Constructions of Fischer's Baby Monster over fields of characteristic not 2

Richard A. Parker

DPMMS, 16, Mill Lane,
Cambridge CB2 1SB

and

Robert A. Wilson

School of Mathematics and Statistics,
The University of Birmingham,
Edgbaston, Birmingham B15 2TT

published in J. Algebra 229 (2000), 109–117

Abstract

In this paper we describe the computer construction of the representations of Fischer's Baby Monster simple group in 4371 dimensions over the fields of 3 and 5 elements. As applications we construct representations for the Thompson group which will assist in determining much of the 3-modular and 5-modular character tables.

1 Introduction

Fischer's Baby Monster group is the second largest of the 26 sporadic simple groups, and has order greater than 4×10^{33}. Many of its basic properties were described by Fischer [2], and its character table was computed by Hunt [3]. It was first constructed by Leon and Sims, essentially as a permutation group on some 10^{10} points.

In [9] the second author constructed the 4370-dimensional representation of the Baby Monster over $GF(2)$. At that time, the available computers were stretched to the limit by a representation of this size even over the field of two elements. A construction of such a representation over a larger field would have been almost impossible. Nowadays, however, this is well within range.

Over fields of characteristic not 2, the obvious subgroups to use for the construction are the large 2-local subgroups. Specifically, we first construct the appropriate representation of the subgroup $2^{1+22}{\cdot}Co_2$, then restrict to $2^{2+10+20}{:}M_{22}$, and finally adjoin an outer automorphism of order 3 to the latter group. We use the experience gained from constructing the Monster [5] to help with the construction of $2^{1+22}{\cdot}Co_2$. Readers who are unfamiliar with the basic nethod of such constructions are advised to consult [7]. All calculations here are performed with the Meataxe [6, 8].

The 4371-dimensional representation of B restricts to the subgroup $2^{1+22}{\cdot}Co_2$ as $2048 \oplus 2300 \oplus 23$. Here the 2048 denotes the unique extension of the unique faithful irreducible representation of 2^{1+22}, while 2300 denotes one of the two faithful monomial representations of this degree for the quotient $2^{22}{\cdot}Co_2$, and 23 denotes the smallest faithful irreducible for Co_2.

2 Construction of $2^{1+22}{\cdot}Co_2$

The strategy here is essentially the same as is described in [5] for the construction of $3^{1+12}{:}2{\cdot}Suz{:}2$. We first write down 22 generators for the group 2_+^{1+22} in its faithful irreducible representation, of degree 2048, and then extend to $2^{1+22}{\cdot}Co_2$, by adjoining elements mapping to two (standard) generators of the natural quotient Co_2.

We define $p = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and $d = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, and then put $p_i = I_{2^{i-1}} \otimes p \otimes I_{2^{11-i}}$, and similarly $d_i = I_{2^{i-1}} \otimes d \otimes I_{2^{11-i}}$ for each i with $1 \leq i \leq 11$. Then the p_i are permutation matrices and the d_i are diagonal matrices, together generating 2_+^{1+22}. The subgroups $\langle -1, p_i \mid 1 \leq i \leq 11 \rangle$ and $\langle -1, d_i \mid 1 \leq i \leq 11 \rangle$ are maximal elementary abelian subgroups, and the commutators satisfy $[p_i, d_j] = (-1)^{\delta_{ij}}$.

Next we take two generators for Co_2 as 22×22 matrices over $GF(2)$, and change basis so that the new basis vectors $v_1, \ldots, v_{22}$ correspond to the pairs $\pm p_i$ and $\pm d_i$. In other words we find a symplectic basis, consisting of isotropic vectors, for the underlying orthogonal space. This can be done by the method described in [5], but we do not need to be so careful in this relatively small representation, so we can use more brutal methods. Specifically, we found a subgroup 23:11, which fixes precisely two totally isotropic 11-dimensional subspaces, V_1, V_2, say, of the 22-dimensional orthogonal space V. Choosing bases for each of these, we calculate the matrix M of the symplectic form with respect to the corresponding basis of V. Finally we change basis on *one* of the two subspaces V_1, V_2, by multiplying it by M^{-1}. This has the effect of changing the symplectic form to the standard one $\begin{pmatrix} 0 & I_{11} \\ I_{11} & 0 \end{pmatrix}$. Thus the basis vectors correspond as required to $\pm p_i$, $\pm d_i$ in order.

