

The transitive permutation groups of degree 32

John J. Cannon, Derek F. Holt

Abstract

We describe our successful computation of a list of representatives of the 2 801 324 conjugacy classes of transitive groups of degree 32.

1 Introduction

Many attempts to classify transitive groups of small degree (up to 15) were carried out in the late 19th and early 20th centuries. See [Short 1992] for details and references. More recently, with assistance from computers, transitive groups up to degree 11 were listed in [Butler & McKay 1983], degree 12 in [Royle 1987], degrees 14 and 15 in [Butler 1993], and all degrees up to 30 in [Hulpke 1996]; see also [Hulpke 2005]. These are available as libraries in GAP [GAP 2004] and MAGMA [Bosma et al 1997] up to degree 30.

Transitive groups of degree 31 are primitive and hence already known. (There are 12 classes, including $L_3(5)$, $L_5(2)$, $\text{Alt}(31)$, $\text{Sym}(31)$ and 8 soluble examples of order dividing 31×30 ; see, for example, [Pogorolov1980].) Here, we report on our recent computation of a complete list of representatives of the conjugacy classes in $\text{Sym}(32)$ of the transitive groups of degree 32. There are 2 801 324 such classes. The magnitude of this number in comparison with smaller degrees (the largest such is 25 000 classes of transitive groups of degree 24) had already been anticipated by Hulpke, who perceived this a disincentive to extending his lists to degree 32.

We have not yet seriously considered the problem of calculating lists of transitive permutation groups of higher degree. We would hazard a guess that for degrees between 33 and 64, 36 and 48 would be very difficult but potentially possible, degree 64 would be impossible with current techniques and resources, and the other degrees would be comparatively straightforward.

Tables of transitive groups are used in numerous applications. One of the more important has been in the study of Galois groups of polynomials. In [Stauduhar 1973], a method for computing the Galois group over the rational field of an irreducible

monic polynomial with integral coefficients is described. Geissler's implementation in KANT and Magma, described in [Geissler & Klüners 2000], stores the transitive lattice for each symmetric group having degree up to 23 and so is capable of finding the Galois group of any irreducible polynomial up to that degree.

A knowledge of the transitive groups of degree n is important in the study of the *inverse problem* of Galois theory: given a transitive group G of degree n , does G occur as the Galois group of a Galois field extension of Q ? Klüners and Malle [Klüners & Malle 2000] have determined such a polynomial for every transitive group of degree less than 17. The discovery of a finite group not realisable as a Galois group would be of major interest and provides motivation for the extension of this work to higher degrees.

Let G be the Galois group of a field extension K of Q and suppose H_1 and H_2 are non-conjugate subgroups of G . If H_1 and H_2 afford the same permutation character of G , then the corresponding invariant subfields K^{H_1} and K^{H_2} of K are non-isomorphic but have the same zeta function. The two fields are then said to be *arithmetically equivalent*. In [Bosma & de Smit 2001, Bosma & de Smit 2002], the existing tables of transitive groups are used to determine the non-isomorphic arithmetically equivalent fields of degree $n \leq 22$, and de Smit has now extended the list up to degree 31. Work is underway using our classification to do the same for degree 32.

As a final application, the enumeration of all vertex-transitive graphs on at most 31 vertices using the classification of transitive groups having degree less than 32 is described in [Royle & Praeger 1989, McKay & Royle 1990].

We discuss the methods used in Section 2. We used two different methods, which we shall call the *brute force method* and the *Hulpe method*, depending on the smallest size of a block in a block system preserved by the group. We have also written a function in MAGMA that takes a transitive group G of degree 32 as input, identifies its unique conjugate G' in our lists, and finds an element of $\text{Sym}(32)$ that conjugates G to G' . We shall describe this function briefly in Section 3.

Since many of our earlier runs had to be aborted to allow us to improve some of our techniques, it is difficult to estimate the total time taken for the computations, but the final successful runs took a total of about two weeks of cpu-time, and the maximum RAM usage was just under 7 Gigabytes. We could to a large extent avoid excessive memory usage in storing the groups by subdividing the enumeration into separate cases, but there were some cohomology computations, which we shall discuss in Section 2, which required moderately large amounts of memory (up to 7GB). The computations were run on an AMD Opteron Model 152 processor running at 2.6 GH with 4 GB of memory and an AMD Opteron Model 280 processor running at 2.4 GH with 16 GB of memory.

