

## ON THE COMPUTATION OF CERTAIN HOMOTOPICAL FUNCTORS

GRAHAM ELLIS

*Abstract*

This paper provides details of a MAGMA computer program for calculating various homotopy-theoretic functors, defined on finitely presented groups. A copy of the program is included as an Add-On. The program can be used to compute: the nonabelian tensor product of two finite groups, the first homology of a finite group with coefficients in the arbitrary finite module, the second integral homology of a finite group relative to its normal subgroup, the third homology of the finite  $p$ -group with coefficients in  $\mathbf{Z}_p$ , Baer invariants of a finite group, and the capability and terminality of a finite group. Various other related constructions can also be computed.

1. *Introduction*

This article provides details of a computer program for calculating various homotopy-theoretic functors defined on finitely presented finite groups. A separate text file contains a version of the program which has been implemented in the computer algebra language MAGMA V2.3 [9] [4] [3]. The text file can be downloaded and used (in the presence of MAGMA V2.3 software) to calculate:

1. The third homotopy group  $\pi_3 SK(G, 1)$  of the suspension of an Eilenberg–Mac Lane space  $K(G, 1)$  with finite fundamental group  $G$ .
2. The fourth homotopy group  $\pi_4 S^2 K(G, 1)$  of the double suspension of  $K(G, 1)$ . (Since  $K(G, 1)$  is connected, the homotopy groups  $\pi_{2+k} S^k K(G, 1)$  stabilize from  $k = 2$  onwards.)
3. The nonabelian tensor product  $G \otimes H$  of two finite groups  $G$  and  $H$  which act compatibly on each other. (Basic properties of this functor are explained in [5], and a comprehensive survey of related group-theoretic results can be found in [20]. Our MAGMA program has also been an aid to the computation of nonabelian tensor squares of certain infinite groups [1].)
4. The relative Schur multiplier  $M(G, N)$  which fits into an exact integral homology sequence

$$H_3(G) \rightarrow H_3(G/N) \rightarrow M(G, N) \rightarrow H_2(G) \rightarrow H_2(G/N)$$

arising from a finite group  $G$  with normal subgroup  $N$ . The exactness of this sequence implies that  $\ker(M(G, N) \rightarrow H_2(G))$  is a homomorphic image of  $H_3(G/N)$ , and that  $M(G, G)$  is the usual Schur multiplier of  $G$ . (Basic theory for the relative multiplier is developed in [13].)

---

Received 23rd June 1997, revised 15th January 1998; published 1st June 1998.

1991 Mathematics Subject Classification 18G50, 18G55, 20J99, 55U99

© 1998, Graham Ellis

5. The first homology  $H_1(G, B)$  of a finite group  $G$  with coefficients in a finite  $\mathbf{Z}G$ -module  $B$ . As a special case one can, in principle, calculate the third homology  $H_3(G, \mathbf{Z}_p)$  of a finite  $p$ -group  $G$  with coefficients in the trivial module  $\mathbf{Z}_p = \mathbf{Z}/p\mathbf{Z}$ ,  $p$  a prime. In practice,  $G$  needs to be of an extremely low order.
6. The Baer invariants

$$M^{(c)}(G) = \frac{R \cap \gamma_{c+1}(F)}{\gamma_{c+1}(R, F)}, \quad c \geq 1$$

where  $F$  is a free group whose quotient  $F/R$  is isomorphic to the finite group  $G$ , and where  $\gamma_1(F) = F$ ,  $\gamma_{i+1}(F) = [\gamma_i(F), F]$ ,  $\gamma_1(R, F) = R$ ,  $\gamma_{i+1}(R, F) = [\gamma_i(R, F), F]$ . The group  $M^{(1)}(G)$  is the usual Schur multiplier of  $G$ . (Basic theory for these invariants is developed in [22] and [7]. Our MAGMA program is used in [7] to compute  $M^{(2)}(G)$  for all groups  $G$  of order less than 32.)

Various other related constructions can also be calculated.

A very brief algebraic account of the above functors and related constructions, together with references to articles containing further details, is given in Section 2. The mathematical results that form the basis of our methods of computation are explained in Section 3. Neither Section 2 nor Section 3 contains any essentially new material. Installation instructions for our MAGMA program are given in Section 4. Some general remarks on the program are given in Section 5, while Section 6 contains detailed descriptions of all of the various procedures implemented in the program.

As mentioned above, a text file is attached to this paper. It contains: (i) our MAGMA program, (ii) a test input file, and (iii) a text copy of Section 6 (which can be used as an on-line help file). Further details of this text file are given in Section 4.

I would like to thank John Cannon for his very generous help in simplifying, shortening, and increasing the speed of an earlier version of our MAGMA program. I would also like to thank the referee for many helpful comments.

## 2. *An algebraic account of various homotopical constructions*

Let  $G$  and  $H$  be two (not necessarily finite) groups endowed with an action  $(g, h) \mapsto {}^g h$  of  $G$  on  $H$ , and an action  $(h, g) \mapsto {}^h g$  of  $H$  on  $G$ . We shall assume that  $G$  acts on itself by conjugation  $(g, g') \mapsto {}^g g' = gg'g^{-1}$ , and that  $H$  similarly acts on itself by conjugation. Let us suppose that the various actions are *compatible* in the sense that

$$\begin{aligned} ({}^g h)g' &= g({}^h(g^{-1}g')), \\ ({}^h g)h' &= h({}^g(h^{-1}h')) \end{aligned}$$

for  $g, g' \in G$ ,  $h, h' \in H$ . In keeping with our convention for conjugation, we write  $[x, y]$  to denote the commutator  $xyx^{-1}y^{-1}$  of two group elements  $x$  and  $y$ .