Now for each generator g of Co_2, given as a 22×22 matrix over $GF(2)$, we need to construct a 2048×2048 matrix $\hat{g}$ which (up to signs) acts by conjugation on 2^{1+22} in the same way that g acts on the natural module V. Note first that the given basis of 2048-space may be defined (up to a single scalar) by taking the first basis vector to be a simultaneous eigenvector of $d_1, \ldots d_{11}$, with all eigenvalues 1, and defining the basis vectors to be the images of this eigenvector under $\langle p_1, \ldots, p_{11} \rangle$ (in a particular order).

Then we can calculate (up to sign) the images of $d_1, \ldots, d_{11}$ under $\hat{g}$. Choose signs arbitrarily, to obtain $e_1, \ldots, e_{11}$, say. Then find the simultaneous eigenvector v_0 of $e_1, \ldots, e_{11}$, with all eigenvalues 1. This is of course only defined up to arbitrary scalar multiplication. We show later how to choose it canonically, up to sign.

Next we calculate images of $p_1, \ldots, p_{11}$ under $\hat{g}$. Again we have 11 arbitrary choices of sign, and we obtain $q_1, \ldots, q_{11}$, say. Then we calculate the images of v_0 under $\langle q_1, \ldots, q_{11} \rangle$ (in the same order as above). These vectors now form the rows of a matrix giving a lift of g into $2^{1+22} \cdot Co_2$ (modulo scalars). Indeed, once we have lifted a generating set for Co_2, we have obtained a central product of a group of scalars with $2^{1+22} \cdot Co_2$. We can therefore remove the unwanted scalars by passing to the derived group.

3 The monomial part of the representation

The group $2^{1+22}{\cdot}Co_2$ has a unique subgroup of index 2300, and it has shape $2^2.2^{20}.(2 \times U_6(2){:}2)$. In particular, the derived quotient is 2^2. Thus there are just four linear characters, which can be induced up to $2^{1+22}{\cdot}Co_2$. The trivial character of course induces up to the permutation representation, while one other character has the 2^{1+22} in its kernel, so induces up to a proper monomial representation of the quotient Co_2. The remaining two both induce up to faithful monomial representations of $2^{22}{\cdot}Co_2$.

It is not immediately obvious which one of these is the correct one (i.e. the one which occurs in the desired representation of the Baby Monster), but they can be distinguished by measuring character values, and therefore if we make them both then we can tell which is the correct one. One of them is the action by conjugation on the 2300 pairs of elements in a particular conjugacy class in the normal subgroup 2_+^{1+22}. It turns out that this one is the wrong one!

In fact, of course, the permutation parts of all four representations are the same, and they differ only in the signs. Moreover, any sign in one of the three proper monomial representations is the product of the two corresponding signs in the other two representations. Therefore we can construct the required representation from the other two. In each case we attach signs to the permutation representation. In one case these signs come from the action of Co_2 on an appropriate orbit of 1-spaces in the Leech lattice, or (for easier calculation) the 23-dimensional irreducible representation over $GF(3)$. In the other case, the signs come from the action by conjugation of $2^{1+22}{\cdot}Co_2$ on a conjugacy class of 2×2300 involutions in the normal subgroup 2^{1+22}.

In practical terms, we need first to fix a numbering of the 2300 points, and use the same numbering in all the different representations. This can be done by a typical 'standard basis' argument, but the problem is that one then needs to store all 2300 'points', and in one of our cases each such 'point' is a 2048×2048 matrix. A naive implementation would therefore require in excess of a gigabyte of storage. We avoid this by precomputing a list of instructions for making all the points in the 'standard' ordering. Details of the algorithm used for this can be found in the next three sections.