Potentially, and also to a large extent in reality, the most time-consuming component of the computations is testing groups for conjugacy in $\text{Sym}(32)$. Computing normalisers and testing conjugacy of subgroups are notoriously difficult problems even in permutation groups of small degree. The best available algorithms are based on partition backtrack methods, as described in [Leon 1997], but the currently available implementations are not particularly good and are in need of improvement. The methods introduced in [Hulpke 2005] attempt to minimise the need for conjugacy testing of subgroups and, in cases where it cannot be avoided, it is usually possible to arrange for the conjugacy tests to take place in a relatively small subgroup of $\text{Sym}(32)$. With the brute force method, more conjugacy testing is carried out, but again we try to do it in as small a subgroup as possible. A large amount of the effort involved in getting these computations to complete was devoted to finding tricks for speeding up conjugacy testing in individual cases. We shall briefly describe some of these methods in Section 2.

Although there are 487 distinct orders of groups in the final list, 2 737 535 of the 2 801 324 groups - that is, about 97.7% of them - are 2-groups of order 2^n for $5 \leq n \leq 31$, the most popular order being 2^{15} , of which there are 391 809 groups. Only 1051 of the groups are insoluble.

All except 8295 of the groups - that is, about 99.7% of them - are imprimitive groups with a block system with blocks of size 2.

We observe also that 11605 of the groups in the list are minimally transitive; that is, all of their proper subgroups are intransitive.

The computations were carried out in MAGMA [Bosma et al 1997] during the second author's visit to the University of Sydney early in 2007, and he is grateful to the Australian Research Council who provided financial support for this visit. The list of transitive groups of degree 32 will be accessible in MAGMA in a future release (probably early in 2008), and is available immediately from the authors on request.

2 Methods used in the calculations

There are far fewer primitive than transitive permutation groups, and complete lists of representatives of conjugacy classes of primitive groups have been computed up to degree 2500; see [Roney-Dougal & Unger 2003, Roney-Dougal 2007] (or [Short 1992] for historical details). In particular, there are just 7 such groups of degree 32.

So we can concentrate on the imprimitive examples. As in [Hulpke 2005], we subdivide the problem according to the minimal blocksize in block systems stabilised by the group, which for degree 32 can be 2,4,8, or 16. Let d be the minimal blocksize and let $t = 32/d$ be the number of blocks in the system.

We shall assume that the transitive groups G that we are enumerating act on the set of integers in the range 1 to 32 and, since we are working up to conjugacy in $\text{Sym}(32)$, we can assume that those groups G with minimal blocksize d stabilise the block system

$$\mathcal{B} := \{ \{dn + k : 1 \leq k \leq d\} : 0 \leq n \leq t - 1 \},$$

which we shall refer to as the *principal minimal block system* fixed by G . In general, G may also fix other minimal block systems.

Of the two methods to be described, the brute force and Hulpke methods, the former is very much easier and more straightforward to program, so we used it whenever practical, which was the case for $d = 16, 8$ and 4 . The brute force method was impractical for $d = 2$, but we note that a variant of this method was used by Hulpke in [Hulpke 2005] for the purpose of performing random tests to verify the correctness of his lists.

2.1 The brute force method

For a fixed minimal blocksize d , we can try a straightforward “brute force” method of listing the groups in this class as follows. All such groups are subgroups of the unique largest group in this class, which is $W_d := \text{Sym}(d) \wr \text{Sym}(t)$. We initialise our list of groups to $[W_d]$. Then, for each group G in the list, we compute its maximal subgroups, using the algorithm described in [Cannon & Holt 2004]. For each such subgroup H , we test whether H is transitive with minimal blocksize d and, if so, we test H for conjugacy in $\text{Sym}(32)$ with the groups already in the list. If H is not conjugate to any of these, then we append it to the list.

For $d = 16$ and $d = 8$, this process completed very quickly. Up to conjugacy in $\text{Sym}(32)$, there are 171 groups with minimal blocksize 16 and 233 with minimal blocksize 8. We also successfully applied this method to the case $d = 4$, for which there are 7884 groups, although this run took about 5 days of cpu-time.

