, 1000000 );
<pc group of size 768 with 9 generators>
gap> G := SmallGroup( [768, 1000000] );
<pc group of size 768 with 9 generators>
```

1.2.2 SmallGroupsAvailable

▷ SmallGroupsAvailable(`order`)   (function)

returns `true` if the library of groups of order `order` is installed, and `false` otherwise.

1.2.3 AllSmallGroups

▷ AllSmallGroups(`arg`)   (function)

returns all groups with certain properties as specified by `arg`. If `arg` is a number n, then this function returns all groups of order n. However, the function can also take several arguments which then must be organized in pairs `function` and `value`. In this case the first function must be `Size` (**Reference: Size**) and the first value an order or a range of orders. If value is a list then it is considered a list of possible function values to include. The function returns those groups of the specified orders having those properties specified by the remaining functions and their values.

Precomputed information is stored for the properties `IsAbelian` (**Reference: IsAbelian**), `IsNilpotentGroup` (**Reference: IsNilpotentGroup**), `IsSupersolvableGroup` (**Reference: IsSupersolvableGroup**), `IsSolvableGroup` (**Reference: IsSolvableGroup**), `RankPGroup` (**Reference: RankPGroup**), `PClassPGroup` (**Reference: PClassPGroup**), `LGLength` (**Reference: LGLength**), `FrattinifactorSize` and `FrattinifactorId` for the groups of order at most 2000 which have more than three prime factors, except those of order 512, 768, 1024, 1152, 1536, 1920 and those of order $p^n \cdot q > 1000$ with $n > 2$.

```
——————————————— Example ———————————————
gap> AllSmallGroups( 6 );
[ <pc group of size 6 with 2 generators>,
  <pc group of size 6 with 2 generators> ]
gap> AllSmallGroups( 60, IsNilpotentGroup );
[ <pc group of size 60 with 4 generators>,
  <pc group of size 60 with 4 generators> ]
```

1.2.4 OneSmallGroup

▷ OneSmallGroup(*arg*) (function)

   returns one group with certain properties as specified by *arg*. The permitted arguments are those supported by AllSmallGroups (1.2.3).

```
───────────────────── Example ─────────────────────
gap> G := OneSmallGroup( 6, IsAbelian );
<pc group of size 6 with 2 generators>
gap> StructureDescription( G );
"C6"
gap> G := OneSmallGroup( 6, IsAbelian, false );
<pc group of size 6 with 2 generators>
gap> StructureDescription( G );
"S3"
gap> G := OneSmallGroup( Size, [1..1000], IsSolvableGroup, false );
Group([ (1,2,3,4,5), (1,2,3) ])
```

1.2.5 NumberSmallGroups

▷ NumberSmallGroups(*order*) (function)
▷ NrSmallGroups(*order*) (function)

   returns the number of groups of order *order*.

```
───────────────────── Example ─────────────────────
gap> NumberSmallGroups( 512 );
10494213
gap> NumberSmallGroups( 2^8 * 23 );
1083472
gap> NumberSmallGroups( 4096 );
Error, the library of groups of size 4096 is not available
```

1.2.6 NumberSmallGroupsAvailable

▷ NumberSmallGroupsAvailable(*order*) (function)

   returns true if the number of groups of order *order* is known, and false otherwise.

```
───────────────────── Example ─────────────────────
gap> NumberSmallGroupsAvailable( 100 );
true
gap> NumberSmallGroups( 100 );
16
gap> NumberSmallGroupsAvailable( 4096 );
false
gap> NumberSmallGroups( 4096 );
Error, the library of groups of size 4096 is not available
```

1.2.7 SelectSmallGroups

▷ SelectSmallGroups(*argl, all, id*) (function)

universal function for 'AllSmallGroups', 'OneSmallGroup' and 'IdsOfAllSmallGroups'.

1.2.8 IdSmallGroup

▷ IdSmallGroup(*G*) (attribute)
▷ IdGroup(*G*) (attribute)

returns the library number of *G*; that is, the function returns a pair [*order, i*] where *G* is isomorphic to SmallGroup(*order, i*).

```
─── Example ───
gap> IdSmallGroup( GL( 2,3 ) );
[ 48, 29 ]
gap> IdSmallGroup( Group( (1,2,3,4),(4,5) ) );
[ 120, 34 ]
```

1.2.9 IdGroupsAvailable

▷ IdGroupsAvailable(*order*) (function)

returns true, if the identification routines for groups of order *order* are installed, otherwise returns false.

1.2.10 IdsOfAllSmallGroups

▷ IdsOfAllSmallGroups(*arg*) (function)

similar to AllSmallGroups but returns ids instead of groups. This may prevent workspace overflows, if a large number of groups are expected in the output.

```
─── Example ───
gap> IdsOfAllSmallGroups( 60, IsNilpotentGroup );
[ [ 60, 4 ], [ 60, 13 ] ]
gap> IdsOfAllSmallGroups( 60, IsNilpotentGroup, false );
[ [ 60, 1 ], [ 60, 2 ], [ 60, 3 ], [ 60, 5 ], [ 60, 6 ], [ 60, 7 ],
  [ 60, 8 ], [ 60, 9 ], [ 60, 10 ], [ 60, 11 ], [ 60, 12 ] ]
gap> IdsOfAllSmallGroups( Size, 60, IsSupersolvableGroup, true );
[ [ 60, 1 ], [ 60, 2 ], [ 60, 3 ], [ 60, 4 ], [ 60, 6 ], [ 60, 7 ],
  [ 60, 8 ], [ 60, 10 ], [ 60, 11 ], [ 60, 12 ], [ 60, 13 ] ]
```

1.2.11 IdGap3SolvableGroup

▷ IdGap3SolvableGroup(*G*) (attribute)
▷ Gap3CatalogueIdGroup(*G*) (attribute)

returns the catalogue number of *G* in the GAP 3 catalogue of solvable groups; that is, the function returns a pair [*order, i*] meaning that *G* is isomorphic to the group SolvableGroup(*order, i*) in GAP 3.

1.2.12 SmallGroupsInformation

▷ SmallGroupsInformation(*order*)   (function)

prints information on the groups of the specified order.

