

# Documentation for the GAP code file `OrbOrd.txt`

Alexander Bors, Michael Giudici and Cheryl E. Praeger\*

October 29, 2019

## Abstract

We give a comprehensive description of the functions and variables defined in the authors' GAP code file `OrbOrd.txt`, which serve mainly to compute (bounds on) the number of  $\text{Aut}(S)$ -orbits on  $S$ , or the set or number of element orders in  $S$  for nonabelian finite simple groups of Lie type  $S$ .

## 1 Introduction

### 1.1 What this code documentation is about

The GAP code file `OrbOrd.txt` was written during the authors' work on their paper [1], and it is available in text form on the first author's homepage under <https://alexanderbors.wordpress.com/sourcecode/orbord/source/>. It consists of various defined functions that are useful for computing (bounds on) the number of  $\text{Aut}(S)$ -orbits on  $S$ , or the set or number of element orders in  $S$ , where  $S$  is mostly assumed to be a finite simple group of Lie type. The file does, however, also contain an efficient algorithm for computing the number of element orders in  $\text{Sym}(n)$ , discussed in Section 2. The functions relating to Lie type groups are discussed in Section 3. For each defined function, we describe its syntax and what it outputs, and in many cases, we also give comments pertaining to the idea behind the algorithm or a source in the literature justifying it. There are also a few defined variables, for which we simply describe what their value is.

---

\*First author's address: Johann Radon Institute for Computational and Applied Mathematics (RICAM), Altenbergerstraße 69, 4040 Linz, Austria. E-mail: alexander.bors@ricam.oeaw.ac.at

Second and third author's address: The University of Western Australia, Centre for the Mathematics of Symmetry and Computation, 35 Stirling Highway, Crawley 6009, WA, Australia. E-mail: michael.giudici@uwa.edu.au and cheryl.praeger@uwa.edu.au

The first author is supported by the Austrian Science Fund (FWF), project J4072-N32 "Affine maps on finite groups". The second and third authors were supported by the Australian Research Council Discovery Project DP160102323.

2010 *Mathematics Subject Classification*: Primary: 20D60. Secondary: 20D05, 20D45.

*Key words and phrases*: Finite groups, Automorphism orbits, Element orders, Simple groups, Monster simple group

## 1.2 Some notation used throughout this documentation

- $\omega(G)$ : the number of  $\text{Aut}(G)$ -orbits on the finite group  $G$ ;
- $\text{Ord}(G)$ : the set of element orders in the finite group  $G$ ;
- $\text{o}(G)$ : the number of distinct element orders in the finite group  $G$  (i.e.,  $\text{o}(G) = |\text{Ord}(G)|$ );
- $\epsilon_\omega(S) := \frac{\log \log \omega(S)}{\log \log |S|}$ , where  $S$  is a nonabelian finite simple group;
- $\epsilon_q(S) := \frac{\log \log (\omega(S)/\text{o}(S)+3)}{\log \log |S|}$ , where  $S$  is a nonabelian finite simple group;
- $O^{p'}(G)$ : the subgroup of the finite group  $G$  generated by the elements whose order is a power of the prime  $p$ ;
- Let  $G = X_d(\overline{\mathbb{F}_p})$  be a simple linear algebraic group of adjoint type over the algebraic closure  $\overline{\mathbb{F}_p}$  of the finite (size  $p$ ) field  $\mathbb{F}_p$ , and let  $\sigma$  be a Lang-Steinberg endomorphism of  $G$  (i.e., a surjective algebraic group endomorphism of  $G$  with finite fixed point subgroup  $G_\sigma$ ). Moreover, let  $B$  be any  $\sigma$ -invariant Borel subgroup of  $X_d(\overline{\mathbb{F}_p})$ , and let  $T$  be any  $\sigma$ -invariant maximal torus of  $X_d(\overline{\mathbb{F}_p})$  contained in  $B$ . We now define two important parameters  $t = t(\sigma)$  and  $f = f(\sigma)$ :  $t$  is defined as the unique smallest positive integer (independent of the choice of  $B$  and  $T$ ) such that the  $t$ -th power of the map  $\sigma^*$  on the character group  $X(T)$  induced by  $\sigma$  is a positive integral multiple of  $\text{id}_{X(T)}$ , and  $f \in \mathbb{N}^+ / 2 = \{\frac{1}{2}, 1, \frac{3}{2}, \dots\}$  is such that  $\sigma^* = p^f \sigma_0$  with  $\sigma_0^t = \text{id}_{X(T)}$ ;  $f$  also does not depend on the choice of  $B$  and  $T$ . So  $p^f = q(\sigma)$  in the notation of [10], which is also a notation we will be using, and  $f \in \mathbb{N}^+$  unless  $X_d \in \{B_2, G_2, F_4\}$  and  $t = 2$  (in which case  $f$  is half of an odd positive integer, and accordingly,  $q = \sqrt{p^{2k+1}}$  for some  $k \in \mathbb{N}$ ). With these parameters  $t$  and  $f$  now defined, we denote the group  $O^{p'}(G_\sigma) = O^{p'}(\text{fix}(\sigma))$ , which is usually a nonabelian finite simple group, by  ${}^t X_d(q^t) = {}^t X_d(p^{ft})$ . For example,  $A_3(2) = \text{PSL}_4(2)$ ,  ${}^2 D_6(9) = \text{P}\Omega_{12}^-(3)$ , and  ${}^2 B_2(8)$  is the Suzuki group of order 29120. This notation is consistently reflected in the syntax of our functions for Lie type groups.

## 1.3 Tips on how to read this documentation

While the documentation itself is rather long, there is lots of repetition, due to the fact that the functions pertaining to groups of Lie type need to be programmed anew for each Lie family. In order for readers to quickly get a good impression of what this code can do for them, it is sufficient to read the following:

- Section 2, as well as Subsections 3.1 and 3.18, each of which contains unique functions without analogues in other parts of the documentation. Moreover, note that the functions described in Subsection 2.6 and Subsubsections 3.18.4 to 3.18.17 were written for very specific checks in proofs found in [1] and thus probably will not be needed and can be skipped by readers other than the authors of this documentation.

- Subsections 3.2 (to get an impression of the functions and their syntax for one family of classical Lie type groups) and 3.8 (the same for one family of exceptional Lie type groups, where mostly the syntax is different).

When using `OrbOrd.txt` to study finite simple Lie type groups in general, there are some subtle differences between Lie families to take note of, most notably the following:

- There are no functions of the form `NrConjugacyClassesX` where `X` is one of the strings "C", "D", "2D", "E6", "2E6" or "E7"; in those cases, one must use `NrConjugacyClassesBoundX`, which only outputs a lower bound on (not the precise value of) the respective conjugacy class number.
- For classical groups, some functions have an argument called "quality level". This is to be chosen from a certain finite set of positive integers and determines what kind of computations for reaching the desired output (a bound on a certain parameter) are chosen. Generally speaking, the higher the quality level, the better the result (i.e., the sharper the bound that is output), but also the more costly the computations are. Functions for exceptional Lie type groups do not have such a quality level, and there are also differences with regard to the available quality levels among classical groups. For example, the functions of the form `NrAutOrbitsBoundX` always have quality level 2, but sometimes 1 as well, available, and the functions of the form `NrElementOrdersBoundX` always have quality levels 1 and 2, but sometimes 3 as well, available.
- The function `NrElementOrdersSharplyDivisibleByPToEBoundX` is only defined when `X` is either "A", "2A", "D" or "2D". This is not because it has no analogues when `X` is "B" or "C", but because the authors only needed it in those cases for their work on [1].
- The groups  $S$  of type  $E_8$  are the only exceptional Lie type groups for which there is no result in the literature (known to the authors) that describes the set of element orders of  $S$ . Hence there are no functions `ElementOrdersE8` and `NrElementOrdersE8`; instead, we have a function called `NrElementOrdersBoundE8` which, when applied to a prime power  $q$ , outputs an upper bound on the number of element orders in  $E_8(q)$ .

## 2 Functions for computing the number of element orders in $\text{Sym}(n)$

Before we start to list the functions (and the one variable) relevant for this section, let us introduce some (nonstandard) notation and terminology. For  $k \in \mathbb{N}^+$ , we denote by  $p_k$  the  $k$ -th prime (so for example,  $p_1 = 2$ ,  $p_2 = 3$  and so on). Moreover, for a prime  $p$  and a positive integer  $n$ , we denote by

- $r_p(n)$  the number of unordered integer partitions of  $n$  into pairwise coprime prime powers greater than 1 such that the smallest prime base which occurs is  $p$ ;

- $r(n)$  the number of unordered integer partitions of  $n$  into pairwise coprime prime powers greater than 1 (so that  $r(n) = \sum_{p \leq n} r_p(n)$  where the index  $p$  ranges over primes).

For each positive integer  $n$ , we define the  $n$ -th partition number matrix to be the matrix with  $n$  rows and  $\max\{1, \pi(n)\}$  columns, where

$$\pi(n) = |\{p \mid p \text{ is prime and } p \leq n\}|,$$

whose  $(i, j)$ -th entry, for  $i \in \{1, \dots, n\}$  and  $j \in \{1, \dots, \pi(n)\}$ , is  $r_{p_j}(i)$ . So, for example, the first partition number matrix is the  $(1 \times 1)$ -matrix  $(0)$ , the second is the  $(2 \times 1)$ -matrix  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ , and the third is the  $(3 \times 2)$ -matrix

$$\begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

## 2.1 PrimesExtended

This is a variable, defined as a comprehensive list of the primes up to 25013 (the smallest prime larger than 25000).

## 2.2 NextPartitionNumberMatrix

- Syntax: `NextPartitionNumberMatrix(m)`,  $m$  being the  $n$ -th partition number matrix (in the above defined sense) for some  $n \in \mathbb{N}^+$ .
- Output: the  $(n + 1)$ -th partition number matrix.
- Comments: This algorithm uses the fact that the numbers  $r_p(k)$  satisfy the recursion

$$r_p(k) = \sum_{e=1}^{\lfloor \log_p(k) \rfloor} \begin{cases} \sum_{p < \ell \leq k, \ell \text{ prime}} r_\ell(k - p^e), & \text{if } k > p^e, \\ 1, & \text{if } k = p^e. \end{cases}$$

Technically speaking, the algorithm only works when  $m$  is the  $n$ -th partition number matrix for some  $n \leq 25012$ , but the argument range can be extended provided that the list `PrimesExtended` is extended accordingly.

## 2.3 PartitionNumberMatrix

- Syntax: `PartitionNumberMatrix(n)`,  $n \in \mathbb{N}^+$ .
- Output: the  $n$ -th partition number matrix.
- Comments: Simply applies `NextPartitionNumberMatrix` iteratively.

## 2.4 NrPartitionsIntoPairwiseCoprimeNontrivialPrimePowers

- Syntax: `NrPartitionsIntoPairwiseCoprimeNontrivialPrimePowers(n)`,  $n \in \mathbb{N}^+$ .
- Output: the number  $r(n)$  of (un)ordered integer partitions of  $n$  into pairwise coprime prime powers greater than 1.
- Comments: Computes the  $n$ -th partition number matrix and sums over the entries in its  $n$ -th row.

## 2.5 NrElementOrdersSym

- Syntax: `NrElementOrdersSym(n)`,  $n \in \mathbb{N}^+$ .
- Output: the number  $o(\text{Sym}(n))$  of element orders in  $\text{Sym}(n)$ .
- Comments: This uses the fact that  $o(\text{Sym}(n)) = 1 + \sum_{k=1}^n r(k)$ .

## 2.6 ConstantsInOmicronSymBoundList

- Syntax: `ConstantsInOmicronSymBoundList(n)`,  $n \in \mathbb{N}^+$ .
- Output: the list of the numbers  $c_k := \frac{\log o(\text{Sym}(k))}{\sqrt{k}}$  for  $k = 1, \dots, n$ .
- Comments: By definition,  $o(\text{Sym}(k)) = \exp(c_k \sqrt{k})$ . The authors conjecture that the maximum value of  $c_k$  for all  $k \in \mathbb{N}^+$  is attained at  $k = 66$ , and using this function, the authors verified this conjecture for  $k \leq 25000$ .

# 3 Functions for dealing with simple Lie type groups

## 3.1 General-purpose functions

### 3.1.1 nrDivisors

- Syntax: `nrDivisors(n)`,  $n \in \mathbb{N}^+$ .
- Output: the number of positive integer divisors of  $n$ .
- Comments: Rather than just calling `Length(DivisorsInt(n))`, which would involve computing the (possibly very long) list of all divisors of  $n$ , it calls `FactorsInt(n)` to compute the factorisation of  $n$  into pairwise coprime prime powers,  $n = p_1^{k_1} \cdots p_r^{k_r}$ , and then returns the product  $(k_1 + 1)(k_2 + 1) \cdots (k_r + 1)$ .

### 3.1.2 lcm

- Syntax: `lcm(l)`,  $l$  a list of positive integers.
- Output: the least common multiple of the elements of  $l$ .
- Comments: Unlike GAP's built-in function `Lcm`, our function `lcm` returns 1 (not an error message) when applied to the empty list.

## 3.2 Functions for groups of type $A_d$

### 3.2.1 OrderA

- Syntax: `OrderA(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: the group order  $|A_d(q)| = |\mathrm{PSL}_{d+1}(q)|$ .

### 3.2.2 LogLogOrderA

- Syntax: `LogLogOrderA(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: the real number  $\log \log |A_d(q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(OrderA(d,q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

### 3.2.3 OrderOutA

- Syntax: `OrderOutA(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: the outer automorphism group order  $|\mathrm{Out}(A_d(q))|$ .

### 3.2.4 NrConjugacyClassesA

- Syntax: `NrConjugacyClassesA(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: the conjugacy class number  $k(A_d(q))$ .
- Comments: Simply calls `NrConjugacyClassesPSL(d+1,q)`.