Following the work of R.K. Dennis [10], C. Miller [26], and A.S.-T. Lue [24], the tensor product  $G \otimes H$  was defined by R. Brown and J.-L. Loday [6] to be the group generated by symbols  $g \otimes h$  (for  $g \in G$ ,  $h \in H$ ), subject to the relations

$$\begin{aligned} gg' \otimes h &= ({}^g g' \otimes {}^g h)(g \otimes h), \\ g \otimes hh' &= (g \otimes h)({}^h g \otimes {}^h h'), \end{aligned}$$

for  $g, g' \in G, h, h' \in H$ . It follows from this definition that there exist homomorphisms

$$\lambda : G \otimes H \rightarrow G, \quad g \otimes h \mapsto g^h g^{-1},$$

$$\mu : G \otimes H \rightarrow H, \quad g \otimes h \mapsto {}^s h h^{-1}.$$

Taking  $G = H$ , and taking all actions to be conjugation, we define

$$J(G) = \ker(\lambda : G \otimes G \rightarrow G).$$

The following isomorphism is proved in [6].

**Theorem 1.** [6]  $\pi_3 SK(G, 1) \cong J(G)$ .

For finite groups  $G, H$  the tensor product  $G \otimes H$  and homomorphism  $\lambda$  can be computed using Function 11 in Section 6. Functions 6 and 8 can be used to compute the tensor square  $G \otimes G$  and group  $J(G)$ .

The *tensor centre* of the group  $G$  is defined as

$$Z^\otimes(G) = \{g \in G : g \otimes x = 1 \in G \otimes G \text{ for all } x \in G\}.$$

This central subgroup of  $G$  is of interest because of the following characterisation proved in [11].

**Proposition 2.** [11] *The tensor centre  $Z^\otimes(G)$  is the largest central subgroup of  $G$  for which any subgroup  $A \leq Z^\otimes(G)$  induces an isomorphism*

$$G \otimes G \cong (G/A) \otimes (G/A)$$

and an injection

$$J(G) \hookrightarrow J(G/A).$$

The tensor centre of a finite group can be computed using Function 10 in Section 6.

The group  $\pi_4 S^2 K(G, 1)$  is a quotient of  $\pi_3 SK(G, 1)$ . To describe this quotient algebraically we let  $\Delta(G)$  denote the subgroup of  $J(G)$  generated by those elements

$$(x \otimes y)(y \otimes x)$$

for  $x, y \in G$ , and set

$$\tilde{J}(G) = J(G)/\Delta(G).$$

The following isomorphism is proved in [6].

**Theorem 3.** [6]  $\pi_4 S^2 K(G, 1) \cong \tilde{J}(G)$ .

The group  $J(G)$  lies in the centre of  $G \otimes G$ . (This is best seen by proving, as in [6], that  $\lambda : G \otimes G \rightarrow G$  is a crossed module.) Hence  $\Delta(G)$  is central in  $G \otimes G$ , and we can define the *symmetric product* as

$$G \tilde{\otimes} G = G \otimes G / \Delta(G).$$

For finite  $G$  the group  $\tilde{J}(G)$  can be computed using Function 9 in Section 6.

The second homology group  $H_2(G)$  of  $G$  with integer coefficients is also a quotient of  $J(G)$ . Let  $\nabla(G)$  denote the subgroup of  $J(G)$  generated by those elements

$$x \otimes x$$

for  $x \in G$ , and set

$$M(G) = J(G)/\nabla(G).$$

The following isomorphism (modulo notation) is proved in [26].

**Theorem 4.** [26]  $H_2(G) \cong M(G)$ .

Function 13 in Section 6 invokes Theorem 4 in its computation of the second homology of a finite group. Two alternative methods of computing the second homology are used in Functions 18 and 19.

Motivated by Theorem 4 we recall from [6] that the *exterior product*  $M \wedge N$  of two normal subgroups  $M$  and  $N$  in  $G$  is the quotient of their tensor product

$$M \wedge N = (M \otimes N)/\nabla(M, N)$$

in which all actions are given by conjugation in  $G$ , and in which  $\nabla(M, N)$  denotes the central subgroup of  $M \otimes N$  generated by those elements  $x \otimes x$  with  $x \in M \cap N$ . We let  $x \wedge y$  denote the coset of  $x \otimes y$  in  $M \wedge N$ . The group  $\nabla(M, N)$  lies in the kernel of the homomorphism  $\mu: M \otimes N \rightarrow N$ ,  $x \otimes y \mapsto [x, y] = xyx^{-1}y^{-1}$ . Consequently there is an induced homomorphism

$$\partial: M \wedge N \rightarrow G, \quad x \wedge y \mapsto [x, y].$$

The *Schur multiplier*  $M(G, N)$  of  $G$  relative to a normal subgroup  $N$  can be defined as

$$M(G, N) = \ker(\partial: G \wedge N \rightarrow N).$$

An alternative, but equivalent, definition is given in [23]. The exact integral homology sequence involving this relative multiplier (quoted in the Introduction) is proved in [6]. Some basic theory for the relative multiplier is developed in [23], [12] and [13]. In particular, the *relative epicentre*

$$Z^*(G, N) = \{n \in N : x \wedge n = 1 \in G \wedge N \text{ for all } x \in G\}$$

is considered, and shown to be the largest central subgroup of  $G$  lying in  $N$  with the property that any subgroup  $A \leq Z^*(G, N)$  induces an injection  $M(G, N) \hookrightarrow M(G/A, N/A)$ . For finite  $G$ , Functions 14 and 15 in Section 6 can be used to compute  $M(G, N)$  and  $Z^*(G, N)$ .