4 Spanning trees of coset graphs

For many purposes it is useful to have an efficient method for listing all elements of a group, or, more generally, all cosets of a given subgroup. The algorithm described here is designed to explore such a coset graph (really a labelled digraph, with directed edges labelled by group generators) in a suitable small representation, and output a list of instructions for making all of the cosets, once each. This list of instructions can then be used in a very large representation, typically dividing the amount of work to be done by the number of generators. More importantly, it avoids the need to store all the cosets—in our example, when we ran the program for Co_2 acting on the 2300 cosets of $U_6(2)$:2, it never needed to store more than 5 cosets at a time.

The instructions are just four—**store**, **load**, **apply**, and **process**. The first two are memory functions: **store** n copies the current coset into memory location n, while **load** n copies the contents of memory location n to the current coset. The third creates a new current coset: **apply** n replaces the current coset by its image under generator n. Finally, **process** does whatever work the user requires to be done, to the current coset. It is the user's job to interpret the four instructions in the context of the required application.

5 The algorithm

The algorithm consists of two parts. In the first part, a spanning tree is computed, and in the second part, this is converted into a list of instructions for traversing the tree.

We assume that the input consists of a list of group generators, each of which is a permutation on n points, stored in image format. Thus each entry of each generator corresponds to a (directed) edge of the coset graph. The first part of the program deletes edges (in a particular order) until only a spanning tree remains.

We maintain a list of points in the order in which we find them, and process each in turn until every generator has been applied to every point. At each stage we also have a 'current point', which is usually the last point in the list at that stage. We apply the next generator to the current point. If this gives us a new point, we add it to the list, and move to the new point. If not, we delete the corresponding edge of the graph, and continue until we run out of generators to apply to the current point. When this happens, we return to the first point in the list which has not been completely processed.

The reason for doing things in this way, is in order to produce a thin straggly spanning tree, rather than a fat bushy tree. This then minimises the number of nodes which need to be remembered (i.e. stored) when traversing the tree.

In the second part of the algorithm, we essentially write out the edges of the spanning tree in a suitable order. At each stage, we first decide whether the current point needs to be stored: this happens if and only if we have visited the point for the first time, but more than one edge of the spanning tree leaves this point. We next apply the *last* possible generator to the current point. If no generator is left in the spanning tree at this point, we return to the *last* point which was stored.

6 Technicalities

First we take the permutation representation of Co_2 on 2300 points, and choose standard generators for the group as in [10]. Then we take our standard copy of the point stabilizer $U_6(2){:}2$ as in [11], and use this as the first coset. In other words, we start our algorithm with the fixed point of this subgroup. Running the algorithm now produces a list of instructions for making all 2300 points, in a particular order.

Consider next the monomial representation of Co_2 on 1-spaces in $GF(3)^{23}$. We again compute our standard copy of the subgroup $U_6(2){:}2$, and compute the fixed 1-space thereof. For each 1-space we choose (arbitrarily) the 'positive' direction to be the vector in the 1-space whose first nonzero coordinate is 1. Then for each such 1-space, we compute the images of the positive vector under each group generator, and see whether the result is positive or negative.

Finally we need to use these instructions in the 2048-dimensional representation. The 'points' are now pairs of 2048×2048 (monomial) matrices. At this point, we have to decide exactly what information we need to calculate in this representation. For our application, we have two generators for $2^{1+22}{\cdot}Co_2$, which are chosen to map onto standard generators for the quotient Co_2, and we need to attach signs to the corresponding permutations, in order to create monomial generators for $2^{22}{\cdot}Co_2$. The 4600 involutions are themselves monomial matrices, and for simplicity we choose the 'positive' one of each pair to be the one which has $+1$ in the top row. Then the signs are obtained by conjugating these monomial matrices by the group generators, and observing whether the top row has a $+1$ or -1.