### 3.2.5 NrAutOrbitsBoundA

- Syntax: `NrAutOrbitsBoundA(d,q,l)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \neq 2$ , it returns the message “This quality level is not available. Please set the quality level to 2.” as a string. If  $l = 2$ , it returns the lower bound  $\underline{\omega}^{(2)}(A_d(q)) := \lceil \frac{k(A_d(q))}{|\mathrm{Out}(A_d(q))|} \rceil$  on  $\omega(A_d(q))$ .

### 3.2.6 epsilonOmegaBoundA

- Syntax: `epsilonOmegaBoundA(d,q,l)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \neq 2$ , it returns the message “This quality level is not available. Please set the quality level to 2.” as a string. If  $l = 2$ , it returns the lower bound  $\underline{\epsilon}_\omega^{(2)}(A_d(q)) := \frac{\log \log \underline{\omega}^{(2)}(A_d(q))}{\log \log |A_d(q)|}$  on  $\epsilon_\omega(A_d(q))$ .

**3.2.7 CoxeterNoA**

- Syntax: `CoxeterNoA(d)`,  $d \in \mathbb{N}^+$ .
- Output: the Coxeter number  $h(A_d) = d + 1$ .

**3.2.8 SemisimpleElementOrdersA**

- Syntax: `SemisimpleElementOrdersA(d,q)`,  $d \in \mathbb{N}^+$ ,  $q = p^f$  a prime power.
- Output: the set of semisimple (viz., not divisible by  $p$ ) element orders in  $A_d(q)$ .
- Comments: It joins the sets of element orders of certain maximal tori  $T$  of  $A_d(q)$ , using [5, Theorem 2.1] to compute the group exponent  $\text{Exp}(T)$ .

**3.2.9 NrSemisimpleElementOrdersA**

- Syntax: `NrSemisimpleElementOrdersA(d,q)`,  $d \in \mathbb{N}^+$ ,  $q = p^f$  a prime power.
- Output: the number  $o_{\text{ss}}(A_d(q)) =: \overline{o_{\text{ss}}}^{(2)}(A_d(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $A_d(q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrdersA(d,q))`.

**3.2.10 NrSemisimpleElementOrdersBoundA**

- Syntax: `NrSemisimpleElementOrdersBoundA(d,q)`,  $d \in \mathbb{N}^+$ ,  $q = p^f$  a prime power.
- Output: an upper bound  $\overline{o_{\text{ss}}}^{(1)}(A_d(q))$  on the number of semisimple (viz., not divisible by  $p$ ) element orders in  $A_d(q)$ .
- Comments: Like `SemisimpleElementOrdersA(d,q)`, it loops over certain maximal tori of  $A_d(q)$ , but it adds the numbers of element orders in the tori that it loops over, rather than joining the corresponding sets. This makes it faster than `SemisimpleElementOrdersA(d,q)`.

**3.2.11 SemisimpleElementOrdersGL**

- Syntax: `SemisimpleElementOrdersGL(n,q)`,  $n \in \mathbb{N}^+$ ,  $q = p^f$  a prime power.
- Output: the set  $\text{Ord}_{\text{ss}}(\text{GL}_n(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $\text{GL}_n(q)$ .
- Comments: Similar to `SemisimpleElementOrdersA(d,q)`, but simpler (because the formula for  $\text{Exp}(T)$  when  $T$  is a maximal torus of  $\text{GL}_n(q)$  is simpler than for maximal tori of  $A_d(q)$ ).

**3.2.12 NrElementOrdersSharplyDivisibleByPToEBoundA**

- Syntax: `NrElementOrdersSharplyDivisibleByPToEBoundA(d,q,e)`,  $d \in \mathbb{N}^+$ ,  $q = p^f$  a prime power,  $e \in \{0, 1, \dots, 1 + \lfloor \log_p(d) \rfloor\}$ .

- Output: an upper bound  $\overline{o_{\text{ss}}^{(e)}}(A_d(q))$  on the number  $o_{\text{ss}}^{(e)}(A_d(q))$  of element orders  $o$  in  $A_d(q)$  such that the largest power of  $p$  dividing  $o$  is  $p^e$ .
- Comments: For an explanation, see [1, Case (5) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

### 3.2.13 NrElementOrdersBoundA

- Syntax: `NrElementOrdersBoundA(d,q,1)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2, 3\}$ , it returns the message “This quality level is not available. Please set the quality level to 1, 2 or 3.” as a string. Otherwise, it returns a certain upper bound  $\overline{o}^{(l)}(A_d(q))$  on the number of element orders  $o(A_d(q))$ . The larger  $l$  is, the better is that bound, but also the more costly it is to compute.
- Comments: Write  $q = p^f$ . For  $l = 3$ , this function returns the sum of the numbers returned by `NrElementOrdersSharplyDivisibleByPToEBoundA(d,q,e)` where  $e = 0, 1, \dots, 1 + \lfloor \log_p(d) \rfloor$ . For  $l \in \{1, 2\}$ , it returns  $\overline{o_{\text{ss}}}^{(l)}(A_d(q)) \cdot (1 + \lfloor \log_p(d+1) \rfloor)$ , the product of an upper bound on the number of semisimple element orders  $o_{\text{ss}}(A_d(q))$  and the exact number of element orders in  $A_d(q)$  that are powers of  $p$ .

### 3.2.14 epsilonQBoundA

- Syntax: `epsilonQBoundA(d,q,l1,l2)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l_1, l_2 \in \mathbb{N}^+$  (the  $l_i$  are interpreted as “quality levels”).
- Output: If  $(l_1, l_2) \notin \{(2, 1), (2, 2), (2, 3)\}$ , then it returns the message “This combination of quality levels is not available. Please set the quality levels to (2, 1), (2, 2) or (2, 3).” as a string. Otherwise, it returns the lower bound

$$\underline{\epsilon}_q^{(l_1, l_2)}(A_d(q)) := \frac{\log \log \left( \frac{\overline{\omega}^{(l_1)}(A_d(q))}{\overline{\sigma}^{(l_2)}(A_d(q))} + 3 \right)}{\log \log |A_d(q)|}$$

on  $\epsilon_q(A_d(q))$  as a decimal floating-point number.

### 3.2.15 epsilonQBoundAd2

- Syntax: `epsilonQBoundAd2(d)`,  $d \in \mathbb{N}^+$  (the intended range of use is  $d \in \{54, \dots, 90\}$ ).
- Output: the real number

$$\frac{\log \log \left( \frac{2^{d-1}}{o_{\text{ss}}(A_{90}(2)) \cdot (1 + \lfloor \log_2(d+1) \rfloor)} + 3 \right)}{\log \log |A_d(2)|},$$

as a decimal floating-point number.

- Comments: If  $d \leq 90$ , the output is a lower bound on  $\epsilon_q(A_d(2))$ . It is used in [1, Case (3) of the proof of Theorem 1.1.3(5) in Subsection 3.3].



### 3.3 Functions for groups of type ${}^2A_d$

#### 3.3.1 Order2A

- Syntax: `Order2A(d,Q)`,  $d \in \mathbb{N}^+$ ,  $Q = q^2$  for a prime power  $q$ .
- Output: the group order  $|{}^2A_d(Q)| = |\text{PSU}_{d+1}(q)|$ .

#### 3.3.2 LogLogOrder2A

- Syntax: `LogLogOrder2A(d,Q)`,  $d \in \mathbb{N}^+$ ,  $Q = q^2$  for a prime power  $q$ .
- Output: the real number  $\log \log |{}^2A_d(Q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(Order2A(d,Q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

#### 3.3.3 OrderOut2A

- Syntax: `OrderOut2A(d,Q)`,  $d \in \mathbb{N}^+$ ,  $Q = q^2$  for a prime power  $q$ .
- Output: the outer automorphism group order  $|\text{Out}({}^2A_d(Q))|$ .

#### 3.3.4 NrConjugacyClasses2A

- Syntax: `NrConjugacyClasses2A(d,Q)`,  $d \in \mathbb{N}^+$ ,  $Q = q^2$  for a prime power  $q$ .
- Output: the conjugacy class number  $k({}^2A_d(Q))$ .
- Comments: Simply calls `NrConjugacyClassesPSU(d+1,q)`.

#### 3.3.5 NrAutOrbitsBound2A

- Syntax: `NrAutOrbitsBound2A(d,Q,l)`,  $d \in \mathbb{N}^+$ ,  $Q = q^2$  for a prime power  $q$ ,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \neq 2$ , it returns the message “This quality level is not available. Please set the quality level to 2.” as a string. If  $l = 2$ , it returns the lower bound  $\underline{\omega}^{(2)}({}^2A_d(Q)) := \lceil \frac{k({}^2A_d(Q))}{|\text{Out}({}^2A_d(Q))|} \rceil$  on  $\omega({}^2A_d(Q))$ .

#### 3.3.6 epsilonOmegaBound2A

- Syntax: `epsilonOmegaBound2A(d,Q,l)`,  $d \in \mathbb{N}^+$ ,  $Q = q^2$  for a prime power  $q$ ,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \neq 2$ , it returns the message “This quality level is not available. Please set the quality level to 2.” as a string. If  $l = 2$ , it returns the lower bound  $\underline{\epsilon}_\omega^{(2)}({}^2A_d(Q)) := \frac{\log \log \underline{\omega}^{(2)}({}^2A_d(Q))}{\log \log |{}^2A_d(Q)|}$  on  $\epsilon_\omega({}^2A_d(Q))$ .

### 3.3.7 SemisimpleElementOrders2A

- Syntax: `SemisimpleElementOrders2A(d,Q)`,  $d \in \mathbb{N}^+$ ,  $Q = p^{2f}$  a prime power square.
- Output: the set of semisimple (viz., not divisible by  $p$ ) element orders in  ${}^2A_d(Q)$ .
- Comments: It joins the sets of element orders of certain maximal tori  $T$  of  ${}^2A_d(Q)$ , using [5, Theorem 2.2] to compute the group exponent  $\text{Exp}(T)$ .

### 3.3.8 NrSemisimpleElementOrders2A

- Syntax: `NrSemisimpleElementOrders2A(d,Q)`,  $d \in \mathbb{N}^+$ ,  $Q = p^{2f}$  a prime power square.
- Output: the number  $\text{o}_{\text{ss}}({}^2A_d(Q)) =: \overline{\text{o}_{\text{ss}}^{(2)}}({}^2A_d(Q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  ${}^2A_d(Q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrders2A(d,Q))`.

### 3.3.9 NrSemisimpleElementOrdersBound2A

- Syntax: `NrSemisimpleElementOrdersBound2A(d,Q)`,  $d \in \mathbb{N}^+$ ,  $Q = p^{2f}$  a prime power square.
- Output: an upper bound  $\overline{\text{o}_{\text{ss}}^{(1)}}({}^2A_d(Q))$  on the number of semisimple (viz., not divisible by  $p$ ) element orders in  ${}^2A_d(Q)$ .
- Comments: Like `SemisimpleElementOrders2A(d,Q)`, it loops over certain maximal tori of  ${}^2A_d(Q)$ , but it adds the numbers of element orders in the tori that it loops over, rather than joining the corresponding sets. This makes it faster than `SemisimpleElementOrders2A(d,Q)`.

### 3.3.10 SemisimpleElementOrdersGU

- Syntax: `SemisimpleElementOrdersGU(n,q)`,  $n \in \mathbb{N}^+$ ,  $q = p^f$  a prime power.
- Output: the set  $\text{Ord}_{\text{ss}}(\text{GU}_n(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $\text{GU}_n(q)$ .
- Comments: Similar to `SemisimpleElementOrders2A(d,Q)`, but simpler (because the formula for  $\text{Exp}(T)$  when  $T$  is a maximal torus of  $\text{GU}_n(q)$  is simpler than for maximal tori of  ${}^2A_d(Q)$ ).

### 3.3.11 NrElementOrdersSharplyDivisibleByPToEBound2A

- Syntax: `NrElementOrdersSharplyDivisibleByPToEBound2A(d,Q,e)`,  $d \in \mathbb{N}^+$ ,  $Q = q^2$  for a prime power  $q$ ,  $e \in \{0, 1, \dots, 1 + \lfloor \log_p(d) \rfloor\}$ .
- Output: an upper bound  $\overline{\text{o}_{\text{ss}}^{(e)}}({}^2A_d(Q))$  on the number  $\text{o}_{\text{ss}}^{(e)}({}^2A_d(Q))$  of element orders  $o$  in  ${}^2A_d(Q)$  such that the largest power of  $p$  dividing  $o$  is  $p^e$ .
- Comments: For an explanation, see [1, Case (5) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

**3.3.12** NrElementOrdersBound2A

- Syntax: `NrElementOrdersBound2A(d, Q, l)`,  $d \in \mathbb{N}^+$ ,  $Q = q^2$  for a prime power  $q$ ,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2, 3\}$ , it returns the message “This quality level is not available. Please set the quality level to 1, 2 or 3.” as a string. Otherwise, it returns a certain upper bound  $\overline{o}^{(l)}(^2A_d(Q))$  on the number of element orders  $o(^2A_d(Q))$ . The larger  $l$  is, the better is that bound, but also the more costly it is to compute.
- Comments: Write  $Q = p^{2f}$ . For  $l = 3$ , this function returns the sum of the numbers returned by `NrElementOrdersSharplyDivisibleByPToEBound2A(d, Q, e)` where  $e = 0, 1, \dots, 1 + \lfloor \log_p(d) \rfloor$ . For  $l \in \{1, 2\}$ , it returns  $\overline{o}_{\text{ss}}^{(l)}(^2A_d(Q)) \cdot (1 + \lfloor \log_p(d + 1) \rfloor)$ , the product of an upper bound on the number of semisimple element orders  $o_{\text{ss}}(^2A_d(Q))$  and the exact number of element orders in  $^2A_d(Q)$  that are powers of  $p$ .