The most time-consuming part of this process is the conjugacy testing, but in fact we do not really need to test conjugacy in $\text{Sym}(32)$. As in the procedure for minimal blocksize 2 to be described below, we take each minimal block system \mathcal{B} stabilised by H in turn, choose $g \in \text{Sym}(32)$ that maps \mathcal{B} to the principal minimal block system, and then test $H' := H^g$ for conjugacy in W_d with the groups already in the list.

The groups with minimal blocksizes 8 and 16 all stabilise a unique minimal block system, and the same is true for all but 107 of the 7884 groups with minimal blocksize 4. Of those, the majority stabilise just two minimal block systems, and the maximum number stabilised is 5, so the conjugacy testing process effectively reduces to tests within W_d . But even these can be unpleasantly time consuming, and we made use

of a number of extra tricks, such as counting the numbers of fixed points of elements of various orders, to help identify pairs of groups that could not be conjugate more quickly than the default conjugacy testing function in MAGMA.

In addition, we can speed up conjugacy testing in W_d by first testing the kernels of the actions of the two groups on the block system for conjugacy in W_d , then testing their images on the blocks for conjugacy in $\text{Sym}(t)$, and finally (if necessary) testing the groups for conjugacy within the normaliser in W_d of the specific kernel and image. For the first of these steps, testing conjugacy of the kernels, we made use of the functions of J.S. Leon, for automorphism and isomorphism testing of designs, which will be described in more detail for the case $d = 2$ in Subsubsection 2.2.1 below. The second step, testing block images for conjugacy in $\text{Sym}(t)$, presented no difficulties using default functions. The third step was time consuming in a few isolated cases for which the kernel and image normalisers were large, but there was nothing further that we could do to alleviate this.

2.2 Groups with minimal blocksize two

For the case $d = 2$, the brute force method is impractical, and so we applied the Hulpke method, which is described in [Hulpke 2005]. Again, all of the groups in our list are subgroups of the largest group $W_2 = \text{Sym}(2) \wr \text{Sym}(16)$ that fixes the principal block system defined earlier. Let K be the base group of W_2 (so K is elementary abelian of order 2^{16}), and let $\phi : W_2 \rightarrow \text{Sym}(16)$ be the induced action of W_2 on the blocks in the system (so $\ker \phi = K$).

Following the notation used in [Hulpke 2005], for a group G on our list (or a candidate for appending to the list), let $M = G \cap K$ be the kernel of the action of G on the principal block system, and let $R = \phi(G)$ be the group induced by G on the block system. So R is a transitive group of degree $t = 16$. Here is an outline description of the method of enumerating the groups. For the remainder of this section, we shall denote W_2 simply by W .

A Find the possible kernels M of the groups G on the principal minimal block system, as defined above, and make these into an ordered list in which kernels of smaller order come before those of larger order.

B For each kernel M in this list do the following.

1. Compute the normaliser $N := N_W(M)$ and let $R := \phi(N)$. (Note that, since K is abelian, we have $K \leq N$.)
2. Compute representatives \bar{S} of the conjugacy classes of transitive subgroups of R , and make these into an ordered list. The brute force method can be used for this purpose when $R \neq \text{Sym}(16)$.