Thus we can implement the four instructions very simply: **store** and **load** just copy files, while **apply** is just conjugation of matrices. Finally, **process** consists of two matrix conjugations, followed by processing of the top rows of the results.

In fact, there are two computational tricks available to simplify this. Firstly, we do not need to do full matrix multiplication just to get the top row. Secondly, we do not need to create the matrices by conjugation, because they are already determined up to sign by the corresponding vector in 2^{22}, so they can be built 'from scratch' in this way when necessary.

7 The full representation

At this stage we can create the full representation of the subgroup $2^{1+22}{\cdot}Co_2$ as $2048 \oplus 2300 \oplus 23$, that is the direct sum of the two representations constructed above with the 23-dimensional representation of Co_2. Everything so far can be done over any field of characteristic not 2, or even over $\mathbb{Z}[\frac{1}{2}]$.

The next step is to restrict to the subgroup $2^2.2^{20}.2^{10}M_{22}$, namely the centralizer of a suitable 4-group. This is easy, for example using the words given in [11], and we can chop up the representation with the Meataxe into its irreducible constituents for this subgroup. We find the following decomposition:

$$\begin{aligned} 2048 &\rightarrow 1024a \oplus 1024b \\ 2300 &\rightarrow 1024c \oplus 1232 \oplus 22a \oplus 22b \\ 23 &\rightarrow 22c \oplus 1 \end{aligned}$$

It is sufficient now to find a suitable element fusing the three constituents of degree 1024, and the three of degree 22. First we need to define, and find explicitly, suitable sets of standard generators for our subgroup, and then find the corresponding standard bases.

Note that each of the 1024-dimensional constituents represents the group modulo one of the three central involutions. Similarly each of the 22-dimensional constituents represents one of the three quotients $2^{10}M_{22}$.

8 Standard generators and standard bases

The group $C(2^2) = 2^{1+1+10+10+10}M_{22}$ has a somewhat subtle structure. Modulo the centre, the group has the shape $2^{10+20}{:}M_{22}$, in which the normal subgroup 2^{10+20} is a special group. Modulo the second centre, we have $2^{20}{:}M_{22}$, in which there are four classes of complements M_{22}. Lifting first to $2^{10+20}M_{22}$ the four classes do not split further. Now lifting to $2^{2+10+20}M_{22}$ we find that only one class lifts to M_{22}, while the other three lift to $2{\cdot}M_{22}$. In particular, there is a unique conjugacy class of subgroups M_{22} in $2^{1+1+10+10+10}M_{22}$.

Returning now to the explicit computations, first we find a subgroup M_{22} in $2^{1+1+10+10+10}M_{22}$, and find standard generators s_1, s_2 for it. It is somewhat harder to define suitable standard generators for the rest of the group. If we take an element of order 7 in M_{22}, it fixes a unique non-zero vector in each 2^{10} factor, so its full centralizer in $2^{2+10+20}$ has order 2^5. We can certainly find elements in this centralizer which are zero in one of the quotient groups $2^{10}{:}M_{22}$, and nonzero in the other two. These are therefore defined modulo a group N of order 8. It turns out that four of the elements in each such coset have order 2, while the other four have order 4. Let us choose representatives s_3, s_4 of order 2 in two of the three cosets, so that there are just 4^2 possible choices for s_3 and s_4.

It turns out that these choices fall into four orbits of size 4 under inner automorphisms, so that really there is a choice of four possibilities for the set $\{s_1, s_2, s_3, s_4\}$ of standard generators at this stage. We choose one arbitrarily, and now try to find the images of s_1, s_2, s_3, s_4 under an outer automorphism of order 3. We may assume that s_1 and s_2 are centralized by this automorphism, and modulo the group N of order 8, the generators s_3, s_4 are mapped to s_3s_4 and s_3, respectively. Again, inner automorphisms allow us to assume that s_4 is mapped to s_3, and s_3 is mapped to one of the four involutions in the coset Ns_3s_4. We then use relations in the group to determine which is the correct one of these cases, say s_3'.