**3.3.13** epsilonQBound2A

- Syntax: `epsilonQBoundA(d, Q, l1, l2)`,  $d \in \mathbb{N}^+$ ,  $Q = q^2$  for a prime power  $q$ ,  $l_1, l_2 \in \mathbb{N}^+$  (the  $l_i$  are interpreted as “quality levels”).
- Output: If  $(l_1, l_2) \notin \{(2, 1), (2, 2), (2, 3)\}$ , then it returns the message “This combination of quality levels is not available. Please set the quality levels to (2, 1), (2, 2) or (2, 3).” as a string. Otherwise, it returns the lower bound

$$\underline{\epsilon}_q^{(l_1, l_2)}(^2A_d(Q)) := \frac{\log \log \left( \frac{\omega^{(l_1)}(^2A_d(Q))}{\overline{o}^{(l_2)}(^2A_d(Q))} + 3 \right)}{\log \log |^2A_d(Q)|}$$

on  $\epsilon_q(^2A_d(Q))$  as a decimal floating-point number.

**3.3.14** epsilonQBound2Ad4

- Syntax: `epsilonQBound2Ad4(d)`,  $d \in \mathbb{N}^+$  (the intended range of use is  $d \in \{54, \dots, 90\}$ ).
- Output: the real number

$$\frac{\log \log \left( \frac{2^d / (2 \cdot \gcd(d+1, 3)^2)}{o_{\text{ss}}(^2A_{90}(4)) \cdot (1 + \lfloor \log_2(d+1) \rfloor)} + 3 \right)}{\log \log |^2A_d(4)|},$$

as a decimal floating-point number.

- Comments: If  $d \leq 90$ , the output is a lower bound on  $\epsilon_q(^2A_d(4))$ . It is used in [1, Case (3) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

## 3.4 Functions for groups of type $B_d$

### 3.4.1 OrderB

- Syntax: `OrderB(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: the group order  $|B_d(q)| = |\mathrm{P}\Omega_{2d+1}(q)|$ .

### 3.4.2 LogLogOrderB

- Syntax: `LogLogOrderB(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: the real number  $\log \log |B_d(q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(OrderB(d,q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

### 3.4.3 OrderOutB

- Syntax: `OrderOutB(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: the outer automorphism group order  $|\mathrm{Out}(B_d(q))|$ .

### 3.4.4 NrConjugacyClassesB

- Syntax: `NrConjugacyClassesB(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: the conjugacy class number  $k(B_d(q)) =: \underline{k}^{(2)}(B_d(q))$ .
- Comments: Uses [8, Theorems 3.13(1) and 3.19(1)].

### 3.4.5 NrConjugacyClassesBoundB

- Syntax: `NrConjugacyClassesBoundB(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: the lower bound  $\underline{k}^{(1)}(B_d(q)) := \lceil \frac{q^d}{\gcd(2,q-1)} \rceil$  on  $k(B_d(q))$ .
- Comments: To see that this is indeed a lower bound on  $k(B_d(q))$ , use [8, Theorem 1.1(1)].

### 3.4.6 NrAutOrbitsBoundB

- Syntax: `NrAutOrbitsBoundB(d,q,l)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. Otherwise, it returns the lower bound  $\underline{\omega}^{(l)}(B_d(q)) := \lceil \frac{\underline{k}^{(l)}(B_d(q))}{|\mathrm{Out}(B_d(q))|} \rceil$  on  $\omega(B_d(q))$ .

### 3.4.7 epsilonOmegaBoundB

- Syntax: `epsilonOmegaBoundB(d,q,l)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. Otherwise, it returns the lower bound  $\underline{\epsilon}_\omega^{(l)}(B_d(q)) := \frac{\log \log \omega^{(l)}(B_d(q))}{\log \log |B_d(q)|}$  on  $\epsilon_\omega(B_d(q))$ .

### 3.4.8 CoxeterNoB

- Syntax: `CoxeterNoB(d)`,  $d \in \mathbb{N}^+$ .
- Output: the Coxeter number  $h(B_d) = 2d$ .

### 3.4.9 SemisimpleElementOrdersB

- Syntax: `SemisimpleElementOrdersB(d,q)`,  $d \in \mathbb{N}^+$ ,  $q = p^f$  a prime power.
- Output: the set of semisimple (viz., not divisible by  $p$ ) element orders in  $B_d(q)$ .
- Comments: It joins the sets of element orders of certain maximal tori  $T$  of  $B_d(q)$ , using [5, Theorems 3 and 4] to compute the group exponent  $\text{Exp}(T)$ .

### 3.4.10 NrSemisimpleElementOrdersB

- Syntax: `NrSemisimpleElementOrdersB(d,q)`,  $d \in \mathbb{N}^+$ ,  $q = p^f$  a prime power.
- Output: the number  $\text{o}_{\text{ss}}(B_d(q)) =: \overline{\text{o}}_{\text{ss}}^{(2)}(B_d(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $B_d(q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrdersB(d,q))`.

### 3.4.11 NrSemisimpleElementOrdersBoundB

- Syntax: `NrSemisimpleElementOrdersBoundB(d,q)`,  $d \in \mathbb{N}^+$ ,  $q = p^f$  a prime power.
- Output: an upper bound  $\overline{\text{o}}_{\text{ss}}^{(1)}(B_d(q))$  on the number of semisimple (viz., not divisible by  $p$ ) element orders in  $B_d(q)$ .
- Comments: Like `SemisimpleElementOrdersB(d,q)`, it loops over certain maximal tori of  $B_d(q)$ , but it adds the numbers of element orders in the tori that it loops over, rather than joining the corresponding sets. This makes it faster than `SemisimpleElementOrdersB(d,q)`.

### 3.4.12 SemisimpleElementOrdersGOOddDim

- Syntax: `SemisimpleElementOrdersGOOddDim(n,q)`,  $n \in 2\mathbb{N}^+ + 1$ ,  $q = p^f$  a prime power.
- Output: the set  $\text{Ord}_{\text{ss}}(\text{GO}_n(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $\text{GO}_n(q)$ .

- Comments: Similar to `SemisimpleElementOrdersB(d,q)`, but simpler (because the formula for  $\text{Exp}(T)$  when  $T$  is a maximal torus of  $\text{GO}_n(q)$  is simpler than for maximal tori of  $B_d(q)$ ).

### 3.4.13 `NrElementOrdersBoundB`

- Syntax: `NrElementOrdersBoundB(d,q,l)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. Otherwise, it returns a certain upper bound  $\overline{o}^{(l)}(B_d(q))$  on the number of element orders  $o(B_d(q))$ . The larger  $l$  is, the better is that bound, but also the more costly it is to compute.
- Comments: Write  $q = p^f$ . For  $l \in \{1, 2\}$ , the output is  $\overline{o}_{\text{ss}}^{(l)}(B_d(q)) \cdot (1 + \lceil \log_p(2d) \rceil)$ , the product of an upper bound on  $o_{\text{ss}}(B_d(q))$  and the exact number of element orders in  $B_d(q)$  that are powers of  $p$ .

### 3.4.14 `epsilonQBoundB`

- Syntax: `epsilonQBoundB(d,q,l1,l2)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l_1, l_2 \in \mathbb{N}^+$  (the  $l_i$  are interpreted as “quality levels”).
- Output: If  $(l_1, l_2) \notin \{(1, 1), (1, 2), (2, 1), (2, 2)\}$ , then it returns the message “This combination of quality levels is not available. Please set the quality levels to (1, 1), (1, 2), (2, 1) or (2, 2).” as a string. Otherwise, it returns the lower bound

$$\underline{\epsilon}_q^{(l_1, l_2)}(B_d(q)) := \frac{\log \log \left( \frac{\omega^{(l_1)}(B_d(q))}{\overline{o}^{(l_2)}(B_d(q))} + 3 \right)}{\log \log |B_d(q)|}$$

on  $\epsilon_q(B_d(q))$  as a decimal floating-point number.

### 3.4.15 `epsilonQBoundBd2`

- Syntax: `epsilonQBoundBd2(d)`,  $d \in \mathbb{N}^+$  (the intended range of use is  $d \in \{54, \dots, 90\}$ ).
- Output: the real number

$$\frac{\log \log \left( \frac{\omega^{(2)}(B_d(2))}{o_{\text{ss}}(B_{90}(2)) \cdot (1 + \lceil \log_2(2d) \rceil)} + 3 \right)}{\log \log |B_d(2)|},$$

as a decimal floating-point number.

- Comments: If  $d \leq 90$ , the output is a lower bound on  $\epsilon_q(B_d(2))$ . It is used in [1, Case (3) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

## 3.5 Functions for groups of type $C_d$

### 3.5.1 OrderC

- Syntax: `OrderC(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: the group order  $|C_d(q)| = |\text{PSP}_{2d}(q)|$ .

### 3.5.2 LogLogOrderC

- Syntax: `LogLogOrderC(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: the real number  $\log \log |C_d(q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(OrderC(d,q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

### 3.5.3 OrderOutC

- Syntax: `OrderOutC(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: the outer automorphism group order  $|\text{Out}(C_d(q))|$ .

### 3.5.4 NrConjugacyClassesSp

- Syntax: `NrConjugacyClassesSp(n,q)`,  $n \in 2\mathbb{N}^+$ ,  $q$  a prime power.
- Output: the conjugacy class number  $k(\text{Sp}_n(q))$ .
- Comments: Uses [15, Case (B), statement (iii), p. 36] [8, Theorem 3.13(1)].

### 3.5.5 NrConjugacyClassesBoundC

- Syntax: `NrConjugacyClassesBoundC(d,q,l)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. If  $l = 2$ , it returns the lower bound  $\underline{k}^{(2)}(C_d(q)) := \lceil \frac{k(\text{Sp}_{2d}(q))}{\gcd(2, q-1)} \rceil$  on  $k(C_d(q))$ . If  $l = 1$ , it returns the lower bound  $\underline{k}^{(1)}(C_d(q)) := \lceil \frac{q^d}{\gcd(2, q-1)} \rceil$  on  $k(C_d(q))$ .
- Comments: To see that  $k(\text{Inndiag}(C_d(q))) \geq q^d$  (and thus that the output for  $l = 1$  is indeed a lower bound on  $k(C_d(q))$ ), use [8, Theorem 1.1(1)].

### 3.5.6 NrAutOrbitsBoundC

- Syntax: `NrAutOrbitsBoundC(d,q,l)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. Otherwise, it returns the lower bound  $\underline{\omega}^{(l)}(C_d(q)) := \lceil \frac{k^{(l)}(C_d(q))}{|\text{Out}(C_d(q))|} \rceil$  on  $\omega(C_d(q))$ .

### 3.5.7 epsilonOmegaBoundC

- Syntax: `epsilonOmegaBoundC(d,q,l)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. Otherwise, it returns the lower bound  $\underline{\epsilon}_\omega^{(l)}(C_d(q)) := \frac{\log \log \omega^{(l)}(C_d(q))}{\log \log |C_d(q)|}$  on  $\epsilon_\omega(C_d(q))$ .

### 3.5.8 CoxeterNoC

- Syntax: `CoxeterNoC(d)`,  $d \in \mathbb{N}^+$ .
- Output: the Coxeter number  $h(C_d) = 2d$ .

### 3.5.9 SemisimpleElementOrdersC

- Syntax: `SemisimpleElementOrdersC(d,q)`,  $d \in \mathbb{N}^+$ ,  $q = p^f$  a prime power.
- Output: the set of semisimple (viz., not divisible by  $p$ ) element orders in  $C_d(q)$ .
- Comments: By [5, Theorems 3 and 4], the sets of semisimple element orders in  $B_d(q)$  and  $C_d(q)$  respectively are equal. Hence this function simply calls `SemisimpleElementOrdersB(d,q)`.

### 3.5.10 NrSemisimpleElementOrdersC

- Syntax: `NrSemisimpleElementOrdersC(d,q)`,  $d \in \mathbb{N}^+$ ,  $q = p^f$  a prime power.
- Output: the number  $o_{\text{ss}}(C_d(q)) =: \overline{o}_{\text{ss}}^{(2)}(C_d(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $C_d(q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrdersC(d,q))`.

### 3.5.11 NrSemisimpleElementOrdersBoundC

- Syntax: `NrSemisimpleElementOrdersBoundC(d,q)`,  $d \in \mathbb{N}^+$ ,  $q = p^f$  a prime power.
- Output: an upper bound  $\overline{o}_{\text{ss}}^{(1)}(C_d(q))$  on the number of semisimple (viz., not divisible by  $p$ ) element orders in  $C_d(q)$ .
- Comments: Simply calls `NrSemisimpleElementOrdersBoundB(d,q)` (cf. also the comments in Subsection 3.5.9).

### 3.5.12 NrElementOrdersBoundC

- Syntax: `NrElementOrdersBoundC(d,q,l)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. Otherwise, it returns a certain upper bound  $\overline{o}^{(l)}(C_d(q))$  on the number of element orders  $o(C_d(q))$ . The larger  $l$  is, the better is that bound, but also the more costly it is to compute.



- Comments: Write  $q = p^f$ . For  $l \in \{1, 2\}$ , the output is  $\overline{o_{ss}^{(l)}}(C_d(q)) \cdot (1 + \lceil \log_p(2d) \rceil)$ , the product of an upper bound on  $o_{ss}(C_d(q))$  and the exact number of element orders in  $C_d(q)$  that are powers of  $p$ .

