Calculating Cocyclic Hadamard Matrices in *Mathematica*: Exhaustive and Heuristic Searches

Dpto. Matemática Aplicada I, Universidad de Sevilla, Avda. Reina Mercedes s/n
41012 Sevilla, Spain
{valvarez, armario, mdfrau, real}@us.es

Abstract. We describe a notebook in *Mathematica* which, taking as input data a homological model for a finite group G of order $|G| = 4t$, performs an exhaustive search for constructing the whole set of cocyclic Hadamard matrices over G. Since such an exhaustive search is not practical for orders $4t \geq 28$, the program also provides an alternate method, in which an heuristic search (in terms of a genetic algorithm) is performed. We include some executions and examples.

Nowadays, most of Computer Algebra Systems (CAS) begin to be concerned with the computation of homological information. This is the case of GAP [13], MAGMA [17]. In a parallel way, some specific softwares for achieving calculations in homological algebra are being developed, as it is the case of KENZO [7] and HAP [15]. However none of them are concerned with the explicit calculation of Hadamard cocyclic matrices. In spite of that, some researchers have developed their own computations, working on different systems [6,11,12,14,2].

In this paper we present a notebook [1] in *Mathematica 4.0*, which implements the homological reduction method described in [4]. When exhaustive search is not feasible, the notebook provides a second routine for performing an heuristic search, which corresponds to the implementation of the genetic algorithm in [3]. The interested reader is referred to these papers for the theoretical background. Another non exhaustive method for finding out Hadamard cocyclic matrices is described in [5], in terms of image restorations.

The notebook takes as input data a homological model $^{\phi:}\bar{B}(\mathbb{Z}[G]) \underset{g}{\overset{f}{\rightleftarrows}} (hG, d)$ for a finite group G of order $|G| = 4t$. The term *homological model* refers to a contraction $^{\phi:}\bar{B}(\mathbb{Z}[G]) \underset{g}{\overset{f}{\rightleftarrows}} hG$ from the *reduced bar construction* of the group G (i.e. the reduced complex associated to the standard bar resolution [18]) to a differential graded module of finite type hG, so that $H_*(G) = H_*(hG)$ and the homology of hG may be effectively computed by means of Veblen's algorithm [19] (involving the Smith's normal forms of the matrices representing the differential operator).

In order to construct a basis for 2-coboundaries, the program needs to know the group law on G. To this end, the user must fix an ordering on the elements of G, say $G = \{g_1 = 1, \ldots, g_{4t}\}$. Now, a matrix P representing the group law in G may be constructed at once, so that $P(i, j) = k$ if and only if $g_i g_j = g_k$.

The only additional information which is needed is that of the diagrams

$$\bar{B}_1(\mathbb{Z}[G]) \xrightarrow{f} \mathcal{B}_1 \qquad\qquad \bar{B}_2(\mathbb{Z}[G]) \xrightarrow{f} \mathcal{B}_2$$
$$\downarrow Q \qquad\qquad\qquad\qquad \downarrow Q$$
$$\bar{\mathcal{B}}_1 \qquad\qquad\qquad\qquad \bar{\mathcal{B}}_2$$

That is, the matrices M_2 and M_3 representing the differentials operators d_2 and d_3 of hG, as well as the matrices F_1 and F_2 representing the maps $f_1 : \bar{B}_1(\mathbb{Z}[G]) \to \mathcal{B}_1$ and $f_2 : \bar{B}_2(\mathbb{Z}[G]) \to \mathcal{B}_2$.

In these circumstances, we may now determine exactly what the input and output data are.

INPUT DATA

– The matrices P, M_{i+1}, F_i representing the group law, d_{i+1} and f_i, $1 \leq i \leq 2$.
– The searching method to use: introduce 1 for an exhaustive search, anything else for a heuristic search.

OUTPUT DATA

– A full basis for normalized 2-cocycles over G.
– In case that an exhaustive search was chosen, the whole set of Hadamard cocyclic matrices over G. Otherwise, a few Hadamard cocyclic matrices, obtained from the heuristic search.

All the executions and examples included below have been worked out with aid of the *Mathematica 4.0* notebook which we have described here, running on a *Pentium IV 2.400 Mhz DIMM DDR266 512 MB*. Since some calculations obtained from the heuristic search are provided in [3], we focus on calculations obtained from an exhaustive search.