```
 ─────────────────────── Example ───────────────────────
  gap> SmallGroupsInformation( 32 );

    There are 51 groups of order 32.
    They are sorted by their ranks.
       1 is cyclic.
       2 - 20 have rank 2.
       21 - 44 have rank 3.
       45 - 50 have rank 4.
       51 is elementary abelian.

    For the selection functions the values of the following attributes
    are precomputed and stored:
       IsAbelian, PClassPGroup, RankPGroup, FrattinifactorSize and
       FrattinifactorId.

    This size belongs to layer 2 of the SmallGroups library.
    IdSmallGroup is available for this size.
```

1.2.13 UnloadSmallGroupsData

▷ UnloadSmallGroupsData()   (function)

GAP loads all necessary data from the library automatically, but it does not delete the data from the workspace again. Usually, this will be not necessary, since the data is stored in a compressed format. However, if a large number of groups from the library have been loaded, then the user might wish to remove the data from the workspace and this can be done by the above function call.

```
 ─────────────────────── Example ───────────────────────
  gap> UnloadSmallGroupsData();
```

1.2.14 SMALL_GROUPS_OLD_ORDER

▷ SMALL_GROUPS_OLD_ORDER   (global variable)

If set to true, then groups of order 3^7, 5^7, 7^7, and 11^7 are ordered in the way they were ordered up to version 1.0 of the package. If this variable is set to false, which is the default as of version 1.4, then a different ordering, that agrees with the one in Magma version 2.23, is used. The functions SMALLGP_PERMx, with $x = 3, 5, 7, 11$, give the old index position corresponding to a new index position. In releases 1.1–1.3 a misunderstood ordering, based on the old ordering and the permutations $(2, 30083)(3, 30084)(4, 30085)(5, 30086)$, $(2, 104599)(3, 104600)(4, 104601)(5, 104602)$, and $(2, 721053)(3, 721054)(4, 721055)(5, 721059)$ respectively were used.

References

[BE99a]   H. U. Besche and B. Eick.   Construction of finite groups.   *J. Symbolic Comput.*,
           27(4):387–404, 1999. 4

[BE99b]   H. U. Besche and B. Eick. The groups of order at most 1000 except 512 and 768. *J.
           Symbolic Comput.*, 27(4):405–413, 1999. 4

[BE01]   H. U. Besche and B. Eick. The groups of order $q^n \cdot p$. *Comm. Algebra*, 29(4):1759–1772,
           2001. 4

[BEO01]   H. U. Besche, B. Eick, and E. A. O'Brien. The groups of order at most 2000. *Electron.
           Res. Announc. Amer. Math. Soc.*, 7:1–4 (electronic), 2001. 4

[BEO02]   H. U. Besche, B. Eick, and E. A. O'Brien. A millennium project: constructing small groups.
           Internat. J. Algebra Comput., 12(5):623–644, 2002. 4

[Bur21]   D. Burrell. On the number of groups of order 1024. *Communications in Algebra*, page 1–3,
           2021. 4

[DE05]   H. Dietrich and B. Eick. On the groups of cube-free order. *J. Algebra*, 292(1):122–137,
           2005. 4

[EO99a]   B. Eick and E. A. O'Brien.   Enumerating p-groups.   *J. Austral. Math. Soc. Ser. A*,
           67(2):191–205, 1999. Group theory. 4

[EO99b]   B. Eick and E. A. O'Brien. The groups of order 512. In B. H. Matzat, G.-M. Greuel,
           and G. Hiss, editors, *Algorithmic algebra and number theory (Heidelberg, 1997)*, page
           379–380. Springer, Berlin, 1999. Proceedings of Abschlusstagung des DFG Schwerpunktes
           Algorithmische Algebra und Zahlentheorie in Heidelberg. 4

[Gir03]   B. Girnat. Klassifikation der Gruppen bis zur Ordnung p^5. Staatsexamensarbeit, TU Braun-
           schweig, Braunschweig, Germany, 2003. 4

[New77]   M. F. Newman. Determination of groups of prime-power order. In R. A. Bryce, J. Cossey,
           and M. F. Newman, editors, *Group theory (Proc. Miniconf., Australian Nat. Univ., Can-
           berra, 1975)*, volume 573 of *Lecture Notes in Math.*, pages 73–84. Lecture Notes in Math.,
           Vol. 573, Berlin, 1977. Springer. Lecture Notes in Mathematics, Vol. 573. 4

[NOV04]   M. F. Newman, E. A. O'Brien, and M. R. Vaughan-Lee. Groups and nilpotent Lie rings
           whose order is the sixth power of a prime. *J. Algebra*, 278(1):383–401, 2004. 4

[O'B90] E. A. O'Brien. The p-group generation algorithm. *J. Symbolic Comput.*, 9(5-6):677–698, 1990. Computational group theory, Part 1. 4

[O'B91] E. A. O'Brien. The groups of order 256. *J. Algebra*, 143(1):219–235, 1991. 4

[OV05] E. A. O'Brien and M. R. Vaughan-Lee. The groups with order p^7 for odd prime p. *J. Algebra*, 292(1):243–258, 2005. 4

Index