3. For each \bar{S} in this list, let $S := \phi^{-1}(\bar{S})$ and do the following.
 - (i) Compute representatives of the conjugacy classes of complements of K/M in S/M . Obtain representatives of the $N_N(S)$ -classes of these. Each such complement has the form G/M for some transitive subgroup G of W with $G \cap K = M$.
 - (ii) For each such complement G , do the following.
 - (a) Find all minimal block systems fixed by G .
 - (b) For each such block system \mathcal{B} other than the principal block system, do the following.
 - (b1) Choose g in $\text{Sym}(32)$ mapping \mathcal{B} to the principal block system, and let $G' = G^g$; so G' also fixes the principal block system.
 - (b2) Let $M' := G' \cap K$. Then M' is conjugate in W to a unique member of the list of possible kernels M . Replace G' and M' by a suitable conjugate in W so that M' becomes equal to one of the groups in the list of kernels.
 - (b3) If M' comes earlier in the list of kernels than M , then mark G as seen already, and proceed immediately to Step (c) below.
 - (b4) If M' comes later in the list of kernels than M , then proceed immediately to the next non-principal block system fixed by G (if any) at Step (b1).
 - (b5) Now we have $M' = M$. Let $\bar{S}' = \phi(G')$. Then \bar{S}' is conjugate in R to a unique member of the list of transitive subgroups of R . Replace G' by a suitable conjugate in N and \bar{S}' by the corresponding conjugate in R so that \bar{S}' becomes equal to one of the groups in the list of transitive subgroups of R .
 - (b6) If \bar{S}' comes earlier in the list of transitive subgroups of R than \bar{S} , then mark G as seen already, and proceed immediately to Step (c) below.
 - (b7) If \bar{S}' comes later in the list of transitive subgroups of R than \bar{S} , then proceed immediately to the next non-principal block system fixed by G (if any) at Step (b1).
 - (b8) Now we have $\bar{S}' = \bar{S}$. Test G' for conjugacy in $N_N(S)$ with any groups G'' that are already in the list, and for which $G'' \cap K = M$ and $\phi(G'') = \bar{S}$. If G' is conjugate to one of these groups G'' , then mark G as seen already.
 - (c) If G has not been marked as seen already, then append it to the list of transitive groups with minimal blocksize 2.

There are a number of differences between the above process and the corresponding procedure presented in Section 5 of [Hulpke 2005].

- Unlike in [Hulpke 2005], we do not treat the case $M = 1$ differently from any of the other cases.
- In Step **B1**, $N_{\text{Sym}(32)}(M)$ rather than $N_W(M)$ occurs in [Hulpke 2005], but these are equal provided that $M \neq 1$, because the blocks of the principal block system are the orbits of M and so the normaliser of M lies in W .
- We have described the treatment of multiple minimal block systems in more detail than in [Hulpke 2005].

It is not difficult to see that the procedure described above has the desired effect, and generates a unique representative of each conjugacy class of transitive subgroups of $\text{Sym}(32)$ that stabilises a block system with blocksize 2. For more details of why this process works, we refer the reader to [Hulpke 2005].

We shall now discuss some of the individual steps in the procedure in more detail.

2.2.1 Finding the kernels

In the classification of groups of degree up to 30, as described in [Hulpke 2005], finding the kernels (Step **A**) turned out to be the most time-consuming part of the whole process, but that was far from being the case for groups of degree 32 with minimal blocksize 2, and we completed this part of the computation using less than an hour of cpu-time. We observe, however, that this step would have been considerably more difficult for groups of degree 32 with minimal blocksize 4, 8 or 16, which is the principal reason why we preferred the brute force method for those cases.

For such a group G with $\phi(G) = R$, we can think of K as a (permutation) module of dimension 16 for R over the field \mathbb{F}_2 of order 2, and then M is a $\mathbb{F}_2 R$ -submodule of K . So to find all possible kernels M , we first need to find all $\mathbb{F}_2 R$ -submodules of K for all possible image groups R . But if M is a submodule under the action of R , then it is also a submodule under the action of any subgroup of R so, to find all such M , we need compute the submodules only for the minimal transitive groups of degree 16 (that is, those in which all proper subgroups are intransitive), and there are just 75 of those up to conjugacy in $\text{Sym}(16)$.

We then need to test the subgroups M found for conjugacy in $\text{Sym}(32)$ but, since the orbits of all of these subgroups are just the blocks of the principal block system, this is equivalent to testing for conjugacy in $W = \text{Sym}(2) \wr \text{Sym}(16)$. For this purpose, and also for finding the normalisers of the kernels in Step **B1** and for the conjugacy tests in Step **B3** (ii)(b2), we used the following technique to speed up the computations.

$ M $	$\#S$	$\#G$	$ M $	$\#S$	$\#G$	$ M $	$\#S$	$\#G$
2^0	1954	10761	2^7	57	476	2^9	449	7616
2^1	1954	31757	2^7	449	75016	2^9	66	213
2^2	2477	64275	2^7	66	329	2^{10}	2649	182005
2^3	2596	77244	2^8	2004	26526	2^{10}	176	944
2^4	2596	64726	2^8	683	20386	2^{10}	1790	43720
2^4	2804	50789	2^8	438	4632	2^{11}	2648	190319
2^5	2648	287868	2^8	2759	185769	2^{11}	599	8172
2^5	599	21262	2^8	2362	38572	2^{12}	2596	33402
2^6	2649	246045	2^8	2119	37160	2^{12}	2804	54413
2^6	176	1516	2^8	30	232	2^{13}	2596	55244
2^6	1790	288315	2^9	2004	76088	2^{14}	2477	24109
2^7	2004	122001	2^9	2362	127185	2^{15}	1954	11532
2^7	2362	319718	2^9	57	738	2^{16}	1954	1954