The tensor product can be used to describe the first homology of  $G$  with coefficients in a  $\mathbf{Z}G$ -module  $B$ . For this we consider  $B$  as a group with a  $G$ -action, and consider the group  $G$  to have a trivial  $B$ -action. The following isomorphism is proved in [17].

**Theorem 5.** [17]  $H_1(G, B) \cong \ker(\mu: G \otimes B \rightarrow B)$ .

Suppose that  $F$  is a free group with normal subgroup  $R$  such that the quotient  $F/R$  is isomorphic to  $G$ . For any integer  $q \geq 0$  the abelian group  $R/[R, R]R^q$  is a  $\mathbf{Z}G$ -module with the action of  $G$  given by conjugation in  $F$ . There are well-known [19] reduction isomorphisms  $H_3(G) \cong H_1(G, R/[R, R])$  and  $H_3(G, \mathbf{Z}_q) \cong H_1(G, R/[R, R]R^q)$ . We consequently have the following description of the third homology of the group  $G$  with coefficients in the trivial module  $\mathbf{Z}_q = \mathbf{Z}/q\mathbf{Z}$ .

**Theorem 6.**  $H_3(G, \mathbf{Z}_q) \cong \ker(\mu: G \otimes (R/[R, R]R^q) \rightarrow R/[R, R]R^q) \quad \text{for } q \geq 0$ .

Function 16 in Section 6 invokes Theorem 6 in its computation of  $H_3(G, \mathbf{Z}_p)$  for a finite  $p$ -group  $G$ .

The Baer invariants  $M^{(c)}(G)$  described in the Introduction are analogues of the second integral homology group  $H_2(G)$ . Function 18 in Section 6, which calculates  $M^{(c)}(G)$  for finite  $G$ , requires as part of the input data an integer  $q$  that is divisible by the exponent of  $M^{(c)}(G)$ . The following results are useful for determining such an integer  $q$ .

**Theorem 7.** [8] *Let  $G$  be a finite group of prime-power exponent  $p^e$  and nilpotency class  $k \geq 2$ . Then  $\exp(M^{(c)}(G))$  divides  $p^{e(k-1)}$ .*

**Proposition 8.** [8] *If  $N$  is a normal subgroup of a finite group  $G$  then  $\exp(M^{(c)}(G/N))$  divides  $\exp(M^{(c)}(G)) \times \exp(N \cap \gamma_{c+1}(G)/\gamma_{c+1}(N, G))$ .*

Other results for determining the exponent of  $M^{(c)}(G)$  are given in [8].

The invariant  $M^{(2)}(G)$  can in fact be computed without prior knowledge of  $\exp(M(G))$ , thanks to the following isomorphism, full details of which are given in [7]. (For non-prime-power groups  $G$  this method seems to be generally more efficient.)

**Theorem 9.** [7] *There is an isomorphism*

$$M^{(2)}(G) \cong \ker(\mu: (G \wedge G) \wedge G \rightarrow G)/\tau(G)$$

where  $\tau(G)$  is the normal subgroup of  $(G \wedge G) \wedge G$  generated by the elements

$$((x \wedge y) \wedge^y z)((y \wedge z) \wedge^z x)((z \wedge x) \wedge^x y)$$

for  $x, y, z \in G$ .

One can write a short procedure based on Theorem 9, using Function 11 in Section 6, to compute  $M^2(G)$  for finite  $G$  (cf. Function 7 which computes a preimage of  $M^2(G)$ ).

In connection with the Baer invariants of  $G$  we define the *upper epicentral series*  $\mathbf{Z}_1^*(G) \leq \mathbf{Z}_2^*(G) \leq \dots \leq G$  by choosing a free group  $F$  such that  $G \cong F/R$ , and setting  $\mathbf{Z}_c^*(G)$  equal to the canonical image in  $G$  of the  $c$ -th term of the upper central series of the group  $F/\gamma_{c+1}(R, F)$ . Function 18 in Section 6 yields the upper epicentral series of a finite group. Among other things, the epicentral series is useful in determining the *capability* of a group: we say that  $G$  is  $c$ -capable if there exists some group  $H$  such that  $G \cong H/\mathbf{Z}_c(H)$ , where  $\mathbf{Z}_c(H)$  denotes the  $c$ -th term of the upper central series of  $H$ .

**Proposition 10.** [7] *A group  $G$  is  $c$ -capable if and only if  $\mathbf{Z}_c^*(G)$  is trivial.*

The capability of groups is of relevance to the classification of prime-power groups (see [18]). A modern account of the basic theory of 1-capable groups can be found in [2].

A dual notion, also relevant to the classification of prime-power groups, is due to L. Evens [15] who defined a group  $G$  to be *terminal* if (i) it is nilpotent of some class  $c$ , and (ii) there exists no group  $H$  such that  $G \cong H/\gamma_{c+1}H$ . This dual notion is also discussed in [2].

The boolean-valued Functions 20 and 21 in Section 6 can be used to determine the capability and terminality of a finite group.

### 3. Methods of computation

The tensor product  $G \otimes H$  of two finite groups is known to be finite (provided that the actions are compatible). Thus, in principle, its presentation could be entered directly into a Todd–Coxeter procedure or nilpotent  $p$ -quotient algorithm. (Indeed, this approach was taken in [5] in order to determine the tensor square  $G \otimes G$  of certain groups  $G$  of order up

to about  $|G| = 50$ .) However, there are  $|G \times H|$  generators in this presentation, and this number of generators can be a problem even when  $G$  and  $H$  are fairly small groups. One method for reducing the number of generators is suggested in [14], and illustrated there by means of a CAYLEY computer program. The MAGMA program in the attached text file (Appendix A) is a development of that CAYLEY program.