### 3.5.13 epsilonQBoundC

- Syntax: `epsilonQBoundC(d,q,l1,l2)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power,  $l_1, l_2 \in \mathbb{N}^+$  (the  $l_i$  are interpreted as “quality levels”).
- Output: If  $(l_1, l_2) \notin \{(1, 1), (1, 2), (2, 1), (2, 2)\}$ , then it returns the message “This combination of quality levels is not available. Please set the quality levels to (1, 1), (1, 2), (2, 1) or (2, 2).” as a string. Otherwise, it returns the lower bound

$$\epsilon_q^{(l_1, l_2)}(C_d(q)) := \frac{\log \log \left( \frac{\omega^{(l_1)}(C_d(q))}{\overline{o_{ss}^{(l_2)}}(C_d(q))} + 3 \right)}{\log \log |C_d(q)|}$$

on  $\epsilon_q(B_d(q))$  as a decimal floating-point number.

### 3.5.14 epsilonQBoundCd2

- Syntax: `epsilonQBoundCd2(d)`,  $d \in \mathbb{N}^+$  (the intended range of use is  $d \in \{54, \dots, 90\}$ ).
- Output: the real number

$$\frac{\log \log \left( \frac{\omega^{(2)}(C_d(2))}{o_{ss}(C_{90}(2)) \cdot (1 + \lceil \log_2(2d) \rceil)} + 3 \right)}{\log \log |C_d(2)|},$$

as a decimal floating-point number.

- Comments: If  $d \leq 90$ , the output is a lower bound on  $\epsilon_q(C_d(2))$ . It is used in [1, Case (3) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

## 3.6 Functions for groups of type $D_d$

### 3.6.1 OrderD

- Syntax: `OrderD(d,q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $q$  a prime power.
- Output: the group order  $|D_d(q)| = |\mathrm{P}\Omega_{2d}^+(q)|$ .

### 3.6.2 LogLogOrderD

- Syntax: `LogLogOrderD(d,q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $q$  a prime power.
- Output: the real number  $\log \log |D_d(q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(OrderD(d,q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

### 3.6.3 OrderOutD

- Syntax: `OrderOutD(d,q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $q$  a prime power.
- Output: the outer automorphism group order  $|\text{Out}(D_d(q))|$ .

### 3.6.4 NrConjugacyClassesSumOmega

- Syntax: `NrConjugacyClassesSumOmega(n,q)`,  $n \in 2\mathbb{N}^+ + 2$ ,  $q$  an *even* prime power.
- Output: the sum  $k(\Omega_n^+(q)) + k(\Omega_n^-(q))$ .
- Comments: See [8, Theorem 3.22(1)].

### 3.6.5 NrConjugacyClassesDifferenceOmega

- Syntax: `NrConjugacyClassesDifferenceOmega(n,q)`,  $n \in 2\mathbb{N}^+ + 2$ ,  $q$  an *even* prime power.
- Output: the difference  $k(\Omega_n^+(q)) - k(\Omega_n^-(q))$ .
- Comments: See [8, Theorem 3.22(2)].

### 3.6.6 NrConjugacyClassesSumSO

- Syntax: `NrConjugacyClassesSumSO(n,q)`,  $n \in 2\mathbb{N}^+ + 2$ ,  $q$  an *odd* prime power.
- Output: the sum  $k(\text{SO}_n^+(q)) + k(\Omega_n^-(q))$ .
- Comments: See [8, Theorem 3.16(1)].

### 3.6.7 NrConjugacyClassesDifferenceSO

- Syntax: `NrConjugacyClassesDifferenceSO(n,q)`,  $n \in 2\mathbb{N}^+ + 2$ ,  $q$  an *odd* prime power.
- Output: the sum  $k(\text{SO}_n^+(q)) - k(\Omega_n^-(q))$ .
- Comments: See [8, Theorem 3.16(2)].

### 3.6.8 NrConjugacyClassesSO

- Syntax: `NrConjugacyClassesSO(epsilon,n,q)`,  $\epsilon \in \{\pm 1\}$ ,  $n \in 2\mathbb{N}^+ + 2$ ,  $q$  an *odd* prime power.
- Output:  $k(\text{SO}_n^{\text{sign}(\epsilon)}(q))$ .

### 3.6.9 NrConjugacyClassesOmega

- Syntax: `NrConjugacyClassesOmega(epsilon,n,q)`,  $\epsilon \in \{\pm 1\}$ ,  $n \in 2\mathbb{N}^+ + 2$ ,  $q$  a prime power.
- Output:  $k(\Omega_n^{\text{sign}(\epsilon)}(q))$ .
- Comments: For odd  $q$ , this uses [8, Theorem 3.18(1) and the paragraph before Theorem 3.18].

### 3.6.10 NrConjugacyClassesBoundD

- Syntax: `NrConjugacyClassesBoundD(d,q,1)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. If  $l = 2$ , it returns the following lower bound  $\underline{k}^{(2)}(D_d(q))$  on  $k(D_d(q))$ :

$$\underline{k}^{(2)}(D_d(q)) := \begin{cases} k(\Omega_{2d}^+(q)), & \text{if } 2 \mid q, \\ \frac{1}{2} k(\text{SO}_{2d}^+(q)), & \text{if } 2 \nmid q, q \equiv 3 \pmod{4} \text{ and } 2 \nmid d, \\ \lceil \frac{1}{2} k(\Omega_{2d}^+(q)) \rceil, & \text{if } 2 \nmid q, \text{ and either } q \equiv 1 \pmod{4} \text{ or } 2 \mid d. \end{cases}$$

If  $l = 1$ , it returns the lower bound  $\underline{k}^{(1)}(D_d(q)) := \frac{q^d}{\gcd(2, q-1)^2}$  on  $k(D_d(q))$ .

### 3.6.11 NrAutOrbitsBoundD

- Syntax: `NrAutOrbitsBoundD(d,q,1)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. Otherwise, it returns the lower bound  $\underline{\omega}^{(l)}(D_d(q)) := \lceil \frac{k^{(l)}(D_d(q))}{|\text{Out}(D_d(q))|} \rceil$  on  $\omega(D_d(q))$ .

### 3.6.12 epsilonOmegaBoundD

- Syntax: `epsilonOmegaBoundD(d,q,1)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. Otherwise, it returns the lower bound  $\underline{\epsilon}_\omega^{(l)}(D_d(q)) := \frac{\log \log \omega^{(l)}(D_d(q))}{\log \log |D_d(q)|}$  on  $\epsilon_\omega(D_d(q))$ .

### 3.6.13 CoxeterNoD

- Syntax: `CoxeterNoD(d)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ .
- Output: the Coxeter number  $h(D_d) = 2d - 2$ .

### 3.6.14 SemisimpleElementOrdersD

- Syntax: `SemisimpleElementOrdersD(d,q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $q = p^f$  a prime power.
- Output: the set of semisimple (viz., not divisible by  $p$ ) element orders in  $D_d(q)$ .
- Comments: It joins the sets of element orders of certain maximal tori  $T$  of  $D_d(q)$ , using [5, Theorems 5 and 7] to compute the group exponent  $\text{Exp}(T)$ .

**3.6.15** `NrSemisimpleElementOrdersD`

- Syntax: `NrSemisimpleElementOrdersD(d,q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $q = p^f$  a prime power.
- Output: the number  $\text{o}_{\text{ss}}(D_d(q)) =: \overline{\text{o}_{\text{ss}}}^{(2)}(D_d(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $D_d(q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrdersD(d,q))`.

**3.6.16** `NrSemisimpleElementOrdersBoundD`

- Syntax: `NrSemisimpleElementOrdersBoundD(d,q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $q = p^f$  a prime power.
- Output: an upper bound  $\overline{\text{o}_{\text{ss}}}^{(1)}(D_d(q))$  on the number of semisimple (viz., not divisible by  $p$ ) element orders in  $D_d(q)$ .
- Comments: Like `SemisimpleElementOrdersD(d,q)`, it loops over certain maximal tori of  $D_d(q)$ , but it adds the numbers of element orders in the tori that it loops over, rather than joining the corresponding sets. This makes it faster than `SemisimpleElementOrdersD(d,q)`.

**3.6.17** `SemisimpleElementOrdersGOEvenDim`

- Syntax: `SemisimpleElementOrdersGOEvenDim(epsilon,n,q)`,  $\epsilon \in \{\pm 1\}$ ,  $n \in 2\mathbb{N}^+ + 2$ ,  $q = p^f$  a prime power.
- Output: the set  $\text{Ord}_{\text{ss}}(\text{GO}_n^{\text{sign}(\epsilon)}(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $\text{GO}_n^{\text{sign}(\epsilon)}(q)$ .

**3.6.18** `NrElementOrdersSharplyDivisibleByPToEBoundD`

- Syntax: `NrElementOrdersSharplyDivisibleByPToEBoundD(d,q,e)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $q = p^f$  a prime power,  $e \in \{0, 1, \dots, 1 + \lceil \log_p(2d - 3) \rceil\}$ .
- Output: an upper bound  $\overline{\text{o}_{\text{ss}}}^{(e)}(D_d(q))$  on the number  $\text{o}_{\text{ss}}^{(e)}(D_d(q))$  of element orders  $o$  in  $D_d(q)$  such that the largest power of  $p$  dividing  $o$  is  $p^e$ .
- Comments: For an explanation, see [1, Case (5) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

**3.6.19** `NrElementOrdersBoundD`

- Syntax: `NrElementOrdersBoundD(d,q,l)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $q$  a prime power,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2, 3\}$ , it returns the message “This quality level is not available. Please set the quality level to 1, 2 or 3.” as a string. Otherwise, it returns a certain upper bound  $\overline{\text{o}}^{(l)}(D_d(q))$  on the number of element orders  $o(D_d(q))$ . The larger  $l$  is, the better is that bound, but also the more costly it is to compute.

- Comments: Write  $q = p^f$ . For  $l = 3$ , this function returns the sum of the numbers returned by `NrElementOrdersSharplyDivisibleByPToEBoundD(d, q, e)` where  $e = 0, 1, \dots, 1 + \lceil \log_p(2d-3) \rceil$ . For  $l \in \{1, 2\}$ , the output is  $\overline{o_{ss}^{(l)}}(D_d(q)) \cdot (1 + \lceil \log_p(2d-2) \rceil)$ , the product of an upper bound on  $o_{ss}(D_d(q))$  and the exact number of element orders in  $D_d(q)$  that are powers of  $p$ .

### 3.6.20 `epsilonQBoundD`

- Syntax: `epsilonQBoundD(d, q, l1, l2)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $q$  a prime power,  $l_1, l_2 \in \mathbb{N}^+$  (the  $l_i$  are interpreted as “quality levels”).
- Output: If  $(l_1, l_2) \notin \{1, 2\} \times \{1, 2, 3\}$ , then it returns the message “This combination of quality levels is not available. Please set the quality levels to (1, 1), (1, 2), (1, 3), (2, 1), (2, 2) or (2, 3).” as a string. Otherwise, it returns the lower bound

$$\epsilon_q^{(l_1, l_2)}(D_d(q)) := \frac{\log \log \left( \frac{\omega^{(l_1)}(D_d(q))}{\overline{o}^{(l_2)}(D_d(q))} + 3 \right)}{\log \log |D_d(q)|}$$

on  $\epsilon_q(D_d(q))$  as a decimal floating-point number.

### 3.6.21 `epsilonQBoundDd2`

- Syntax: `epsilonQBoundDd2(d)`,  $d \in \mathbb{N}^+$  (the intended range of use is  $d \in \{54, \dots, 90\}$ ).
- Output: the real number

$$\frac{\log \log \left( \frac{\omega^{(2)}(D_d(2))}{o_{ss}(D_{90}(2)) \cdot (1 + \lceil \log_2(2d-2) \rceil)} + 3 \right)}{\log \log |D_d(2)|},$$

as a decimal floating-point number.

- Comments: If  $d \leq 90$ , the output is a lower bound on  $\epsilon_q(D_d(2))$ . It is used in [1, Case (3) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

## 3.7 Functions for groups of type ${}^2D_d$

### 3.7.1 `Order2D`

- Syntax: `Order2D(d, q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $Q = q^2$  for a prime power  $q$ .
- Output: the group order  $|{}^2D_d(Q)| = |P\Omega_{2d}^-(q)|$ .

### 3.7.2 `LogLogOrder2D`

- Syntax: `LogLogOrder2D(d, Q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $Q = q^2$  for a prime power  $q$ .
- Output: the real number  $\log \log |{}^2D_d(Q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(Order2D(d, Q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

### 3.7.3 OrderOut2D

- Syntax: `OrderOut2D(d,Q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $Q = q^2$  a prime power.
- Output: the outer automorphism group order  $|\text{Out}({}^2D_d(Q))|$ .

### 3.7.4 NrConjugacyClassesBound2D

- Syntax: `NrConjugacyClassesBound2D(d,Q,l)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $Q = q^2$  for a prime power  $q$ ,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. If  $l = 2$ , it returns the following lower bound  $\underline{k}^{(2)}({}^2D_d(Q))$  on  $k({}^2D_d(Q))$ :

$$\underline{k}^{(2)}({}^2D_d(Q)) := \begin{cases} k(\Omega_{2d}^-(q)), & \text{if } 2 \mid q, \\ \frac{1}{2} k(\text{SO}_{2d}^-(q)), & \text{if } 2 \nmid q, \text{ and either } q \equiv 1 \pmod{4} \text{ or } 2 \mid d, \\ \lceil \frac{1}{2} k(\Omega_{2d}^-(q)) \rceil, & \text{if } 2 \nmid q, q \equiv 3 \pmod{4} \text{ and } 2 \nmid d. \end{cases}$$

If  $l = 1$ , it returns the lower bound  $\underline{k}^{(1)}({}^2D_d(Q)) := \frac{q^d}{\gcd(2, q-1)^2}$  on  $k({}^2D_d(Q))$ .

