PrimGrp

GAP Primitive Permutation Groups Library

Version 3.3.2

27 October 2018

Alexander Hulpke
Colva Roney-Dougal
Christopher Russell

Alexander Hulpke Email: hulpke@math.colostate.edu
Homepage: http://www.math.colostate.edu/~hulpke
Address: Department of Mathematics
Colorado State University
Fort Collins, CO, 80523-1874, USA

Colva Roney-Dougal Email: colva@mcs.st-andrews.ac.uk
Homepage: http://www-groups.mcs.st-and.ac.uk/~colva/
Address: School of Mathematics and Statistics
University of St Andrews
North Haugh, St Andrews
Fife, KY16 9SXS Scotland

Christopher Russell Email: cr66@st-andrews.ac.uk
Address: School of Mathematics and Statistics
University of St Andrews
North Haugh, St Andrews
Fife, KY16 9SXS Scotland

Abstract

The GAP package PrimGrp provides the library of primitive permutation groups which includes, up to permutation isomorphism (i.e., up to conjugacy in the corresponding symmetric group), all primitive permutation groups of degree < 4096.

Copyright

Acknowledgements

The conversion of the GAP database of primitive permutation groups to a separate GAP package has been supported by the EPSRC Collaborative Computational Project EP/M022641/1 CoDiMa (CCP in the area of Computational Discrete Mathematics), http://www.codima.ac.uk/.

Colophon

Versions history:

- Version 3.1 - Autumn 2017.

Contents

Chapter 1

Primitive Permutation Groups

1.1 Primitive Permutation Groups

GAP contains a library of primitive permutation groups which includes, up to permutation isomorphism (i.e., up to conjugacy in the corresponding symmetric group), all primitive permutation groups of degree < 4096, calculated in [RD05] and [Qui11], in particular,

- the primitive permutation groups up to degree 50, calculated by C. Sims,
- the primitive groups with insoluble socles of degree < 1000 as calculated in [DM88],
- the solvable (hence affine) primitive permutation groups of degree < 256 as calculated by M. Short [Sho92],
- some insolvable affine primitive permutation groups of degree < 256 as calculated in [The97].
- The solvable primitive groups of degree up to 999 as calculated in [EH03].
- The primitive groups of affine type of degree up to 999 as calculated in [RDU03].

Not all groups are named, those which do have names use ATLAS notation. Not all names are necessary unique!

The list given in [RD05] is believed to be complete, correcting various omissions in [DM88], [Sho92] and [The97].

In detail, we guarantee the following properties for this and further versions (but *not* versions which came before GAP 4.2) of the library:

- All groups in the library are primitive permutation groups of the indicated degree.
- The positions of the groups in the library are stable. That is `PrimitiveGroup(n,nr)` will always give you a permutation isomorphic group. Note however that we do not guarantee to keep the chosen S_n-representative, the generating set or the name for eternity.
- Different groups in the library are not conjugate in S_n.
- If a group in the library has a primitive subgroup with the same socle, this group is in the library as well.

(Note that the arrangement of groups is not guaranteed to be in increasing size, though it holds for many degrees.)

The selection functions (see (**Reference: Selection Functions**)) for the primitive groups library are `AllPrimitiveGroups` and `OnePrimitiveGroup`. They obtain the following properties from the database without having to compute them anew:

`NrMovedPoints` (**Reference: NrMovedPoints for a list or collection of permutations**), `Size` (**Reference: Size**), `Transitivity` (**Reference: Transitivity for a group and an action domain**), `ONanScottType` (**Reference: ONanScottType**), `IsSimpleGroup` (**Reference: IsSimpleGroup**), `IsSolvableGroup` (**Reference: IsSolvableGroup**), and `SocleTypePrimitiveGroup` (**Reference: SocleTypePrimitiveGroup**).

(Note, that for groups of degree up to 2499, O'Nan-Scott types 4a, 4b and 5 cannot occur.)

1.1.1 PrimitiveGroup

▷ `PrimitiveGroup(`*`deg, nr`*`)` (function)

returns the primitive permutation group of degree *deg* with number *nr* from the list.

The arrangement of the groups of degrees not greater than 50 differs from the arrangement of primitive groups in the list of C. Sims, which was used in GAP 3. See `SimsNo` (1.2.2).

1.1.2 NrPrimitiveGroups

▷ `NrPrimitiveGroups(`*`deg`*`)` (function)

returns the number of primitive permutation groups of degree *deg* in the library.

Example

```
gap> NrPrimitiveGroups(25);
28
gap> PrimitiveGroup(25,19);
5^2:((Q(8):3)'4)
gap> PrimitiveGroup(25,20);
ASL(2, 5)
gap> PrimitiveGroup(25,22);
AGL(2, 5)
gap> PrimitiveGroup(25,23);
(A(5) x A(5)):2
```

1.1.3 AllPrimitiveGroups

▷ `AllPrimitiveGroups(`*`attr1, val1, attr2, val2, ...`*`)` (function)

This is a selection function which permits to select all groups from the Primitive Group Library that have a given set of properties. It accepts arguments as specified in Section (**Reference: Selection Functions**) of the GAP reference manual.