In order to explain the method for reducing the number of generators we continue with the assumption that  $G$  and  $H$  are arbitrary groups which act compatibly on each other. Let  $G * H$  be the free product of  $G$  and  $H$ , and let  $J$  denote the normal subgroup of  $G * H$  normally generated by the elements

$$x[g, h]x^{-1}[\bar{h}, \bar{g}]$$

for  $g \in G, h \in H, x \in G \cup H$ , where  $\bar{g} = xgx^{-1} \in G$  and  $\bar{h} = xhx^{-1} \in H$ . The following theorem is due to N.D. Gilbert and P. Higgins [16].

**Theorem 11.** [16] *There is an isomorphism*

$$((G \otimes H) \rtimes H) \rtimes G \cong G * H / J$$

where  $\rtimes$  denotes a semi-direct product. This isomorphism restricts to an isomorphism

$$G \otimes H \cong \bar{G} \cap \bar{H}$$

where  $\bar{G}$  and  $\bar{H}$  are the normal closures in  $G * H / J$  of  $G$  and  $H$ .

If  $G$  and  $H$  have presentations  $\langle X \mid R \rangle$  and  $\langle Y \mid S \rangle$  then their free product  $G * H$  is presented by  $\langle X \cup Y \mid R \cup S \rangle$ . Thus for finitely presented groups Theorem 11 yields  $G \otimes H$  as a subgroup of a finitely presented group on  $|X \cup Y|$  generators. This is a significant improvement on  $|G \times H|$  generators. Our MAGMA program computes  $G \otimes H$  by first applying the Todd–Coxeter procedure or nilpotent  $p$ -quotient algorithm to a presentation of  $G * H / J$ , and then computing the subgroup  $\bar{G} \cap \bar{H}$ . However, the above presentation of  $F * G / J$  contains many unnecessary relations (that is, the description of  $J$  contains many unnecessary normal generators). Our MAGMA program uses the more efficient description of  $J$  given in the next proposition.

Let  $X$  be a generating set for  $G$ , and  $Y$  a generating set for  $H$ . Let  $Z_i(G)_H$  be the subgroup of  $G$  consisting of those elements of  $G$  that lie in the  $i$ -th term  $Z_i(H \rtimes G)$  of the upper central series of the semi-direct product  $H \rtimes G$ . Let  $D_i(G)_H$  denote some generating set for the subgroup  $Z_i(G)_H$ , and let  $T_i(G)_H$  denote some transversal of  $Z_i(G)_H$  in  $G$ . For  $c \geq 1$  set

$$X_c = D_1(G)_H \cup D_2(G)_H \cup \dots \cup D_c(G)_H \cup T_c(G)_H,$$

$$Y_c = D_1(H)_G \cup D_2(H)_G \cup \dots \cup D_c(H)_G \cup T_c(H)_G.$$

The following proposition, which is a development of material in [14], has been proved by Aidan McDermott in his thesis [25].

**Proposition 12.** [25] *Fix some  $c \geq 1$ . The subgroup  $J$  is normally generated by the elements*

$$x[g, h]x^{-1}[\bar{h}, \bar{g}]$$

for  $g \in X, h \in Y, x \in X_c \cup Y_c$ , where  $\bar{g} = xgx^{-1} \in G$  and  $\bar{h} = xhx^{-1} \in H$ .

Theorem 11 and Proposition 12 are used in Functions 11 and 17 in Section 6.

In order to efficiently compute the quotient  $\tilde{J}(G) = J(G)/\Delta(G)$  we consider the elements

$$\{x, y\} = (x \otimes y)(y \otimes x)$$

for  $x, y \in G$ . Proposition 4.10 in [6] implies the identities

$$\{xx', y\} = \{x, y\}\{x', y\},$$

$$\{x, yy'\} = \{x, y\}\{x, y'\},$$

for  $x, x', y \in G$ . This bilinearity leads to the following more efficient description of the subgroup  $\Delta(G)$  of  $G \otimes G$ .

**Proposition 13.** *Let  $W$  be a subset of  $G$  which generates  $G^{ab}$ . Then  $\Delta(G)$  is generated by the elements*

$$(x \otimes y)(y \otimes x)$$

for  $x, y \in W$ .

Proposition 13 is used in Function 9 in Section 6.

We similarly derive the following description of the central subgroup  $\nabla(M, N)$  of  $M \otimes N$ , where  $M$  and  $N$  are normal subgroups of  $G$ .

**Proposition 14.** *Let  $W$  be a subset of  $M \cap N$  which generates  $M \cap N/[M, N]$ . Then  $\nabla(M, N)$  is generated by the elements*

$$(x \otimes y)(y \otimes x)$$

and

$$(x \otimes x)$$

for  $x, y \in W$ .

Proposition 14 is used in Function 14 in Section 6.

In [7] the Baer invariant  $M^{(2)}(G)$  was computed for certain non prime-power groups  $G$  via the isomorphism  $M^{(2)}(G) \cong \ker(\mu: (G \wedge G) \wedge G \rightarrow G)/\tau(G)$  of Theorem 9. For this computation it is useful to consider the elements

$$\langle x, y, z \rangle = ((x \wedge y) \wedge^y z)((y \wedge z) \wedge^z x)((z \wedge x) \wedge^x y)$$

in  $(G \wedge G) \wedge G$ , where  $x, y, z \in G$ . If  $G$  is a free group then, as explained in [11], the triple Pontryagin product

$$G^{ab} \times G^{ab} \times G^{ab} \rightarrow H_3(G^{ab}) \cong \ker(\mu: (G \wedge G) \wedge G \rightarrow G)$$

can be identified with the function

$$G^{ab} \times G^{ab} \times G^{ab} \rightarrow (G \wedge G) \wedge G, (\bar{x}, \bar{y}, \bar{z}) \mapsto \langle x, y, z \rangle.$$

Since this Pontryagin product is trilinear, it follows that for an arbitrary group  $G$  the bracket  $\langle x, y, z \rangle$  is trilinear. This yields the following efficient description of  $\tau(G)$ .