### 3.7.5 NrAutOrbitsBound2D

- Syntax: `NrAutOrbitsBound2D(d,Q,l)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $Q = q^2$  for a prime power  $q$ ,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. Otherwise, it returns the lower bound  $\underline{\omega}^{(l)}({}^2D_d(Q)) := \lceil \frac{k^{(l)}({}^2D_d(Q))}{|\text{Out}({}^2D_d(Q))|} \rceil$  on  $\omega({}^2D_d(Q))$ .

### 3.7.6 epsilonOmegaBound2D

- Syntax: `epsilonOmegaBound2D(d,Q,l)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $Q = q^2$  for a prime power  $q$ ,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2\}$ , it returns the message “This quality level is not available. Please set the quality level to 1 or 2.” as a string. Otherwise, it returns the lower bound  $\underline{\epsilon}_\omega^{(l)}({}^2D_d(Q)) := \frac{\log \log \underline{\omega}^{(l)}({}^2D_d(Q))}{\log \log |\text{Out}({}^2D_d(Q))|}$  on  $\epsilon_\omega({}^2D_d(Q))$ .

### 3.7.7 SemisimpleElementOrders2D

- Syntax: `SemisimpleElementOrders2D(d,Q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $Q = p^{2f}$  a prime power square.
- Output: the set of semisimple (viz., not divisible by  $p$ ) element orders in  ${}^2D_d(Q)$ .
- Comments: It joins the sets of element orders of certain maximal tori  $T$  of  ${}^2D_d(Q)$ , using [5, Theorems 6 and 7] to compute the group exponent  $\text{Exp}(T)$ .

### 3.7.8 NrSemisimpleElementOrders2D

- Syntax: `NrSemisimpleElementOrders2D(d,Q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $Q = p^{2f}$  a prime power square.
- Output: the number  $o_{\text{ss}}(^2D_d(Q)) =: \overline{o_{\text{ss}}^{(2)}}(^2D_d(Q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $^2D_d(Q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrders2D(d,Q))`.

### 3.7.9 NrSemisimpleElementOrdersBound2D

- Syntax: `NrSemisimpleElementOrdersBound2D(d,Q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $Q = p^{2f}$  a prime power square.
- Output: an upper bound  $\overline{o_{\text{ss}}^{(1)}}(^2D_d(Q))$  on the number of semisimple (viz., not divisible by  $p$ ) element orders in  $^2D_d(Q)$ .
- Comments: Like `SemisimpleElementOrders2D(d,Q)`, it loops over certain maximal tori of  $^2D_d(Q)$ , but it adds the numbers of element orders in the tori that it loops over, rather than joining the corresponding sets. This makes it faster than `SemisimpleElementOrders2D(d,Q)`.

### 3.7.10 NrElementOrdersSharplyDivisibleByPToEBound2D

- Syntax: `NrElementOrdersSharplyDivisibleByPToEBound2D(d,Q,e)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $Q = p^{2f}$  a prime power square,  $e \in \{0, 1, \dots, 1 + \lceil \log_p(2d-3) \rceil\}$ .
- Output: an upper bound  $\overline{o_{\text{ss}}^{(e)}}(^2D_d(Q))$  on the number  $o_{\text{ss}}^{(e)}(^2D_d(Q))$  of element orders  $o$  in  $^2D_d(Q)$  such that the largest power of  $p$  dividing  $o$  is  $p^e$ .
- Comments: For an explanation, see [1, Case (5) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

### 3.7.11 NrElementOrdersBound2D

- Syntax: `NrElementOrdersBound2D(d,Q,l)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $Q = q^2$  for a prime power  $q$ ,  $l \in \mathbb{N}^+$  ( $l$  is interpreted as a “quality level”).
- Output: If  $l \notin \{1, 2, 3\}$ , it returns the message “This quality level is not available. Please set the quality level to 1, 2 or 3.” as a string. Otherwise, it returns a certain upper bound  $\overline{o}^{(l)}(^2D_d(Q))$  on the number of element orders  $o(^2D_d(Q))$ . The larger  $l$  is, the better is that bound, but also the more costly it is to compute.
- Comments: Write  $q = p^f$ . For  $l = 3$ , this function returns the sum of the numbers returned by `NrElementOrdersSharplyDivisibleByPToEBound2D(d,Q,e)` where  $e = 0, 1, \dots, 1 + \lceil \log_p(2d-3) \rceil$ . For  $l \in \{1, 2\}$ , the output is  $\overline{o_{\text{ss}}^{(l)}}(^2D_d(Q)) \cdot (1 + \lceil \log_p(2d-2) \rceil)$ , the product of an upper bound on  $o_{\text{ss}}(^2D_d(Q))$  and the exact number of element orders in  $^2D_d(Q)$  that are powers of  $p$ .

### 3.7.12 epsilonQBound2D

- Syntax: `epsilonQBound2D(d,Q,l1,l2)`,  $d \in \mathbb{N}^+$ ,  $d \geq 2$ ,  $Q = q^2$  for a prime power  $q$ ,  $l_1, l_2 \in \mathbb{N}^+$  (the  $l_i$  are interpreted as “quality levels”).
- Output: If  $(l_1, l_2) \notin \{1, 2\} \times \{1, 2, 3\}$ , then it returns the message “This combination of quality levels is not available. Please set the quality levels to (1, 1), (1, 2), (1, 3), (2, 1), (2, 2) or (2, 3).” as a string. Otherwise, it returns the lower bound

$$\epsilon_q^{(l_1, l_2)}({}^2D_d(Q)) := \frac{\log \log \left( \frac{\omega^{(l_1)}({}^2D_d(Q))}{\sigma^{(l_2)}({}^2D_d(Q))} + 3 \right)}{\log \log |{}^2D_d(Q)|}$$

on  $\epsilon_q({}^2D_d(Q))$  as a decimal floating-point number.

### 3.7.13 epsilonQBound2Dd4

- Syntax: `epsilonQBound2Dd4(d)`,  $d \in \mathbb{N}^+$  (the intended range of use is  $d \in \{54, \dots, 90\}$ ).
- Output: the real number

$$\frac{\log \log \left( \frac{\omega^{(2)}({}^2D_d(4))}{\text{oss}({}^2D_{90}(4)) \cdot (1 + \lceil \log_2(2d-2) \rceil)} + 3 \right)}{\log \log |{}^2D_d(4)|},$$

as a decimal floating-point number.

- Comments: If  $d \leq 90$ , the output is a lower bound on  $\epsilon_q({}^2D_d(4))$ . It is used in [1, Case (3) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

## 3.8 Functions for groups of type ${}^2B_2$

### 3.8.1 Order2B2

- Syntax: `Order2B2(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the group order  $|{}^2B_2(Q)|$ .

### 3.8.2 LogLogOrder2B2

- Syntax: `LogLogOrder2B2(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the real number  $\log \log |{}^2B_2(Q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(Order2B2(Q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

### 3.8.3 OrderOut2B2

- Syntax: `OrderOut2B2(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the outer automorphism group order  $|\text{Out}({}^2B_2(Q))|$ .



**3.8.4** `NrConjugacyClasses2B2`

- Syntax: `NrConjugacyClasses2B2(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the conjugacy class number  $k(^2B_2(Q))$ .
- Comments: See Lübeck's database [11].

**3.8.5** `NrAutOrbitsBound2B2`

- Syntax: `NrAutOrbitsBound2B2(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the lower bound  $\underline{\omega}({}^2B_2(Q)) := \lceil \frac{k({}^2B_2(Q))}{|{}^2B_2(Q)|} \rceil$  on  $\omega({}^2B_2(Q))$ .

**3.8.6** `epsilonOmegaBound2B2`

- Syntax: `epsilonOmegaBound2B2(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the lower bound  $\underline{\epsilon}_\omega({}^2B_2(Q)) := \frac{\log \log \omega({}^2B_2(Q))}{\log \log |{}^2B_2(Q)|}$  on  $\epsilon_\omega({}^2B_2(Q))$ .

**3.8.7** `SemisimpleElementOrders2B2`

- Syntax: `SemisimpleElementOrders2B2(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the set of semisimple (viz., odd) element orders in  ${}^2B_2(Q)$ .
- Comments: Based on [13, Theorem 2].

**3.8.8** `NrSemisimpleElementOrders2B2`

- Syntax: `NrSemisimpleElementOrders2B2(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the number  $o_{\text{ss}}({}^2B_2(Q))$  of semisimple (viz., odd) element orders in  ${}^2B_2(Q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrders2B2(Q))`.

**3.8.9** `ElementOrders2B2`

- Syntax: `ElementOrders2B2(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the set of element orders in  ${}^2B_2(Q)$ .
- Comments: Based on [13, Theorem 2].

**3.8.10** `NrElementOrders2B2`

- Syntax: `NrElementOrders2B2(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the number  $o({}^2B_2(Q))$  of element orders in  ${}^2B_2(Q)$ .
- Comments: Simply calls `Length(ElementOrders2B2(Q))`.

### 3.8.11 epsilonQBound2B2

- Syntax: `epsilonQBound2B2(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the lower bound  $\underline{\epsilon}_q(2B_2(Q)) := \frac{\log \log (\omega(2B_2(Q)) / \omega(2B_2(Q)) + 3)}{\log \log |2B_2(Q)|}$  on  $\epsilon_q(2B_2(Q))$ .

## 3.9 Functions for groups of type $G_2$

### 3.9.1 OrderG2

- Syntax: `OrderG2(q)`,  $q$  a prime power.
- Output: the group order  $|G_2(q)|$ .

### 3.9.2 LogLogOrderG2

- Syntax: `LogLogOrderG2(q)`,  $q$  a prime power.
- Output: the real number  $\log \log |G_2(q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(OrderG2(q)))`), because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

### 3.9.3 OrderOutG2

- Syntax: `OrderOutG2(q)`,  $q$  a prime power.
- Output: the outer automorphism group order  $|\text{Out}(G_2(q))|$ .

### 3.9.4 NrConjugacyClassesG2

- Syntax: `NrConjugacyClassesG2(q)`,  $q$  a prime power.
- Output: the conjugacy class number  $k(G_2(q))$ .
- Comments: See Lübeck's database [11].

### 3.9.5 NrAutOrbitsBoundG2

- Syntax: `NrAutOrbitsBoundG2(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\omega}(G_2(q)) := \lceil \frac{k(G_2(q))}{|\text{Out}(G_2(q))|} \rceil$  on  $\omega(G_2(q))$ .

### 3.9.6 epsilonOmegaBoundG2

- Syntax: `epsilonOmegaBoundG2(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\epsilon}_\omega(G_2(q)) := \frac{\log \log \omega(G_2(q))}{\log \log |G_2(q)|}$  on  $\epsilon_\omega(G_2(q))$ .

**3.9.7** CoxeterNoG2

- Syntax: `CoxeterNoG2()`.
- Output: the Coxeter number  $h(G_2) = 6$ .

**3.9.8** SemisimpleElementOrdersG2

- Syntax: `SemisimpleElementOrdersG2(q)`,  $q = p^f$  a prime power.
- Output: the set of semisimple (viz., not divisible by  $p$ ) element orders in  $G_2(q)$ .
- Comments: Based on [14, Lemma 1.4].

**3.9.9** NrSemisimpleElementOrdersG2

- Syntax: `NrSemisimpleElementOrdersG2(q)`,  $q = p^f$  a prime power.
- Output: the number  $o_{ss}(G_2(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $G_2(q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrdersG2(q))`.

**3.9.10** ElementOrdersG2

- Syntax: `ElementOrdersG2(q)`,  $q$  a prime power.
- Output: the set of element orders in  $G_2(q)$ .
- Comments: Based on [14, Lemma 1.4].

**3.9.11** NrElementOrdersG2

- Syntax: `NrElementOrdersG2(q)`,  $q$  a prime power.
- Output: the number  $o(G_2(q))$  of element orders in  $G_2(q)$ .
- Comments: Simply calls `Length(ElementOrdersG2(q))`.

**3.9.12** epsilonQBoundG2

- Syntax: `epsilonQBoundG2(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\epsilon}_q(G_2(q)) := \frac{\log \log (\omega(G_2(q))/o(G_2(q))+3)}{\log \log |G_2(q)|}$  on  $\epsilon_q(G_2(q))$ .

**3.10** Functions for groups of type  ${}^2G_2$ **3.10.1** Order2G2

- Syntax: `Order2G2(Q)`,  $Q = 3^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the group order  $|{}^2G_2(Q)|$ .

**3.10.2** LogLogOrder2G2

- Syntax: `LogLogOrder2G2(Q)`,  $Q = 3^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the real number  $\log \log |{}^2G_2(Q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(Order2G2(Q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

**3.10.3** OrderOut2G2

- Syntax: `OrderOut2G2(Q)`,  $Q = 3^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the outer automorphism group order  $|\text{Out}({}^2G_2(Q))|$ .

**3.10.4** NrConjugacyClasses2G2

- Syntax: `NrConjugacyClasses2G2(Q)`,  $Q = 3^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the conjugacy class number  $k({}^2G_2(Q))$ .
- Comments: See Lübeck's database [11].

**3.10.5** NrAutOrbitsBound2G2

- Syntax: `NrAutOrbitsBound2G2(Q)`,  $Q = 3^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the lower bound  $\underline{\omega}({}^2G_2(Q)) := \lceil \frac{k({}^2G_2(Q))}{|\text{Out}({}^2G_2(Q))|} \rceil$  on  $\omega({}^2G_2(Q))$ .

**3.10.6** epsilonOmegaBound2G2

- Syntax: `epsilonOmegaBound2G2(Q)`,  $Q = 3^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the lower bound  $\underline{\epsilon}_\omega({}^2G_2(Q)) := \frac{\log \log \omega({}^2G_2(Q))}{\log \log |{}^2G_2(Q)|}$  on  $\epsilon_\omega({}^2G_2(Q))$ .