1.1.4 OnePrimitiveGroup

▷ `OnePrimitiveGroup(`*`attr1, val1, attr2, val2, ...`*`)` (function)

This is a selection function which permits to select at most one group from the Primitive Group Library that have a given set of properties. It accepts arguments as specified in Section (**Reference: Selection Functions**) of the GAP reference manual.

1.1.5 PrimitiveGroupsIterator

▷ `PrimitiveGroupsIterator(attr1, val1, attr2, val2, ...)` (function)

returns an iterator through `AllPrimitiveGroups(attr1,val1,attr2,val2,...)` without creating all these groups at the same time.

1.1.6 COHORTS_PRIMITIVE_GROUPS

▷ `COHORTS_PRIMITIVE_GROUPS` (global variable)

In [DM88] the primitive groups are sorted in "cohorts" according to their socle. For each degree less than 2500, the variable `COHORTS_PRIMITIVE_GROUPS` (1.1.6) contains a list of the cohorts for the primitive groups of this degree. Each cohort is represented by a list of length 2, the first entry specifies the socle type (see `SocleTypePrimitiveGroup` (**Reference: SocleTypePrimitiveGroup**)), the second entry listing the index numbers of the groups in this degree.

For example in degree 49, we have four cohorts with socles $(\mathbb{Z}/7\mathbb{Z})^2$, $L_2(7)^2$, A_7^2 and A_{49} respectively. the group `PrimitiveGroup(49,36)`, which is isomorphic to $(A_7 \times A_7) : 2^2$, lies in the third cohort with socle $(A_7 \times A_7)$.

Example

```
gap> COHORTS_PRIMITIVE_GROUPS[49];
[ [ rec( parameter := 7, series := "Z", width := 2 ),
      [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
          20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33 ] ],
  [ rec( parameter := [ 2, 7 ], series := "L", width := 2 ), [ 34 ] ],
  [ rec( parameter := 7, series := "A", width := 2 ), [ 35, 36, 37, 38 ] ],
  [ rec( parameter := 49, series := "A", width := 1 ), [ 39, 40 ] ] ]
```

1.2 Index numbers of primitive groups

1.2.1 PrimitiveIdentification

▷ `PrimitiveIdentification(G)` (attribute)

For a primitive permutation group for which an S_n-conjugate exists in the library of primitive permutation groups (see 1.1), this attribute returns the index position. That is `G` is conjugate to `PrimitiveGroup(NrMovedPoints(G),PrimitiveIdentification(G))`.

Methods only exist if the primitive groups library is installed.

Note: As this function uses the primitive groups library, the result is only guaranteed to the same extent as this library. If it is incomplete, `PrimitiveIdentification` might return an existing index number for a group not in the library.

Example

```
gap> PrimitiveIdentification(Group((1,2),(1,2,3)));
2
```

1.2.2 SimsNo

▷ SimsNo(*G*) (attribute)

If *G* is a primitive group of degree not greater than 50, obtained by PrimitiveGroup (1.1.1) (respectively one of the selection functions), then this attribute contains the number of the isomorphic group in the original list of C. Sims. (This is the arrangement as it was used in GAP 3.)

Example

```
gap> g:=PrimitiveGroup(25,2);
5^2:S(3)
gap> SimsNo(g);
3
```

As mentioned in the previous section, the index numbers of primitive groups in GAP are guaranteed to remain stable. (Thus, missing groups will be added to the library at the end of each degree.) In particular, it is safe to refer to a primitive group of type *deg*, *nr* in the GAP library.

1.2.3 PRIMITIVE_INDICES_MAGMA

▷ PRIMITIVE_INDICES_MAGMA (global variable)

The system Magma also provides a list of primitive groups (see [RDU03]). For historical reasons, its indexing up to degree 999 differs from the one used by GAP. The variable PRIMITIVE_INDICES_MAGMA (1.2.3) can be used to obtain this correspondence. The magma index number of the GAP group PrimitiveGroup(*deg*,*nr*) is stored in the entry PRIMITIVE_INDICES_MAGMA[*deg*][*nr*], for degree at most 999.

Vice versa, the group of degree *deg* with Magma index number *nr* has the GAP index Position(PRIMITIVE_INDICES_MAGMA[*deg*],*nr*), in particular it can be obtained by the GAP command

PrimitiveGroup(*deg*,Position(PRIMITIVE_INDICES_MAGMA[*deg*],*nr*));

Chapter 2

Irreducible Matrix Groups

2.1 Irreducible Solvable Matrix Groups

2.1.1 IrreducibleSolvableGroupMS

▷ `IrreducibleSolvableGroupMS(n, p, i)` (function)

This function returns a representative of the i-th conjugacy class of irreducible solvable subgroup of GL(n, p), where n is an integer > 1, p is a prime, and $p^n < 256$.

The numbering of the representatives should be considered arbitrary. However, it is guaranteed that the i-th group on this list will lie in the same conjugacy class in all future versions of GAP, unless two (or more) groups on the list are discovered to be duplicates, in which case `IrreducibleSolvableGroupMS` will return `fail` for all but one of the duplicates.