**Proposition 15.** *Let  $W$  be a subset of  $G$  which generates  $G^{ab}$ . Then  $\tau(G)$  is generated by the elements  $\langle x, y, z \rangle$  for  $x, y, z \in W$ .*

Proposition 15 can be used to construct an analogue of Function 7 in Section 6 for computing  $M^2(G)$ .

For  $c \geq 1$  our MAGMA program computes  $M^{(c)}(G)$  from a presentation  $\mathbf{P} = \langle x_1, \dots, x_s \mid r_1, \dots, r_t \rangle$  of  $G$ , and an integer  $q$  divisible by the exponent of  $M^{(c)}(G)$ . We set

$$\Theta_1^q(\mathbf{P}) = \langle x_1, \dots, x_s \mid [r_j, x_i], (r_j)^q \text{ for } 1 \leq i \leq s, 1 \leq j \leq t \rangle,$$

$$\Theta_{c+1}^q(\mathbf{P}) = \Theta_1^q(\Theta_c^q(\mathbf{P})),$$

and let  $F_c^q$  be the group presented by  $\Theta_c^q(\mathbf{P})$ . Our program uses the following isomorphism, proved in [7].

**Theorem 16.** [7]  $M^{(c)}(G) \cong \ker(\rho: \gamma_{c+1}(F_{c+1}^q) \rightarrow G)$  where  $\rho$  is the canonical homomorphism.

Theorem 16 is the basis of Function 18 in Section 6.

#### 4. Installation instructions

An attached text file (Appendix A) comprises three sections which contain, respectively: (i) a help file, (ii) our MAGMA program, and (iii) a test input file. The program and test input file can be run on any machine with MAGMA (Version 2.3) installed [3]. It is important to note that they will not run with earlier versions of MAGMA.

The help file in Section I should be extracted and saved under the file-name *homotopy-help.m*. The program in Section II should be saved under the file-name *homotopy.m*. The test file in Section III should be saved under the file-name *homotopytest.m*. These three files need to be stored in a single directory.

To run the program, users should start MAGMA, and then type `load "homotopy.m" ;`. The help file can be accessed by typing `help( ) ;`. To test that the program runs correctly on a given computer, type `load "homotopytest.m" ;`. This test file will call the various homotopy functions, apply them to particular groups, and then compare results with previously calculated data. The test file may take about ten minutes to run, and will conclude with a declaration of any detected errors.

#### 5. General remarks about the program

The program contains a number of homotopy-theoretic functions, each of which calls certain variables and returns other variables. For instance, the command `T := TensorSquare(G,p) ;` will call variables  $G$  and  $p$ , and return a variable under the name  $T$ . In this example  $G$  must be a finitely presented finite group, and  $p$  a non-negative integer. The variable  $T$  will represent an “enumerated” version of the tensor square  $G \otimes G$ . The basic group-theoretic properties of  $G \otimes G$  (such as order, exponent, nilpotency class, abelian quotient invariants) can then be determined using standard procedures on  $T$ . A full list of the homotopy-theoretic functions is given in Section 6.

Our program needs to call various standard group theory functions, such as the Todd–Coxeter procedure and the Nilpotent Quotient algorithm. Thus, any “stand-alone” C implementation of our program would to some extent involve “re-inventing the wheel”. It seems more reasonable to implement the program in a computer algebra language such as MAGMA or GAP in which a number of group theory procedures already exist. In this article



we have opted for MAGMA. However, Aidan McDermott has written a GAP version of part of the program in his thesis [25]. Even with the enormous help of such computer algebra languages, a number of technicalities the program. The nature of these technicalities differs between MAGMA and GAP. But in both cases it seems to be a non-trivial step from the ideas explained in this paper to the production of efficient code. For this reason we have attached a complete version of our program which can be run (and read) by the reader.

The structure of the program is simplified by introducing the concept of an *enumerated group*. Informally, such a group is a computer representation  $eG$  of a finite group  $G$  for which: (i) we can iterate over the elements in  $eG$ , and (ii) for any two elements  $x, y$  in  $eG$  there is a unique element in  $eG$  representing their multiple  $xy$ . More precisely, the term *enumerated group* is used throughout the program in three very specific senses: in the presence of a prime  $p$  it means a group obtained as the output from MAGMA's nilpotent  $p$ -quotient algorithm applied to a finitely presented group; in a situation where  $p = 1$  it means a finite permutation group (often obtained as the output from MAGMA's Todd–Coxeter procedure); in the absence of  $p$  it can have either of the previous meanings, or it can mean a finite matrix group.

The original motivation for our program was the desire to compute the kernel

$$K(G) = \ker(G \otimes (G \otimes G) \rightarrow G)$$

of the homomorphism from the tensor cube of a finite group  $G$  to  $G$ . Here the action of  $G$  on  $G \otimes G$  is defined on generators by

$${}^g(x \otimes y) = ({}^g x) \otimes ({}^g y)$$

for  $g, x, y \in G$ . An element  $\tau \in G \otimes G$  is understood to act on  $g \in G$  by

$${}^\tau g = \partial(\tau)g\partial(\tau)^{-1}$$

where  $\partial: G \otimes G, x \otimes y \mapsto xyx^{-1}y^{-1}$ . These actions are shown to be well-defined in [5]. A certain quotient of  $K(G)$  provides the calculations of the Baer invariants  $M^{(2)}(G)$  listed in [7] (cf. Theorem 9). In order to illustrate the nature of our program, we present a sketch algorithm for the computation of  $K(G)$ :

- Input a finite presentation for a finite group  $G$ . Also input a non-negative integer  $p$ .
- Construct an enumerated version  $eG$  of  $G$ , together with the ‘isomorphisms’  $\phi: G \rightarrow eG$  and  $\phi^{-1}: eG \rightarrow G$ .
- Construct the function

$$GactG: G \times G \rightarrow G, (x, y) \mapsto \phi^{-1}(\phi(xyx^{-1})).$$

(Since  $G$  is “represented” by a free group, the word  $\phi^{-1}(\phi(xyx^{-1}))$  will generally be shorter than  $xyx^{-1}$ .)