Table 1: Numbers of groups for each kernel

For each kernel M , we form a combinatorial design on 32 points of which the blocks of the design are the fixed points of the non-trivial elements of M . To test conjugacy of two kernels M , or to find the normaliser of one such kernel, we first test whether the corresponding designs are isomorphic or, respectively, find the automorphism group of the design. This process uses implementations by J.S. Leon [Leon 1991, Leon 1997] of partition backtrack methods, and is very much faster than conjugacy testing or normaliser computation of permutation groups. The normaliser of M in W can then be computed as the normaliser within the automorphism group of the design, which is in all cases significantly smaller than W . Similarly, to test conjugacy in the case when the designs are isomorphic, we first replace one of the groups M by a conjugate to make them both stabilise the same design, and then we can test them for conjugacy within the normaliser of that design.

Up to conjugacy in $\text{Sym}(32)$, there were 39 such kernels altogether, one each of orders 2^i for $i = 0, 1, 2, 3, 13, 14, 15, 16$, two of orders 2^i for $i = 4, 5, 11, 12$, three of orders 2^6 and 2^{10} , five of orders 2^7 and 2^9 , and seven of order 2^8 .

For the remainder of the computation, we carried out 39 separate runs, one for each kernel, and stored the resulting lists in 39 separate files. Table 1 shows, in the third column, the numbers of groups in the lists for each kernel, and also, in the second column, the number of conjugacy classes of transitive subgroups of $R = \phi(N_W(M))$ found in Step **B2**.

The brute force method described above in Subsection 2.1 was used to find the subgroups in Step **B2**. But, for the kernels of orders $2^0, 2^1, 2^{15}, 2^{16}$, we have $R = \text{Sym}(16)$ so we could use the existing list of transitive groups of degree 16.

2.2.2 Finding classes of complements

Step 3(i) of the procedure turned out to be the most expensive in terms of memory resources required and, together with conjugacy testing of subgroups, was also the most time-consuming step. Since we were finding complements of a normal elementary abelian subgroup, we could treat this subgroup as a module for the group, and use cohomological methods. These methods, including the calculation of class representatives under the action of $N_N(S)$, are described in detail in Section 5 of [Cannon et al 2001], so we shall not discuss them further here, except to remark that we were fortunate that the necessary code was already available to carry them out.

The magnitude of the required resources resulted from the fact that there were a few cases in which there were 2^{22} classes of complements of K/M in S/M , and we needed to store representatives of all of these in order to calculate orbit representatives under the action of $N_N(S)$. Typically, the number of orbits in these cases was a few thousand. The complements themselves were not stored as permutation groups, but as 1-cocycles, which could be represented by lists of vectors specifying the images of the group generators under the cocycles. If the number of classes of complements had been higher by a few powers of 2, then we would probably have been unable to complete the computations, so this type of calculation might turn out to be the factor that determines whether it is feasible to extend the classification of transitive groups to higher degrees.

The remaining steps in the algorithm were all relatively easy, except occasionally for the conjugacy test in Step (b8), but this was rarely required at all. It was only needed in cases where there was more than one minimal block system fixed by G and even then it was not always needed. In fact, of the 2 793 029 groups on the list with minimal blocksize 2, only 155 461 (that is, about 5.5%) stabilise more than one minimal block system.

3 Identifying a given group

If we are given an arbitrary transitive permutation group G of degree 32, then we would like to be able to identify the (unique) group G' on our lists to which G is conjugate, and to find $g \in \text{Sym}(32)$ with $G^g = G'$.

Doing this is a similar process to that displayed in the procedure above for finding representatives of the conjugacy classes of the groups. Although we used the brute force method to compute the lists of groups with minimal block systems of size $d = 4, 8$ and 16 , we retrospectively reorganised these lists according to the kernels of their actions on the principal block systems, and replaced groups by conjugates in

order to ensure that groups with conjugate kernels had equal kernels. This facilitates the identification process. For the case $d = 2$, we have also stored the lists of transitive subgroups of $R = \phi(N_{W_2}(M))$, which were computed in Step **B1** of the procedure above.