**3.10.7** SemisimpleElementOrders2G2

- Syntax: `SemisimpleElementOrders2G2(Q)`,  $Q = 3^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the set of semisimple (viz., not divisible by 3) element orders in  ${}^2G_2(Q)$ .
- Comments: Based on [2, Lemma 4].

**3.10.8** NrSemisimpleElementOrders2G2

- Syntax: `NrSemisimpleElementOrders2G2(Q)`,  $Q = 3^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the number  $o_{\text{ss}}({}^2G_2(Q))$  of semisimple (viz., not divisible by 3) element orders in  ${}^2G_2(Q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrders2G2(Q))`.

**3.10.9** ElementOrders2G2

- Syntax: `ElementOrders2G2(Q)`,  $Q = 3^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the set of element orders in  ${}^2G_2(Q)$ .
- Comments: Based on [2, Lemma 4].

**3.10.10** NrElementOrders2G2

- Syntax: `NrElementOrders2G2(Q)`,  $Q = 3^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the number  $o({}^2G_2(Q))$  of element orders in  ${}^2G_2(Q)$ .
- Comments: Simply calls `Length(ElementOrders2G2(Q))`.

**3.10.11** epsilonQBound2G2

- Syntax: `epsilonQBound2G2(Q)`,  $Q = 3^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the lower bound  $\epsilon_q({}^2G_2(Q)) := \frac{\log \log (\omega({}^2G_2(Q))/o({}^2G_2(Q))+3)}{\log \log |{}^2G_2(Q)|}$  on  $\epsilon_q({}^2G_2(Q))$ .

**3.11** Functions for groups of type  ${}^3D_4$ **3.11.1** Order3D4

- Syntax: `Order3D4(Q)`,  $Q = q^3$  for some prime power  $q$ .
- Output: the group order  $|{}^3D_4(q)|$ .

**3.11.2** LogLogOrder3D4

- Syntax: `LogLogOrder3D4(Q)`,  $Q = q^3$  for some prime power  $q$ .
- Output: the real number  $\log \log |{}^3D_4(Q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(Order3D4(Q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

**3.11.3** OrderOut3D4

- Syntax: `OrderOut3D4(Q)`,  $Q = q^3$  for some prime power  $q$ .
- Output: the outer automorphism group order  $|\text{Out}({}^3D_4(q))|$ .

**3.11.4** NrConjugacyClasses3D4

- Syntax: `NrConjugacyClasses3D4(Q)`,  $Q = q^3$  for some prime power  $q$ .
- Output: the conjugacy class number  $k({}^3D_4(Q))$ .
- Comments: See Lübeck's database [11].

**3.11.5** NrAutOrbitsBound3D4

- Syntax: `NrAutOrbitsBound3D4(Q)`,  $Q = q^3$  for some prime power  $q$ .
- Output: the lower bound  $\underline{\omega}({}^3D_4(Q)) := \lceil \frac{k({}^3D_4(Q))}{|\text{Out}({}^3D_4(Q))|} \rceil$  on  $\omega({}^3D_4(Q))$ .

**3.11.6** epsilonOmegaBound3D4

- Syntax: `epsilonOmegaBound3D4(Q)`,  $Q = q^3$  for some prime power  $q$ .
- Output: the lower bound  $\underline{\epsilon_\omega}({}^3D_4(Q)) := \frac{\log \log \omega({}^3D_4(Q))}{\log \log |{}^3D_4(Q)|}$  on  $\epsilon_\omega({}^3D_4(Q))$ .

**3.11.7** SemisimpleElementOrders3D4

- Syntax: `SemisimpleElementOrders3D4(Q)`,  $Q = p^{3f}$  some prime power cube.
- Output: the set of semisimple (viz., not divisible by 3) element orders in  ${}^3D_4(Q)$ .
- Comments: Based on [9, Theorem 3.2].

**3.11.8** NrSemisimpleElementOrders3D4

- Syntax: `NrSemisimpleElementOrders3D4(Q)`,  $Q = p^{3f}$  some prime power cube.
- Output: the number  $o_{\text{ss}}({}^3D_4(Q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  ${}^3D_4(Q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrders3D4(Q))`.

**3.11.9** ElementOrders3D4

- Syntax: `ElementOrders3D4(Q)`,  $Q = q^3$  for some prime power cube.
- Output: the set of element orders in  ${}^3D_4(Q)$ .
- Comments: Based on [9, Theorem 3.2].

**3.11.10** NrElementOrders3D4

- Syntax: `NrElementOrders3D4(Q)`,  $Q = q^3$  for some prime power  $q$ .
- Output: the number  $o({}^3D_4(Q))$  of element orders in  ${}^3D_4(Q)$ .
- Comments: Simply calls `Length(ElementOrders3D4(Q))`.

**3.11.11** epsilonQBound3D4

- Syntax: `epsilonQBound3D4(Q)`,  $Q = q^3$  for some prime power  $q$ .
- Output: the lower bound  $\underline{\epsilon_q}({}^3D_4(Q)) := \frac{\log \log (\omega({}^3D_4(Q))/o({}^3D_4(Q))+3)}{\log \log |{}^3D_4(Q)|}$  on  $\epsilon_q({}^3D_4(Q))$ .

## 3.12 Functions for groups of type $F_4$

### 3.12.1 OrderF4

- Syntax: `OrderF4(q)`,  $q$  a prime power.
- Output: the group order  $|F_4(q)|$ .

### 3.12.2 LogLogOrderF4

- Syntax: `LogLogOrderF4(q)`,  $q$  a prime power.
- Output: the real number  $\log \log |F_4(q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(OrderF4(q))))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

### 3.12.3 OrderOutF4

- Syntax: `OrderOutF4(q)`,  $q$  a prime power.
- Output: the outer automorphism group order  $|\text{Out}(F_4(q))|$ .

### 3.12.4 NrConjugacyClassesF4

- Syntax: `NrConjugacyClassesF4(q)`,  $q$  a prime power.
- Output: the conjugacy class number  $k(F_4(q))$ .
- Comments: See Lübeck's database [11].

### 3.12.5 NrAutOrbitsBoundF4

- Syntax: `NrAutOrbitsBoundF4(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\omega}(F_4(q)) := \lceil \frac{k(F_4(q))}{|\text{Out}(F_4(q))|} \rceil$  on  $\omega(F_4(q))$ .

### 3.12.6 epsilonOmegaBoundF4

- Syntax: `epsilonOmegaBoundF4(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\epsilon}_\omega(F_4(q)) := \frac{\log \log \underline{\omega}(F_4(q))}{\log \log |F_4(q)|}$  on  $\epsilon_\omega(F_4(q))$ .

### 3.12.7 CoxeterNoF4

- Syntax: `CoxeterNoF4()`.
- Output: the Coxeter number  $h(F_4) = 12$ .

**3.12.8 SemisimpleElementOrdersF4**

- Syntax: `SemisimpleElementOrdersF4(q)`,  $q = p^f$  a prime power.
- Output: the set of semisimple (viz., not divisible by  $p$ ) element orders in  $F_4(q)$ .
- Comments: Based on [9, Theorem 3.1].

**3.12.9 NrSemisimpleElementOrdersF4**

- Syntax: `NrSemisimpleElementOrdersF4(q)`,  $q = p^f$  a prime power.
- Output: the number  $o_{\text{ss}}(F_4(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $F_4(q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrdersF4(q))`.

**3.12.10 ElementOrdersF4**

- Syntax: `ElementOrdersF4(q)`,  $q$  a prime power.
- Output: the set of element orders in  $F_4(q)$ .
- Comments: Based on [9, Theorem 3.1].

**3.12.11 NrElementOrdersF4**

- Syntax: `NrElementOrdersF4(q)`,  $q$  a prime power.
- Output: the number  $o(F_4(q))$  of element orders in  $F_4(q)$ .
- Comments: Simply calls `Length(ElementOrdersF4(q))`.

**3.12.12 epsilonQBoundF4**

- Syntax: `epsilonQBoundF4(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\epsilon}_q(F_4(q)) := \frac{\log \log (\omega(F_4(q))/o(F_4(q))+3)}{\log \log |F_4(q)|}$  on  $\epsilon_q(F_4(q))$ .

**3.13 Functions for groups of type  ${}^2F_4$** **3.13.1 Order2F4**

- Syntax: `Order2F4(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the group order  $|{}^2F_4(Q)|$ .

**3.13.2 LogLogOrder2F4**

- Syntax: `LogLogOrder2F4(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the real number  $\log \log |{}^2F_4(Q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(Order2F4(Q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.



**3.13.3** OrderOut2F4

- Syntax: `OrderOut2F4(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the outer automorphism group order  $|\text{Out}({}^2F_4(Q))|$ .

**3.13.4** NrConjugacyClasses2F4

- Syntax: `NrConjugacyClasses2F4(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the conjugacy class number  $k({}^2F_4(Q))$ .
- Comments: See Lübeck's database [11].

**3.13.5** NrAutOrbitsBound2F4

- Syntax: `NrAutOrbitsBound2F4(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the lower bound  $\underline{\omega}({}^2F_4(Q)) := \lceil \frac{k({}^2F_4(Q))}{|\text{Out}({}^2F_4(Q))|} \rceil$  on  $\omega({}^2F_4(Q))$ .

**3.13.6** epsilonOmegaBound2F4

- Syntax: `epsilonOmegaBound2F4(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the lower bound  $\underline{\epsilon}_\omega({}^2F_4(Q)) := \frac{\log \log \omega({}^2F_4(Q))}{\log \log |{}^2F_4(Q)|}$  on  $\epsilon_\omega({}^2F_4(Q))$ .

**3.13.7** SemisimpleElementOrders2F4

- Syntax: `SemisimpleElementOrders2F4(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the set of semisimple (viz., odd) element orders in  ${}^2F_4(Q)$ .
- Comments: Based on [6, Lemma 3].

**3.13.8** NrSemisimpleElementOrders2F4

- Syntax: `NrSemisimpleElementOrders2F4(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the number  $o_{\text{ss}}({}^2F_4(Q))$  of semisimple (viz., odd) element orders in  ${}^2F_4(Q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrders2F4(Q))`.

**3.13.9** ElementOrders2F4

- Syntax: `ElementOrders2F4(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the set of element orders in  ${}^2F_4(Q)$ .
- Comments: Based on [6, Lemma 3].

**3.13.10** NrElementOrders2F4

- Syntax: `NrElementOrders2F4(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the number  $o({}^2F_4(Q))$  of element orders in  ${}^2F_4(Q)$ .
- Comments: Simply calls `Length(ElementOrdersF4(q))`.

**3.13.11** `epsilonQBound2F4`

- Syntax: `epsilonQBound2F4(Q)`,  $Q = 2^{2k+1}$  for some  $k \in \mathbb{N}^+$ .
- Output: the lower bound  $\underline{\epsilon}_q(2F_4(Q)) := \frac{\log \log (\omega(2F_4(Q)) / \alpha(2F_4(Q)) + 3)}{\log \log |2F_4(Q)|}$  on  $\epsilon_q(2F_4(Q))$ .

**3.14** Functions for groups of type  $E_6$ **3.14.1** `OrderE6`

- Syntax: `OrderE6(q)`,  $q$  a prime power.
- Output: the group order  $|E_6(q)|$ .

**3.14.2** `LogLogOrderE6`

- Syntax: `LogLogOrderE6(q)`,  $q$  a prime power.
- Output: the real number  $\log \log |E_6(q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(OrderE6(q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

**3.14.3** `OrderOutE6`

- Syntax: `OrderOutE6(q)`,  $q$  a prime power.
- Output: the outer automorphism group order  $|\text{Out}(E_6(q))|$ .

**3.14.4** `NrConjugacyClassesBoundE6`

- Syntax: `NrConjugacyClassesBoundE6(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{k}(E_6(q)) := \lceil \frac{k(\text{Inndiag}(E_6(q)))}{|\text{Outdiag}(E_6(q))|} \rceil$  on  $k(E_6(q))$ .
- Comments: For  $k(\text{Inndiag}(E_6(q)))$  see Lübeck's database [11].

**3.14.5** `NrAutOrbitsBoundE6`

- Syntax: `NrAutOrbitsBoundE6(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\omega}(E_6(q)) := \lceil \frac{k(E_6(q))}{|\text{Out}(E_6(q))|} \rceil$  on  $\omega(E_6(q))$ .

**3.14.6** `epsilonOmegaBoundE6`

- Syntax: `epsilonOmegaBoundE6(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\epsilon}_\omega(E_6(q)) := \frac{\log \log \underline{\omega}(E_6(q))}{\log \log |E_6(q)|}$  on  $\epsilon_\omega(E_6(q))$ .

**3.14.7** CoxeterNoE6

- Syntax: `CoxeterNoE6()`.
- Output: the Coxeter number  $h(E_6) = 12$ .

**3.14.8** SemisimpleElementOrdersE6

- Syntax: `SemisimpleElementOrdersE6(q)`,  $q = p^f$  a prime power.
- Output: the set of semisimple (viz., not divisible by  $p$ ) element orders in  $E_6(q)$ .
- Comments: Based on [3, Theorem 1].

**3.14.9** NrSemisimpleElementOrdersE6

- Syntax: `NrSemisimpleElementOrdersE6(q)`,  $q = p^f$  a prime power.
- Output: the number  $o_{ss}(E_6(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $E_6(q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrdersE6(q))`.

**3.14.10** ElementOrdersE6

- Syntax: `ElementOrdersE6(q)`,  $q$  a prime power.
- Output: the set of element orders in  $E_6(q)$ .
- Comments: Based on [3, Theorem 1].