- Let  $H$  be an isomorphic copy of  $G$ , and construct a finite presentation for the free product  $G * H$ .
- Use Proposition 12, the action  $GactG$ , and the presentation for  $G * H$  to construct a finite presentation of the group  $G * H/J$ . (As explained in Section 3, the intersection of the normal closures of  $G$  and  $H$  in  $G * H/J$  is isomorphic to  $G \otimes G$ .)
- Construct an enumerated version  $e(G * H/J)$  of the group  $G * H/J$ , together with “isomorphisms”  $\psi: G * H/J \rightarrow e(G * H/J)$  and  $\psi^{-1}: e(G * H/J) \rightarrow G * H/J$ .
- Construct the subgroup  $[\psi(G), \psi(H)]$  of  $e(G * H/J)$  generated by commutators of the form  $[\psi(g), \psi(h)]$ . (This subgroup is an enumerated version of  $G \otimes G$ .)

- Construct a finitely presented group  $[G, H]$  isomorphic to  $[\psi(G), \psi(H)]$ , together with the “inclusion homomorphism”  $\iota: [G, H] \rightarrow e(G * H / J)$  and the “isomorphism”  $\iota^{-1}: [\psi(G), \psi(H)] \rightarrow [G, H]$ .
- Construct the functions

$$[G, H]\text{act}G: [G, H] \times G \rightarrow G, (x, y) \mapsto \psi^{-1}(\iota(x)\psi(y)\iota(x^{-1}))$$

$$G\text{act}[G, H]: G \times [G, H] \rightarrow [G, H], (y, x) \mapsto \iota^{-1}(\psi(y)\iota(x)\psi(y^{-1})).$$

- Let  $L$  be an isomorphic copy of  $G$ , and construct a finite presentation for the free product  $[G, H] * L$ .
- Use Proposition 12, the actions  $[G, H]\text{act}G$  and  $G\text{act}[G, H]$ , and the presentation for  $[G, H] * L$  to construct a finite presentation of the group  $[G, H] * L / J'$  in which the intersection of the normal closures of  $[G, H]$  and  $L$  is (by Theorem 11) isomorphic to  $(G \otimes G) \otimes G$ .
- Construct an enumerated version  $e([G, H] * L / J')$  of the group  $[G, H] * L / J'$ , together with “isomorphisms”  $\nu: [G, H] * L / J' \rightarrow e([G, H] * L / J')$  and  $\nu^{-1}: e([G, H] * L / J') \rightarrow [G, H] * L / J'$ .
- Construct the subgroup  $[\nu([G, H]), \nu(L)]$  of  $e([G, H] * L / J')$  which is generated by the appropriate commutators. (This subgroup is an enumerated version of  $(G \otimes G) \otimes G$ .)
- Construct the enumerated group  $eK(G)$  consisting of all those elements in  $[\nu([G, H]), \nu(L)]$  which get sent to the identity by homomorphism  $e([G, H] * L / J') \rightarrow eG$  induced by  $[G, H] * L \rightarrow G, [g, h] * l \mapsto [g, h]l$ .

The MAGMA language is near enough to a generic pseudocode, and so more precise details on this sketch algorithm can be obtained by reading our MAGMA program. Note that the program employs two helpful conventions: (i) an enumerated group  $eG$  is always prefixed by a lower-case “e”, whereas a finitely presented group  $G$  (i.e., a presentation of a group) is not; (ii) a homomorphism from a group  $G$  to a group  $H$  is (nearly) always denoted by  $G\text{hom}H$ .

We shall now try to give some indication of the capabilities and limitations of our program. All timings refer to the CPU time on a Digital AlphaStation 200<sup>(4/100)</sup> with a 100 MHz processor and 64 MB of memory.

Let us first remark that the test input file contained in the attachment calls most of our homotopical functions and applies them to various groups. The test file takes 792 seconds to run. However, 436 seconds of this CPU time is taken up by the function  $J\text{tilde}(G, 1)$  which invokes the Todd–Coxeter procedure in its computation of  $\tilde{J}(G)$  for the group  $G = \langle x, y \mid x^2 = y^8 = (xy)^2 = 1 \rangle$  of order 16. Since  $G$  is a 2-group we could use  $J\text{tilde}(G, 1)$  as an alternative method for computing  $\tilde{J}(G)$ ; this alternative invokes the p-quotient algorithm in place of the Todd–Coxeter procedure, and takes just 3 seconds to run!

One of the authors of [1] has recently been working on a generalisation of the results in that paper. As part of this work he has run the function  $\text{TensorSquare}(G, 3)$  on the  $d$ -generator exponent-three Burnside groups  $G = B(d, 3)$  for  $d = 3, 4, 5$ , in the hope of determining the exponent, nilpotency class, and number of generators of  $B(d, 3) \otimes B(d, 3)$ . For  $d = 3$  he used the presentation given in the example of Section 6; the group  $B(3, 3)$  has order  $3^7$ , its tensor square has order  $3^{20}$ , and the function took 59 seconds to complete. For  $d = 4$  he used a presentation on 4 generators and 56 relations; the group  $B(4, 3)$  has order

$3^{14}$ , and the function took just under one hour to complete. For  $d = 5$  he used a presentation on 5 generators and 152 relations; the group  $B(5, 3)$  has order  $3^{25}$ , and the function ran for over five days without completing. In this last case the program was stopped in order to allow other jobs to run.