Primitive groups are easily handled. Here is an outline of the method for imprimitive groups.

- (a) Find all minimal block systems fixed by G .
- (b) For each such block system \mathcal{B} , do the following.
 - (b1) Choose g_1 in $\text{Sym}(32)$ mapping \mathcal{B} to the principal block system, and replace G by G^{g_1} ; so G now fixes the principal block system.
 - (b2) Let M be the kernel of the action of G on the principal block system. Then M is conjugate in $W_d := \text{Sym}(d) \wr \text{Sym}(t)$ to a unique kernel M' of a group on our list. Find $g_2 \in W_d$ with $M^{g_2} = M'$ and replace G by G^{g_2} .
 - (b3) If $d > 2$ then test G for conjugacy with all groups G' in the list with kernel M . If $G^{g_3} = G'$ for some such G' then return G' and conjugating element $g_1g_2g_3$. Otherwise proceed to the next block system at Step (b1).
 - (b4) Now $d = 2$. Let $\bar{S} = \phi(G)$. Then, setting $N = N_{W_2}(M)$ and $R = \phi(N)$, \bar{S} is conjugate in R to a unique member of the list of transitive subgroups of R . Replace G by a suitable conjugate G^{g_3} with $g_3 \in N$ to make \bar{S} equal to its conjugate in the list.
 - (b5) Test G for conjugacy in $N_N(S)$ with the groups G' in the list for which $G' \cap K = M$ and $\phi(G') = \bar{S}$. If $G^{g_4} = G'$ for some such G' then return G' and conjugating element $g_1g_2g_3g_4$. Otherwise proceed to the next block system at Step (b1).

We have tested this process extensively, partly because such testing helps to provide evidence for the correctness of the lists. We have used two types of tests.

For the first type, we have taken groups from the lists themselves, conjugated them by random elements of $\text{Sym}(32)$, and then applied the above identification process. This would of course fail to reveal any groups that were missing from the lists, but it could potentially identify any duplicates in the lists, because the presence of duplicates would result in some of the groups being found to be conjugate to different groups in the lists. We have carried out these tests on all of the groups with minimal blocksize larger than 2, and on random samples of several thousands of each of the 39 collections of groups with minimal blocksize 2.

For the second type of test, we form sequences of groups $G_0 = \text{Sym}(32), G_1, \dots, G_r$, where each G_i is a randomly chosen maximal transitive subgroup of G_{i-1} , and r is

a small integer. We then apply the identification test to G_r . This process could in principle detect any omissions from the lists, because if G_r was missing, then it would fail to be identified. We have carried out such tests many thousands of times.

Experiments indicate that we achieve the most even distribution of the different types and orders of groups on the lists by choosing r itself at random, and by using something like a Poisson distribution for r , with maximum likelihood at about $r = 5$. But unfortunately this method still does not succeed in choosing groups from our lists with anything like uniform distribution. The largest numbers of groups on the lists are those with minimal blocksize 2 and with kernel M having order 2^n for intermediate values of n , such as $5 \leq n \leq 11$. The procedure above, however, seems to result in a disproportionately large number of groups with minimal blocksize 2, but with kernels of order 2^n for large and small values of n , such as 1,2,15,16.

The performance of the identification process is good on average, but there are occasional bad cases, so there is scope for further improvements. Groups with minimal blocksize 8 or 16 are identified almost instantly. The average cpu-time for those with minimal blocksize 4 is about 7 seconds, but with a worst case encountered of 1456 seconds. The bad cases result from the occasional need to carry out time-consuming subgroup conjugacy tests.

For blocksize 2, the average identification time depends heavily on the number of groups in the particular list to which the group belongs. (Recall that there is a separate list for each kernel.) The average for the shorter lists is around 1 second, whereas for the longer lists, that contain more than 100000 groups, the average increases to around 60 seconds. There is however much less variation in the times than for blocksize 4, presumably because there are far fewer difficult subgroup conjugacy tests to be carried out.