**3.14.11** NrElementOrdersE6

- Syntax: `NrElementOrdersE6(q)`,  $q$  a prime power.
- Output: the number  $o(E_6(q))$  of element orders in  $E_6(q)$ .
- Comments: Simply calls `Length(ElementOrdersE6(q))`.

**3.14.12** epsilonQBoundE6

- Syntax: `epsilonQBoundE6(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\epsilon}_q(E_6(q)) := \frac{\log \log (\omega(E_6(q))/o(E_6(q))+3)}{\log \log |E_6(q)|}$  on  $\epsilon_q(E_6(q))$ .

**3.15** Functions for groups of type  ${}^2E_6$ **3.15.1** Order2E6

- Syntax: `Order2E6(Q)`,  $Q = q^2$  for a prime power  $q$ .
- Output: the group order  $|{}^2E_6(q^2)|$ .

### 3.15.2 LogLogOrder2E6

- Syntax: `LogLogOrder2E6(Q)`,  $Q = q^2$  for a prime power  $q$ .
- Output: the real number  $\log \log |{}^2E_6(Q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(Order2E6(Q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

### 3.15.3 OrderOut2E6

- Syntax: `OrderOut2E6(Q)`,  $Q = q^2$  for a prime power  $q$ .
- Output: the outer automorphism group order  $|\text{Out}({}^2E_6(Q))|$ .

### 3.15.4 NrConjugacyClassesBound2E6

- Syntax: `NrConjugacyClassesBound2E6(Q)`,  $Q = q^2$  for a prime power  $q$ .
- Output: the lower bound  $\underline{k}({}^2E_6(Q)) := \lceil \frac{k(\text{Inndiag}({}^2E_6(Q)))}{|\text{Outdiag}({}^2E_6(Q))|} \rceil$  on  $k({}^2E_6(Q))$ .
- Comments: For  $k(\text{Inndiag}({}^2E_6(Q)))$  see Lübeck's database [11].

### 3.15.5 NrAutOrbitsBound2E6

- Syntax: `NrAutOrbitsBound2E6(Q)`,  $Q = q^2$  for a prime power  $q$ .
- Output: the lower bound  $\underline{\omega}({}^2E_6(Q)) := \lceil \frac{k({}^2E_6(Q))}{|\text{Out}({}^2E_6(Q))|} \rceil$  on  $\omega({}^2E_6(Q))$ .

### 3.15.6 epsilonOmegaBound2E6

- Syntax: `epsilonOmegaBound2E6(Q)`,  $Q = q^2$  for a prime power  $q$ .
- Output: the lower bound  $\underline{\epsilon_\omega}({}^2E_6(Q)) := \frac{\log \log \omega({}^2E_6(Q))}{\log \log |{}^2E_6(Q)|}$  on  $\epsilon_\omega({}^2E_6(Q))$ .

### 3.15.7 SemisimpleElementOrders2E6

- Syntax: `SemisimpleElementOrders2E6(Q)`,  $Q = p^{2f}$  a prime power square.
- Output: the set of semisimple (viz., not divisible by  $p$ ) element orders in  ${}^2E_6(Q)$ .
- Comments: Based on [3, Theorem 1].

### 3.15.8 NrSemisimpleElementOrders2E6

- Syntax: `NrSemisimpleElementOrders2E6(Q)`,  $Q = p^{2f}$  a prime power square.
- Output: the number  $\text{o}_{\text{ss}}({}^2E_6(Q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  ${}^2E_6(Q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrders2E6(Q))`.

**3.15.9** ElementOrders2E6

- Syntax: `ElementOrders2E6(Q)`,  $Q = q^2$  for a prime power  $q$ .
- Output: the set of element orders in  ${}^2E_6(Q)$ .
- Comments: Based on [3, Theorem 1].

**3.15.10** NrElementOrders2E6

- Syntax: `NrElementOrders2E6(Q)`,  $Q = q^2$  for a prime power  $q$ .
- Output: the number  $o({}^2E_6(Q))$  of element orders in  ${}^2E_6(Q)$ .
- Comments: Simply calls `Length(ElementOrders2E6(Q))`.

**3.15.11** epsilonQBound2E6

- Syntax: `epsilonQBound2E6(Q)`,  $Q = q^2$  for a prime power  $q$ .
- Output: the lower bound  $\underline{\epsilon}_q({}^2E_6(Q)) := \frac{\log \log (\omega({}^2E_6(Q))/o({}^2E_6(Q))+3)}{\log \log |{}^2E_6(Q)|}$  on  $\epsilon_q({}^2E_6(Q))$ .

**3.16** Functions for groups of type  $E_7$ **3.16.1** OrderE7

- Syntax: `OrderE7(q)`,  $q$  a prime power.
- Output: the group order  $|E_7(q)|$ .

**3.16.2** LogLogOrderE7

- Syntax: `LogLogOrderE7(q)`,  $q$  a prime power.
- Output: the real number  $\log \log |E_7(q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(OrderE7(q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

**3.16.3** OrderOutE7

- Syntax: `OrderOutE7(q)`,  $q$  a prime power.
- Output: the outer automorphism group order  $|\text{Out}(E_7(q))|$ .

**3.16.4** NrConjugacyClassesBoundE7

- Syntax: `NrConjugacyClassesBoundE7(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{k}(E_7(q)) := \lceil \frac{k(\text{Inndiag}(E_7(q)))}{|\text{Outdiag}(E_7(q))|} \rceil$  on  $k(E_7(q))$ .
- Comments: For  $k(\text{Inndiag}(E_7(q)))$  see Lübeck's database [11].

**3.16.5** `NrAutOrbitsBoundE7`

- Syntax: `NrAutOrbitsBoundE7(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\omega}(E_7(q)) := \lceil \frac{k(E_7(q))}{|\text{Out}(E_7(q))|} \rceil$  on  $\omega(E_7(q))$ .

**3.16.6** `epsilonOmegaBoundE7`

- Syntax: `epsilonOmegaBoundE7(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\epsilon}_\omega(E_7(q)) := \frac{\log \log \omega(E_7(q))}{\log \log |E_7(q)|}$  on  $\epsilon_\omega(E_7(q))$ .

**3.16.7** `CoxeterNoE7`

- Syntax: `CoxeterNoE7()`.
- Output: the Coxeter number  $h(E_7) = 18$ .

**3.16.8** `SemisimpleElementOrdersE7`

- Syntax: `SemisimpleElementOrdersE7(q)`,  $q = p^f$  a prime power.
- Output: the set of semisimple (viz., not divisible by  $p$ ) element orders in  $E_7(q)$ .
- Comments: Based on [4, Theorem 2].

**3.16.9** `NrSemisimpleElementOrdersE7`

- Syntax: `NrSemisimpleElementOrdersE7(q)`,  $q = p^f$  a prime power.
- Output: the number  $o_{\text{ss}}(E_7(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $E_7(q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrdersE7(q))`.

**3.16.10** `ElementOrdersE7`

- Syntax: `ElementOrdersE7(q)`,  $q$  a prime power.
- Output: the set of element orders in  $E_7(q)$ .
- Comments: Based on [4, Theorem 2].

**3.16.11** `NrElementOrdersE7`

- Syntax: `NrElementOrdersE7(q)`,  $q$  a prime power.
- Output: the number  $o(E_7(q))$  of element orders in  $E_7(q)$ .
- Comments: Simply calls `Length(ElementOrdersE7(q))`.

**3.16.12** `epsilonQBoundE7`

- Syntax: `epsilonQBoundE7(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\epsilon}_q(E_7(q)) := \frac{\log \log (\omega(E_7(q))/o(E_7(q))+3)}{\log \log |E_7(q)|}$  on  $\epsilon_q(E_7(q))$ .

## 3.17 Functions for groups of type $E_8$

### 3.17.1 OrderE8

- Syntax: `OrderE8(q)`,  $q$  a prime power.
- Output: the group order  $|E_8(q)|$ .

### 3.17.2 LogLogOrderE8

- Syntax: `LogLogOrderE8(q)`,  $q$  a prime power.
- Output: the real number  $\log \log |E_8(q)|$  as a decimal floating-point number.
- Comments: This does *not* just call `Log(Log(Float(OrderE8(q)))`, because applying `Float` to too large integers does not result in an appropriate floating-point representation of those integers, but in the output `inf`, which is treated like infinity.

### 3.17.3 OrderOutE8

- Syntax: `OrderOutE8(q)`,  $q$  a prime power.
- Output: the outer automorphism group order  $|\text{Out}(E_8(q))|$ .

### 3.17.4 NrConjugacyClassesE8

- Syntax: `NrConjugacyClassesE8(q)`,  $q$  a prime power.
- Output: the conjugacy class number  $k(E_8(q))$ .
- Comments: See Lübeck's database [11].

### 3.17.5 NrAutOrbitsBoundE8

- Syntax: `NrAutOrbitsBoundE8(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\omega}(E_8(q)) := \lceil \frac{k(E_8(q))}{|\text{Out}(E_8(q))|} \rceil$  on  $\omega(E_8(q))$ .

### 3.17.6 epsilonOmegaBoundE8

- Syntax: `epsilonOmegaBoundE8(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\epsilon}_\omega(E_8(q)) := \frac{\log \log \underline{\omega}(E_8(q))}{\log \log |E_8(q)|}$  on  $\epsilon_\omega(E_8(q))$ .

### 3.17.7 CoxeterNoE8

- Syntax: `CoxeterNoE8()`.
- Output: the Coxeter number  $h(E_8) = 30$ .

**3.17.8 SemisimpleElementOrdersE8**

- Syntax: `SemisimpleElementOrdersE8(q)`,  $q = p^f$  a prime power.
- Output: the set of semisimple (viz., not divisible by  $p$ ) element orders in  $E_8(q)$ .
- Comments: Based on [7].

**3.17.9 NrSemisimpleElementOrdersE8**

- Syntax: `NrSemisimpleElementOrdersE8(q)`,  $q = p^f$  a prime power.
- Output: the number  $o_{ss}(E_8(q))$  of semisimple (viz., not divisible by  $p$ ) element orders in  $E_8(q)$ .
- Comments: Simply calls `Length(SemisimpleElementOrdersE8(q))`.

**3.17.10 NrElementOrdersE8Bound**

- Syntax: `NrElementOrdersE8Bound(q)`,  $q = p^f$  a prime power.
- Output: the upper bound  $\bar{o}(E_8(q)) := o_{ss}(E_8(q)) \cdot (1 + \lceil \log_p(30) \rceil)$  on  $o(E_8(q))$ , the number of element orders in  $E_8(q)$ .

**3.17.11 epsilonQBoundE8**

- Syntax: `epsilonQBoundE8(q)`,  $q$  a prime power.
- Output: the lower bound  $\underline{\epsilon}_q(E_8(q)) := \frac{\log \log (\underline{\omega}(E_8(q)) / \bar{o}(E_8(q)) + 3)}{\log \log |E_8(q)|}$  on  $\epsilon_q(E_8(q))$ .

**3.18 High-end functions for finite simple groups of Lie type****3.18.1 NrAutOrbits**

- Syntax: `NrAutOrbits(G)`,  $G$  a finite group.
- Output: the number  $\omega(G)$  of  $\text{Aut}(G)$ -orbits on  $G$ .
- Comments: Proceeds by computing the conjugacy classes of  $G$  as well as a transversal for  $\text{Inn}(G)$  in  $\text{Aut}(G)$ , and then determining the number of orbits of the action of  $\text{Out}(G)$  on the conjugacy classes of  $G$ .

**3.18.2 ElementOrders**

- Syntax: `ElementOrders(G)`,  $G$  a finite group.
- Output: the set  $\text{Ord}(G)$  of element orders in  $G$ .
- Comments: Proceeds by computing first the conjugacy classes of  $G$ , and then the order in  $G$  of a representative of each conjugacy class.



### 3.18.3 NrElementOrders

- Syntax: `NrElementOrders(G)`,  $G$  a finite group.
- Output: the number  $o(G)$  of element orders in  $G$ .
- Comments: Simply calls `Length(ElementOrders(G))`.

### 3.18.4 epsilonOmegaBoundGeneral1

- Syntax: `epsilonOmegaBoundGeneral1(s,Q)`,  $s$  one of the strings "A1", "A2", "2A2", "B2", "C2", "G2", "2B2" or "2G2",  $Q$  a prime power such that if  ${}^tX_d$  is the Lie symbol encoded by  $s$ , then the notation  ${}^tX_d(Q)$  is defined.
- Output: the lower bound on  $\epsilon_\omega({}^tX_d(Q))$  discussed in [1, Case (1) of the proof of Theorem 1.1.3(2) in Subsection 3.3], based on the data displayed in [1, Table 3].

### 3.18.5 epsilonOmegaBoundGeneral2

- Syntax: `epsilonOmegaBoundGeneral2(d)`,  $d \in \mathbb{N}^+$ ,  $d \geq 3$ .
- Output: The GAP evaluation of the expression

$$\frac{\log\left(d - \frac{2\log(d+1)}{\log 2} - \frac{\log 6}{\log 2}\right) + \log \log 2}{\log(4d^2) + \log \log 2},$$

which is a uniform lower bound on  $\epsilon_\omega(S)$  where  $S$  is a finite simple Lie type group of the form  ${}^tX_d(2^t)$ , see [1, Case (2a) of the proof of Theorem 1.1.3(2) in Subsection 3.3].

- Comments: If  $d < 10$ , the argument in the first logarithm in the numerator is negative, whence the output then is `nan`. For  $d \geq 10$ , the output is a decimal floating-point number.