The program has been used in [7] to compute the Baer invariants  $M^2(G)$  of all groups  $G$  of order less than 32. For the non-prime-power groups we computed  $M^2(G)$  using output from the function `TensorCube(G, 1)`. This function took up to an hour to complete on some of the more difficult groups (cf. the above comments on `Jtilde(G, 1)`.) (We should remark that we know of no other programs/computer software for computing the Baer invariants  $M^c(G)$  of finite groups when  $c \geq 2$ .)

The program's least successful feature is the function `ThirdHomologyModP(G, p)` for determining the dimension of the vector space  $H_3(G, \mathbf{Z}_p)$ , where  $p$  denotes a prime. We have encountered difficulties when running this function with certain groups  $G$  of order 64 and  $p = 2$ , in that a parameter in MAGMA's p-quotient algorithm was exceeded. Even allowing for an easily made adjustment in this parameter, we feel it is unlikely that the function would work for groups of any significantly larger prime-power order. (We should remark that J.F. Carlson and J.J. Cannon have been developing computer methods for determining the cohomology rings  $H^*(G, \mathbf{Z}_p)$  of prime-power groups  $G$ . As of April 1, 1997 they had computed these rings for 250 of the 267 groups of order 64. Further details can be found at <http://www.math.uga.edu/jfc/groups/cohomology.html>.)

### 6. Summary of functions implemented in the program

The following is a list of our MAGMA functions for calculating a variety of homotopical functors and related constructions, which are defined on finitely presented finite groups. Many of the functions require an input variable  $p$ . If all of the groups called by the function are prime-power groups then  $p$  should be set equal to this prime; otherwise set  $p = 1$ . When  $p$  is prime, the p-quotient algorithm is used to enumerate finitely presented groups (fp-groups); when  $p = 1$ , the Todd-Coxeter procedure is used.

1. `EnumeratedGroup(G, p) = (eG, GhomeG, eGhomG)`  
`FpGrp, RngIntElt → EnumGrp, Map, Map`  
 =====  
 Given a finite fp-group  $G$ , this function yields an enumerated version  $eG$  of the group, together with the 'isomorphisms'  $GhomeG: G \rightarrow eG$  and  $eGhomG: eG \rightarrow G$ .
  
2. `EnumeratedKernel(delta) = K`  
`Map → EnumGrp`  
 =====  
 Given a homomorphism  $\delta: G \rightarrow H$  from an enumerated group  $G$  to any group  $H$ , this function returns the kernel of  $\delta$  as an enumerated group  $K$ . This function should be used only in situations where MAGMA's internal `Kernel` function is undefined.
  
3. `PresentedGroup(eG) = (G, GhomeG, eGhomG)`  
`EnumGrp → FpGrp, Map, Map`  
 =====  
 Given an enumerated group  $eG$ , this function yields an fp-group  $G$  together with "isomorphisms"  $GhomeG: G \rightarrow eG$  and  $eGhomG: eG \rightarrow G$ .

4. SelfAction(G, p) = (GactG, NiceGeneratorsG)  
 FpGrp, RngIntElt → Map, Set  
 =====

Given a finite fp-group G, this function yields the mapping

$$\text{GactG: } G \times G \rightarrow G, \langle x, y \rangle \mapsto x^{-1}yx$$

from the cartesian product  $G \times G$  into  $G$ . It also yields a “nice” generating set for  $G$ . (More precisely, the generating set NiceGeneratorsG has the form  $D_1(G) \cup D_2(G) \cup \dots \cup D_c(G) \cup T_c(G)$  where  $D_i(G)$  is a generating set for the  $i$ th term  $Z_i(G)$  of the upper central series of  $G$ , and  $T_c(G)$  is a transversal of  $Z_c(G)$  in  $G$ . Such a generating set is called by Function 11, cf. Proposition 12.)

5. Action(GhomQ, HhomQ, p) = (HactG, NiceGeneratorsG)  
 Map, Map, RngIntElt → Map, Set  
 =====

Given finite fp-groups  $G, H$  and an enumerated group  $Q$ , and a normal inclusion  $\text{GhomQ: } G \hookrightarrow Q$ , and any homomorphism  $\text{HhomQ: } H \rightarrow Q$ , this function produces the mapping

$$\text{HactG: } H \times G \rightarrow G, \langle x, y \rangle \mapsto \text{GhomQ}^{-1}(\text{HhomQ}(x^{-1})\text{GhomQ}(y)\text{HhomQ}(x))$$

from the cartesian product  $G \times H$  to  $G$  given by conjugation in  $Q$ . It also produces a ‘nice’ generating set for  $Q$ . The groups  $G, H, Q$  can be accessed by the commands  $G := \text{Domain}(\text{GhomQ}); H := \text{Domain}(\text{HhomQ}); Q := \text{Codomain}(\text{GhomQ});$ . In order to obtain the conjugation action of a group  $G$  on itself, the more efficient function SelfAction should be used. (The generating set NiceGeneratorsG is of the form  $D_1(G)_H \cup D_2(G)_H \cup \dots \cup D_c(G)_H \cup T_c(G)_H$  where  $D_i(G)_H$  is a generating set for  $Z_i(G)_H = H \cap Z_i(H \rtimes G)$ , and  $T_c(G)_H$  is a transversal of  $Z_c(G)_H$  in  $G$ . Such a generating set is called by Function 11, cf. Proposition 12.)