References

- [Bosma et al 1997] Wieb Bosma, John Cannon, and Catherine Playoust. The MAGMA algebra system I: The user language. *J. Symbolic Comput.*, 24:235–265, 1997.
- [Bosma & de Smit 2001] Wieb Bosma and Bart de Smit. Class number relations from a computational point of view. *J. Symbolic Comput.*, 31:97–112, 2001.
- [Bosma & de Smit 2002] Wieb Bosma and Bart de Smit. On arithmetically equivalent number fields of small degree,. Algorithmic Number Theory (Sydney, 2002). Lecture Notes in Comput. Sci., 2369: 67–79, 2002. Springer, Berlin.
- [Butler & McKay 1983] G. Butler and J. McKay. The transitive groups of degree up to 11. *Comm. Algebra*, 11:863–911, 1983.

- [Butler 1993] G. Butler. The transitive groups of degree fourteen and fifteen. *J. Symbolic Comput.*, 16:413–422, 1993.
- [Cannon et al 2001] J.J. Cannon, B.C. Cox, and D.F. Holt. Computing the subgroups of a permutation group. *J. Symbolic Comput.*, 31:149–161, 2001.
- [Cannon & Holt 2004] J.J. Cannon and D.F. Holt. Computing maximal subgroups of finite groups. *J. Symbolic Comput.*, 37:589–609, 2004.
- [GAP 2004] The GAP Group, GAP – Groups, Algorithms, and Programming, Version 4.4; 2004 (<http://www.gap-system.org>).
- [Geissler & Klüners 2000] K. Geissler and J. Klüners. Galois group computation for rational polynomials. *J. Symbolic Comput.*, 30:653–674, 2000.
- [Hulpke 1996] A. Hulpke. *Konstruktion transitiver Permutationsgruppen*. PhD thesis, RWTH Aachen, 1996.
- [Hulpke 2005] A. Hulpke. Constructing transitive permutation groups. *J. Symbolic Comput.*, 39:1–30, 2005.
- [Klüners & Malle 2000] J. Klüners and G. Malle. Explicit Galois realization of transitive groups of degree up to 15. *J. Symbolic Comput.*, 30:675–716, 2000.
- [Leon 1991] J.S. Leon. Permutation group algorithms based on partitions, I: Theory and algorithms. *J. Symbolic Comput.*, 12:533–583, 1991.
- [Leon 1997] J.S. Leon. Partitions, refinements, and permutation group computation. In *Groups and Computation II*, Larry Finkelstein and William M. Kantor, editors. volume 28 of *Amer. Math. Soc. DIMACS Series*. (DIMACS, 1995), 123–158, 1997.
- [McKay & Royle 1990] Brendan D. McKay and Gordon F. Royle. The transitive graphs with at most 26 vertices. *Ars Combin.*, 30:161–176, 1990.
- [Pogorolov1980] Pogorolov, B.A. Primitive permutation groups of small degrees, I and II. *Algebra and Logic*, 19:230–254 and 278–296, 1980.
- [Roney-Dougal & Unger 2003] C.M. Roney-Dougal and W.R. Unger. The affine primitive permutation groups of degree less than 1000. *J. Symbolic Comput.*, 35:421–439, 2003.
- [Roney-Dougal 2007] C.M. Roney-Dougal. The primitive groups of degree less than 2500. *J. Algebra*, to appear.
- [Royle & Praeger 1989] Gordon F. Royle and Cheryl E. Praeger. Constructing the vertex-transitive graphs of order 24. *J. Symbolic Comput.*, 8:309–326, 1989.

- [Royle 1987] Gordon F. Royle. The transitive groups of degree twelve. *J. Symbolic Comput.*, 4:255–268, 1987.
- [Short 1992] M.W. Short. *The Primitive Soluble Permutation Groups of Degree Less than 256*, volume 1519 of *Lecture Notes in Math.* Springer-Verlag, Berlin, Heidelberg, New York, 1992.
- [Stauduhar 1973] R. P. Stauduhar. The determination of Galois groups. *Math. Comp.*, 27:981–996, 1973.

Addresses:

School of Mathematics and Statistics
University of Sydney
NSW 2006
Australia
e-mail: john@maths.usyd.edu.au

Mathematics Institute
University of Warwick
Coventry CV4 7AL
Great Britain
e-mail: dfh@maths.warwick.ac.uk