In the sequel, the elements of a product $A \times B$ are ordered as the rows of a matrix indexed in $|A| \times |B|$. For instance, if $|A| = r$ and $|B| = c$, the ordering is

$$< a_1 b_1, a_1 b_2, \ldots, a_1, b_c, a_2 b_1, a_2 b_2, \ldots, a_2 b_c, \ldots, a_r b_1, \ldots, a_r b_c >$$

We assume $\mathbb{Z}_k = \{0, 1, \ldots, k-1\}$ with additive law. Next we include a table with the number of cocyclic Hadamard matrices that we have found in each case.

t	$\mathbb{Z}_{4t}$	$\mathbb{Z}_2 \times \mathbb{Z}_{2t}$	$\mathbb{Z}_4 \times \mathbb{Z}_t$	$\mathbb{Z}_2^2 \times \mathbb{Z}_t$	D_{4t}	$\mathbb{Z}_{2t} \ltimes_f \mathbb{Z}_2$	$(\mathbb{Z}_t \ltimes_f \mathbb{Z}_2) \rtimes_\chi \mathbb{Z}_2$
1	2	6	2	6	6	6	6
2	0	**16**	**16**	168	32	**16**	168
3	0	**24**	**0**	24	72	**0**	72
4	0	**96**	192	1984	768	**0**	**272**
5	0	120	0	120	**2200**	120	**2200**

Here f denotes the 2-cocycle $f(g_i, g_j) = \begin{cases} \lceil \frac{t}{2} \rceil + 1 \text{ if } g_i = g_j = 1 \\ 0 \qquad\qquad \text{otherwise} \end{cases}$ and χ denotes the dihedral action $\chi(a, b) = \begin{cases} -b \text{ if } a = 1 \\ b \text{ if } a = 0 \end{cases}$

The computing time required in all cases was practically of the same order: less than 3 seconds for $1 \leq t \leq 3$, a couple of minutes for $t = 4$ and about 30 minutes for $t = 5$. The black entries correspond to new results, as far as we know (the case of $G_5^5 = D_{4.5}$ is included, since the computation of Flannery in [16], 2380, differs from ours, 2200).

References

1. V. Álvarez. http://mathworld.wolfram.com/HadamardSearch.html (2006). To appear.
2. V. Álvarez, J.A. Armario, M.D. Frau and P. Real. An algorithm for computing cocyclic matrices developed over some semidirect products. *AAECC-14 Proceedings*, LNCS **2227**, 287–296 (2001).
3. V. Álvarez, J.A. Armario, M.D. Frau and P. Real. A genetic algorithm for cocyclic Hadamard matrices. *AAECC-16 Proceedings*, LNCS **3857**, 144–153, (2006).
4. V. Álvarez, J.A. Armario, M.D. Frau and P. Real. Homological reduction method for constructing Hadamard cocyclic matrices. Communication submitted to the EACA-06 conference, Sevilla (2006).
5. A. Baliga and J. Chua. Self-dual codes using image resoration techniques. *Proceedings AAECC'14*. Eds. S. Boztas, I.E. Shparlinski. *Springer Lecture Notes in Computer Science*, Springer-Verlag, Heidelberg, (2001).
6. A. Baliga and K.J. Horadam. Cocyclic Hadamard matrices over $\mathbb{Z}_t \times \mathbb{Z}_2^2$. *Australas. J. Combin.*, **11**, 123–134, (1995).
7. X. Dousson, J. Rubio, F. Sergeraert and Y. Siret. The Kenzo program. Institute Fourier, Grenoble (19998). http://www-fourier.ujf-grenoble.fr/~sergeraert/Kenzo/
8. W. de Launey and K.J. Horadam. Cocyclic development of designs. *J. Algebraic Combin.*, **2** (3), 267–290, 1993. Erratum: *J. Algebraic Combin.*, (1), pp. 129, 1994.
9. W. de Launey and K.J. Horadam. Generation of cocyclic Hadamard matrices. *Computational algebra and number theory* (Sydney, 1992), volume **325** of *Math. Appl.*, 279–290. Kluwer Acad. Publ., Dordrecht, (1995).
10. D.L. Flannery. Calculation of cocyclic matrices. *J. of Pure and Applied Algebra*, **112**, 181–190, (1996).
11. D.L. Flannery. Cocyclic Hadamard matrices and Hadamard groups are equivalent. *J. Algebra*, **192**, 749–779, (1997).
12. D.L. Flannery and E.A. O'Brien. Computing 2-cocycles for central extensions and relative difference sets. *Comm. Algebra*, **28(4)**, 1939–1955, (2000).
13. The GAP group, 'GAP- Group, Algorithms and programming', School of Mathematical and Computational Sciences, University of St. Andrews, Scotland (1998).
14. J. Grabmeier, L.A. Lambe. Computing Resolutions Over Finite p-Groups. *Proceedings ALCOMA'99*. Eds. A. Betten, A. Kohnert, R. Lave, A. Wassermann. *Springer Lecture Notes in Computational Science and Engineering*, Springer-Verlag, Heidelberg, (2000).
15. G. Ellis. GAP package HAP, 'Homological Algebra Programming', http:// hamilton.nuigalway.ie/Hap/www/

16. K. Horadam. Progress in cocyclic matrices. *Congressus numerantium*, **118**, 161–171 (1996).
17. The MAGMA computational algebra system, http://magma.maths.usyd.edu.au
18. S. Mac Lane. Homology. *Classics in Mathematics*, Springer-Verlag, Berlin, (1995). Reprint of the 1975 edition.
19. O. Veblen. Analisis situs. A.M.S. Publications, **5** (1931).