Finally we put the entire representation into standard basis, first with respect to the generator list s_1, s_2, s_3, s_4, and again with respect to the list s_1, s_2, s_3', s_3. This gives us a matrix conjugating the first list of generators to the second.

9 Checking the cases

In practice, therefore, we adjoined an element of order 3 cycling the three consituents of degree 1024, and the three of degree 22. Thus the resulting group $2^{2+10+20}(M_{22} \times 3)$ has constituent degrees $3072 + 1232 + 66 + 1$. In particular, the number of degrees of freedom is $8 - 3 - 4 + 1 = 2$. This means that once we have determined the correct standard generators, over $GF(3)$, say, there are just $2^2 = 4$ cases to consider. As usual, we very quickly eliminate three of these, and what is left must be the Baby Monster.

Similarly, over $GF(5)$ there are $4^2 = 16$ cases to check, and again it is easy to eliminate all but one of these.

Finally, we want to find 'standard generators' for the Baby Monster in both these representations. We do this by following the instructions in [11], with some shortcuts provided by using the trace to help identify conjugacy classes. We found an element of order 30 with trace 1, which is therefore in class $30B$, and has 10th power, y say, in $3A$. We also found an element of order 52, whose 26th power, x say, is therefore in $2C$. Then by conjugating x and y by pseudorandom elements of the group we quickly find standard generators as defined in [11].

10 Applications

The impetus for this construction came from modular representation theory. Specifically, the restriction to the Thompson group Th is a uniserial module with structure $\begin{array}{c}248\\3875\\248\end{array}$, and the resulting 3875-dimensional module will be very helpful in calculating more of the 3-modular character table of Th. Jürgen Müller is working on this problem, by condensing the tensor square of this 3875-dimensional module.

A second application is to identifying conjugacy classes of elements given as words in the standard generators. Such elements can now be computed in the natural representations in characteristics 2, 3 and 5, and so the (rational) character values on the 4371-dimensional representation can be determined modulo 30.

Acknowledgements. The research described in this paper forms part of a joint project between Lehrstuhl D für Mathematik, RWTH Aachen and the School of Mathematics and Statistics in the University of Birmingham, supported by grants from the British Council and the Deutsche Akademische Austauschdienst.

References

[1] J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker and R. A. Wilson, *An ATLAS of Finite Groups*, Oxford University Press, 1985.

[2] B. Fischer, A note concerning {3,4}-transposition groups, *unpublished*.

[3] D. C. Hunt, The character table of the Baby Monster, *unpublished*.

[4] J. S. Leon and C. C. Sims, The existence and uniqueness of a simple group generated by $\{3,4\}$-transpositions, *Buil. Amer. Math. Soc.* **83** (1977), 1039–1040.

[5] S. A. Linton, R. A. Parker, P. G. Walsh and R. A. Wilson, Computer construction of the Monster, *J. Group Theory* **1** (1998), 307–337.

[6] R. A. Parker, The computer calculation of modular characters (The 'Meat-axe'), in *Computational Group Theory (ed. M. D. Atkinson)*, Academic Press, 1984, pp. 267–274.

[7] R. A. Parker and R. A. Wilson, Computer construction of matrix representations of finite groups over finite fields, *J. Symbolic Comput.* **9** (1990), 583–590.

[8] M. Ringe, *The C Meat-axe 2.3, documentation*, RWTH Aachen, 1995.

[9] R. A. Wilson, A new construction of the Baby Monster, and its applications, *Bull. London Math. Soc.* **25** (1993), 431–437.

[10] R. A. Wilson, Standard generators for sporadic simple groups, *J. Algebra* **184** (1996), 505–515.

[11] R. A. Wilson et al., A world-wide-web Atlas of Group Representations, http://www.mat.bham.ac.uk/atlas/