### 3.18.6 epsilonOmegaBoundGeneral3

- Syntax: `epsilonOmegaBoundGeneral3(d,q)`,  $d \in \mathbb{N}^+$ ,  $d \geq 3$ ,  $q > 2$  either a prime power or a positive algebraic number of the form  $\sqrt{2^{2k+1}}$  for some  $k \in \mathbb{N}^+$ .
- Output: the GAP evaluation of the expression

$$\frac{\log\left(d - \frac{2\log(d+1)}{\log q} - \frac{\log 6}{\log q} - \frac{1}{e \log 2}\right)}{\log(4d^2)},$$

which is a uniform lower bound on  $\epsilon_\omega(S)$  where  $S$  is a finite simple group of Lie type of the form  ${}^tX_d(q^t)$ , see [1, Case (2b) of the proof of Theorem 1.1.3(2) in Subsection 3.3].

### 3.18.7 q0ListOmega

This is a variable, defined as a certain list of length 16. For each  $d \in \{3, \dots, 18\}$ , the list contains a pair of the form  $[d, q_0]$  such that  $q_0$  is a prime power and such that for all finite simple Lie type groups  $S = {}^tX_d(q^t)$  with  $q \geq q_0$ ,  $\epsilon_\omega(S) > \epsilon_\omega(\text{Alt}(5))$ . See also [1, Table 4].

### 3.18.8 exceptionsListOmega

- Syntax: `exceptionsListOmega()`.
- Output: a finite list, consisting of triples of the form  $[s_1, d, Q]$  and pairs of the form  $[s_2, Q]$  where
  - $s_1$  is one of the strings "A", "2A", "B", "C", "D" or "2D",
  - $s_2$  is one of the strings "3D4", "F4", "2F4", "E6", "2E6", "E7" or "E8",
  - $d \in \mathbb{N}^+$ ,  $d \geq 3$ ,
  - $Q$  is a prime power,
  - if  ${}^tX$  resp.  ${}^tX_d$  is the Lie symbol naturally encoded by  $s_1$  resp.  $s_2$ , then the notation  ${}^tX_d(Q)$  is defined.

The corresponding list of finite simple groups of Lie type encoded by the output list has the property that if any nonabelian finite simple group  $S$  with  $S \not\cong \text{Alt}(5)$  and  $\epsilon_\omega(S) \leq \epsilon_\omega(\text{Alt}(5))$  exists, then  $S$  is in that list. See also [1, end of the proof of Theorem 1.1.3(2) in Subsection 3.3].

### 3.18.9 nrDistinctPartsPartitions

This is a variable, defined as a certain list of length 2012. For each  $i \in \{1, \dots, 2012\}$ , the  $i$ -th entry of the list is  $s(i-1)$ , the number of ordered integer partitions of  $i-1$  with pairwise distinct parts. This list of values was copied from the one linked in [12].

### 3.18.10 g2

- Syntax: `g2(d)`,  $d \in \mathbb{N}^+$ .
- Output: the function value  $g_2(d) := \sum_{d_++d_-=d} \sum_{i_+=0}^{d_+} \sum_{i_-=0}^{d_-} s(i_+)s(i_-)$  where  $s(n)$  denotes the number of ordered integer partitions of  $n$  with pairwise distinct parts.
- Comments: This function is defined and used in [1, Case (2) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

### 3.18.11 epsilonQBoundExceptional

- Syntax: `epsilonQBoundExceptional(s, Q)`, where
  - $s$  is one of the strings "2B2", "G2", "2G2", "3D4", "F4", "2F4", "E6", "2E6", "E7" or "E8",

- $Q$  is a prime power,
- if  ${}^tX_d$  is the Lie symbol naturally encoded by  $s$ , then the notation  ${}^tX_d(Q)$  is defined.
- Output: the GAP evaluation of a certain expression that is a lower bound on  $\epsilon_q({}^tX_d(Q))$ , defined in [1, beginning of the proof of Theorem 1.1.3(5) in Subsection 3.3] and based on the data displayed in [1, Table 5].
- Comments: As for `epsilonOmegaBoundGeneral2` (Subsubsection 3.18.5), the output may be `nan` for a few (finitely many) pairs  $(s, Q)$ .

### 3.18.12 `q0ListQExceptional`

This is a variable, defined as a certain list of length 10. For each string  $s$  of one of the forms "2B2", "G2", "2G2", "3D4", "F4", "2F4", "E6", "2E6", "E7" or "E8", the list contains an ordered pair of the form  $[s, q_0]$  where  $q_0$  is a prime power such that if  ${}^tX_d$  is the (exceptional) Lie symbol naturally encoded by  $s$  and  $Q$  is a prime power with  $Q \geq q_0$  such that the notation  ${}^tX_d(Q)$  is defined, then  $\epsilon_q({}^tX_d(Q)) > \epsilon_q(M)$ . See also [1, Table 5].

### 3.18.13 `exceptionsListQExceptional()`

- Syntax: `exceptionsListQExceptional()`
- Output: a finite list of pairs  $[s, Q]$  where  $s$  is a string of one of the forms "2B2", "G2", "2G2", "3D4", "F4", "2F4", "E6", "2E6", "E7" or "E8", and  $Q$  is a prime power such that if  ${}^tX_d$  is the (exceptional) Lie symbol naturally encoded by  $s$ , then the notation  ${}^tX_d(Q)$  is defined. Moreover, the (finite) list of exceptional finite simple groups of Lie type encoded by the output list has the property that if any exceptional finite simple group of Lie type  $S$  with  $\epsilon_q(S) \leq \epsilon_q(M)$  exists, then  $S$  is in that list.

### 3.18.14 `epsilonQBoundClassical1`

- Syntax: `epsilonQBoundClassical1(d,t)`,  $d, t \in \mathbb{N}^+$  ( $t$  is interpreted as a "type" specification).
- Output: If  $t \notin \{1, 2, 3, 4\}$ , then it returns the message "This type is not available. Please set the type to 1, 2, 3 or 4." as a string.

If  $t = 1$ , it returns the GAP evaluation of the expression

$$\frac{\log\left(\left(1 - \frac{\log 3}{\log 4}\right)d - \frac{\frac{2\pi}{\sqrt{3}}\sqrt{d} + 3\log(d+1) + \log\left(2 + \frac{\log(2d)}{\log 2}\right) + \log 4}{\log 2}\right) + \log \log 2}{\log(4d^2) + \log \log 2},$$

which is a uniform lower bound on  $\epsilon_q({}^tX_d(2^t))$  for  ${}^tX \in \{A, {}^2A, B, C, D, {}^2D\}$ ; see also [1, Case (1) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

If  $t = 2$ , it returns the GAP evaluation of the expression

$$\frac{\log\left(\left(1 - \frac{\log 4}{\log 9}\right)d - \frac{\frac{2\pi}{\sqrt{3}}\sqrt{d} + 3\log(d+1) + \log\left(2 + \frac{\log(2d)}{\log 2}\right) + \log 4}{\log 3} - \frac{1}{e \log 2}\right)}{\log(4d^2)},$$

which is a uniform lower bound on  $\epsilon_q({}^tX_d(q^t))$  for  ${}^tX \in \{A, {}^2A, B, C, D, {}^2D\}$  and  $q > 2$ ; see also [1, Case (1) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

If  $t = 3$ , it returns the GAP evaluation of the expression

$$\frac{\log\left(\left(1 - 0.311 \cdot \frac{\log 3}{\log 2}\right)d - \frac{\log g_2(d) + 2 \log 3 + \log\left(2 + \frac{\log(2d)}{\log 2}\right) + \log 2}{\log 2}\right) + \log \log 2}{\log(4d^2) + \log \log 2},$$

which is a uniform lower bound on  $\epsilon_q({}^tX_d(2^t))$  for  ${}^tX \in \{A, {}^2A, B, C, D, {}^2D\}$  if  $d \geq 91$ ; see also [1, Case (2) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

If  $t = 4$ , it returns the GAP evaluation of the expression

$$\epsilon_q(S) \geq \frac{\log\left(\left(1 - \frac{\log 4}{\log 27}\right)d - \frac{\log g_2(d) + 2 \log(d+1) + \log\left(2 + \frac{\log(2d)}{\log 2}\right) + \log 2}{\log 3} - \frac{1}{e \log 2}\right)}{\log(4d^2)},$$

which is a uniform lower bound on  $\epsilon_q({}^tX_d(q^t))$  for  ${}^tX \in \{A, {}^2A, B, C, D, {}^2D\}$  if  $q > 2$  and  $d \geq 55$ ; see also [1, Case (2) of the proof of Theorem 1.1.3(5) in Subsection 3.3].

### 3.18.15 epsilonQBoundClassical2

- Syntax: `epsilonQBoundClassical2(d,q)`,  $d \in \mathbb{N}^+$ ,  $q$  a prime power.
- Output: If  $d = 1$ , it returns the GAP evaluation of the expression

$$\frac{\log \log \left( \frac{q+1}{8 \log_2(q) \cdot (\sqrt{(q+1)/2} + \sqrt{(q-1)/2})} \right)}{\log \log (q(q^2 - 1))},$$

which is a lower bound on  $\epsilon_q(A_1(q))$ ; see also [1, Case (4) of the proof of Theorem 1.1.3(5) in Subsection 3.3]. If  $d > 1$ , it returns the GAP evaluation of the expression

$$\frac{\log \log \frac{q^d}{2c(d) \cdot \log_2(q) \cdot \min(d+1, q+1)^2 \cdot g_2(d) \cdot (1 + \lceil \log_2(2d) \rceil) \cdot (q+1)^{d/2}}}{\log \log q^{4d^2}}$$

where

$$c(d) = \begin{cases} 2, & \text{if } d \neq 4, \\ 6, & \text{if } d = 4. \end{cases}$$

Unless the output is `nan`, it is a uniform lower bound on  $\epsilon_q({}^tX_d(q^t))$  for  ${}^tX \in \{A, {}^2A, B, C, D, {}^2D\}$ ; see also [1, Case (4) of the proof of Theorem 1.1.3(5)].

### 3.18.16 q0ListQClassical

This is a variable, defined as a certain list of length 53. For each  $d \in \{1, \dots, 53\}$ , it contains a pair of the form `[d, q0]` where  $q_0$  is a prime power such that for all prime powers  $q \geq q_0$  and all (classical) Lie symbols  ${}^tX \in \{A, {}^2A, B, C, D, {}^2D\}$ ,  $\epsilon_q({}^tX_d(q^t)) > \epsilon_q(M)$ . See also [1, Table 8].

**3.18.17** exceptionsListQClassical

- Syntax: `exceptionsListQClassical()`.
- Output: a finite list of triples `[s,d,Q]` where
  - $s$  is one of the strings "A", "2A", "B", "C", "D" or "2D",
  - $d \in \mathbb{N}^+$  and
  - $Q$  is a prime power such that if  ${}^tX$  is the (classical) Lie symbol naturally encoded by  $s$ , then the notation  ${}^tX_d(Q)$  is defined.

Such a triple `[s,d,Q]` is then to be viewed as a code for the classical finite simple Lie type group  ${}^tX_d(Q)$ , and the output list encodes a list of such groups. This latter list has the property that if any nonabelian finite simple group  $S \not\cong M$  with  $\epsilon_q(S) \leq \epsilon_q(M)$  exists, then  $S$  occurs in the list. See also [1, Case (5) of the proof of Theorem 1.1.3(5)].

**References**

- [1] A. Bors, M. Giudici and C.E. Praeger, Automorphism orbits and element orders in finite groups: almost-solubility and the Monster, preprint (2019), <https://arxiv.org/abs/1910.11781>.
- [2] R. Brandl and W.J. Shi, A characterization of finite simple groups with abelian Sylow 2-subgroups, *Ric. Mat.* **42**(1):193–198, 1993.
- [3] A.A. Buturlakin, Spectra of finite simple groups  $E_6(q)$  and  ${}^2E_6(q)$ , *Algebra Logic* **52**(3):188–202, 2013.
- [4] A.A. Buturlakin, Spectra of finite simple groups  $E_7(q)$ , *Sib. Math. J.* **57**(5):769–777, 2016.
- [5] A.A. Buturlakin and M.A. Grechkoseeva, The cyclic structure of maximal tori of the finite classical groups, *Algebra Logic* **46**(2):73–89, 2007.
- [6] H.W. Deng and W.J. Shi, The characterization of Ree groups  ${}^2F_4(q)$  by their element orders, *J. Alg.* **217**(1):180–187, 1999.
- [7] D.I. Deriziotis and A.P. Fakiolas, The maximal tori in the finite Chevalley groups of type  $E_6$ ,  $E_7$  and  $E_8$ , *Comm. Algebra* **19**(3):889–903, 1991.
- [8] J. Fulman and R. Guralnick, Bounds on the number and sizes of conjugacy classes in finite Chevalley groups with applications to derangements, *Trans. Amer. Math. Soc.* **364**(6):3023–3070, 2012.
- [9] M.A. Grechkoseeva and M.A. Zvezdina, On spectra of automorphic extensions of finite simple groups  $F_4(q)$  and  ${}^3D_4(q)$ , *J. Algebra Appl.* **15**(9):165–168, 2016.
- [10] B. Hartley, A general Brauer-Fowler theorem and centralizers in locally finite groups, *Pacific J. Math* **152**(1):101–117, 1992.

- [11] F. Lübeck, Data for Finite Groups of Lie Type and Related Algebraic Groups, online database, <http://www.math.rwth-aachen.de/~Frank.Luebeck/chev/index.html>.
- [12] The On-Line Encyclopedia of Integer Sequences, A000009, <https://oeis.org/A000009>.
- [13] W.J. Shi, A characterization of Suzuki's simple groups, *Proc. Amer. Math. Soc.* **114**(3):589–591, 1992.
- [14] A.V. Vasil'ev and A.M. Staroletov, Recognizability of groups  $G_2(q)$  by spectrum, *Algebra Logic* **52**(1):1–14, 2013.
- [15] G.E. Wall, On the conjugacy classes in the unitary, symplectic and orthogonal groups, *J. Aust. Math. Soc.* **3**:1–62, 1963.