For values of n, p, and i admissible to `IrreducibleSolvableGroup` (2.1.6), `IrreducibleSolvableGroupMS` returns a representative of the same conjugacy class of subgroups of GL(n, p) as `IrreducibleSolvableGroup` (2.1.6). Note that it currently adds two more groups (missing from the original list by Mark Short) for $n = 2$, $p = 13$.

2.1.2 NumberIrreducibleSolvableGroups

▷ `NumberIrreducibleSolvableGroups(n, p)` (function)

This function returns the number of conjugacy classes of irreducible solvable subgroup of GL(n, p).

2.1.3 AllIrreducibleSolvableGroups

▷ `AllIrreducibleSolvableGroups(func1, val1, func2, val2, ...)` (function)

This function returns a list of conjugacy class representatives G of matrix groups over a prime field such that $f(G) = v$ or $f(G) \in v$, for all pairs (f, v) in (*func1*, *val1*), (*func2*, *val2*), ... The following possibilities for the functions f are particularly efficient, because the values can be read off the information in the data base: `DegreeOfMatrixGroup` (or `Dimension` (**Reference: Dimension**) or `DimensionOfMatrixGroup` (**Reference: DimensionOfMatrixGroup**)) for the linear

degree, `Characteristic` (**Reference: Characteristic**) for the field characteristic, `Size` (**Reference: Size**), `IsPrimitiveMatrixGroup` (or `IsLinearlyPrimitive`), and `MinimalBlockDimension>`.

2.1.4 OneIrreducibleSolvableGroup

▷ `OneIrreducibleSolvableGroup(`*`func1, val1, func2, val2, ...`*`)` (function)

This function returns one solvable subgroup G of a matrix group over a prime field such that $f(G) = v$ or $f(G) \in v$, for all pairs (f, v) in (*`func1`*, *`val1`*), (*`func2`*, *`val2`*), ... The following possibilities for the functions f are particularly efficient, because the values can be read off the information in the data base: `DegreeOfMatrixGroup` (or `Dimension` (**Reference: Dimension**) or `DimensionOfMatrixGroup` (**Reference: DimensionOfMatrixGroup**)) for the linear degree, `Characteristic` (**Reference: Characteristic**) for the field characteristic, `Size` (**Reference: Size**), `IsPrimitiveMatrixGroup` (or `IsLinearlyPrimitive`), and `MinimalBlockDimension>`.

2.1.5 PrimitiveIndexIrreducibleSolvableGroup

▷ `PrimitiveIndexIrreducibleSolvableGroup` (global variable)

This variable provides a way to get from irreducible solvable groups to primitive groups and vice versa. For the group G = `IrreducibleSolvableGroup( n, p, k )` and $d = p^n$, the entry `PrimitiveIndexIrreducibleSolvableGroup[d][i]` gives the index number of the semidirect product $p^n : G$ in the library of primitive groups.

Searching for an index in this list with `Position` (**Reference: Position**) gives the translation in the other direction.

2.1.6 IrreducibleSolvableGroup

▷ `IrreducibleSolvableGroup(`*`n, p, i`*`)` (function)

This function is obsolete, because for *`n`* $= 2$, *`p`* $= 13$, two groups were missing from the underlying database. It has been replaced by the function `IrreducibleSolvableGroupMS` (2.1.1). Please note that the latter function does not guarantee any ordering of the groups in the database. However, for values of *`n`*, *`p`*, and *`i`* admissible to `IrreducibleSolvableGroup`, `IrreducibleSolvableGroupMS` (2.1.1) returns a representative of the same conjugacy class of subgroups of GL(*`n`*, *`p`*) as `IrreducibleSolvableGroup` did before.

References

[DM88] J. D. Dixon and B. Mortimer. The primitive permutation groups of degree less than 1000. *Math. Proc. Cambridge Philos. Soc.*, 103(2):213–238, 1988. 4, 6

[EH03] B. Eick and B. Höfling. The solvable primitive permutation groups of degree at most 6560. *LMS J. Comput. Math.*, 6:29–39 (electronic), 2003. 4

[Qui11] M. Quick. The primitive permutation groups of degree less than 4096. *Comm. Algebra*, 39(10):3526–3546, 2011. 4

[RD05] C. M. Roney-Dougal. The primitive permutation groups of degree less than 2500. *J. Algebra*, 292(1):154–183, 2005. 4

[RDU03] C. M. Roney-Dougal and W. R. Unger. The affine primitive permutation groups of degree less than 1000. *J. Symbolic Comput.*, 35(4):421–439, 2003. 4, 7

[Sho92] M. W. Short. *The primitive soluble permutation groups of degree less than* 256, volume 1519 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1992. 4

[The97] H. Theißen. *Eine Methode zur Normalisatorberechnung in Permutationsgruppen mit Anwendungen in der Konstruktion primitiver Gruppen*. Dissertation, Rheinisch Westfälische Technische Hochschule, Aachen, Germany, 1997. 4

Index