6. TensorSquare(G, p) = (GtensorG, delta)  
 FpGrp, RngIntElt → EnumGrp, Map  
 =====

This function is defined for any finite fp-group  $G$ . It yields an enumerated version GtensorG of the tensor square  $G \otimes G$ . It also yields the homomorphism

$$\text{delta: } \text{GtensorG} \rightarrow G$$

which sends a tensor to the corresponding commutator in  $G$ .

7. TensorCube(G, p) = (TCube, delta)  
 FpGrp, RngIntElt → EnumGrp, Map  
 =====

This function is defined for any finite fp-group  $G$ . It yields an enumerated version Tcube of the triple tensor product  $(G \otimes G) \otimes G$ . The function also returns the homomorphism

$$\text{delta: } \text{TCube} \rightarrow G$$

which sends a triple tensor to the corresponding triple commutator in  $G$ . (The tensor square is understood to act on  $G$  by conjugation via the homomorphism  $\delta: G \otimes G \rightarrow G$ . The group  $G$  is understood to act on the tensor square by the diagonal action  ${}^g(x \otimes y) = {}^g x \otimes {}^g y$ .)

8.  $Jgroup(G, p) = JG$   
 $FpGrp, RngIntElt \rightarrow EnumGrp$   
 =====

This function is defined for any finite fp-group  $G$ . It yields the kernel  $JG$  of the homomorphism  $\delta: G \otimes G \rightarrow G$  which sends each tensor to the corresponding commutator. The group  $JG$  is returned as an enumerated group.

9.  $Jtilde(G, p) = JtildeG$   
 $FpGrp, RngIntElt \rightarrow EnumGrp$   
 =====

This function is defined for any finite fp-group  $G$ . It yields the quotient of  $JG$  by those elements  $(x \otimes y)(y \otimes x)$  for  $x, y \in G$ . The quotient is returned as an enumerated group  $Jtilde$ .

10.  $TensorCentre(G, p) = (TC, phi)$   
 $FpGrp, RngIntElt \rightarrow EnumGrp, Map$   
 =====

This function is defined for any finite fp-group  $G$ . It yields an enumerated version  $TC$  of the tensor centre  $Z^{\otimes}(G)$ . It also yields the inclusion  $phi: TC \hookrightarrow G$ .

11.  $Tensor(G, H, GactH, HactG, NiceGeneratorsG, NiceGeneratorsH, p) = (GtensorH, delta, TC, TChomG, phi, act)$   
 $FpGrp, FpGrp, Map, Map, Set, Set, RngIntElt \rightarrow EnumGrp, Map, EnumGrp, Map, Map, Map$   
 =====

Given two finite fp-groups  $G, H$  with compatible actions (i.e., mappings  $GactH : G \times H \rightarrow H$  and  $HactG : H \times G \rightarrow G$  and “nice” generating sets (for instance, those which are closed under conjugation and under the given actions, or those given as the output from Functions 4 or 5), this function yields:

- i) an enumerated version  $GtensorH$  of the tensor product  $G \otimes H$ ;
- ii) the homomorphism  $delta: GtensorH \rightarrow G$  which sends a tensor to the corresponding “commutator” in  $G$ ;
- iii) a subgroup  $TC$  of (an enumerated version of)  $G$  consisting of those elements  $g$  for which  $g \otimes h$  is trivial in  $G \otimes H$  for all  $h$  in  $H$ ;
- iv) the inclusion homomorphism  $TChomG: TC \rightarrow G$ ;
- v) the mapping  $phi: G \times H \rightarrow GtensorH$  from the cartesian product into the tensor product which sends a pair to the corresponding tensor;
- vi) the  $G$ -action  $act: G \times GtensorH \rightarrow (GtensorH)$  from the cartesian product to the tensor product.

12.  $ExteriorSquare(G, p) = (GexteriorG, delta)$   
 $FpGrp, RngIntElt \rightarrow EnumGrp, Map$   
 =====

This function is defined for any finite fp-group  $G$ . It yields an enumerated version  $GexteriorG$  of the exterior square  $G \wedge G$ . It also yields the homomorphism

$$delta: GexteriorG \rightarrow G$$

which sends an exterior form to the corresponding commutator in  $G$ .

13.  $Htwo(G, p) = HG$   
 $FpGrp, RngIntElt \rightarrow EnumGrp$   
 =====

This function is defined for any finite fp-group  $G$ . It yields the kernel  $HG$  of the homomorphism  $G \wedge G \rightarrow G$  which sends each exterior form to the corresponding commutator. The group  $HG$  is returned as an enumerated group. (The group  $HG$  is isomorphic to the Schur multiplier of  $G$ , cf. Theorem 4.)

14. `RelativeSchurMultiplier(G,N,p) = M(G,N)`  
`FpGrp, SubFpGrp, RngIntElt → EnumGrp`  
 =====

This function is defined for any finite fp-group  $G$  with normal subgroup  $N$ . It yields the relative Schur multiplier  $M(G, N)$  as an enumerated group.

15. `RelativeEpiCentre(G,N,p) = C`  
`FpGrp, SubFpGrp, RngIntElt → EnumGrp`  
 =====

This function is defined for any finite fp-group  $G$  with normal subgroup  $N$ . It yields the relative epicentre  $C$  as a subgroup of an enumerated version of  $N$ .

16. `ThirdHomologyModP(G,p) = d`  
`FpGrp, RngIntElt → RngIntElt`  
 =====

Given a finitely presented finite  $p$ -group  $G$ , this function returns the dimension  $d$  of the third homology group  $H_3(G, \mathbf{Z}/p\mathbf{Z})$  of  $G$  with coefficients in the integers modulo  $p$ . This homology group is a vector space. In practice, the function works only for groups  $G$  of extremely low order. (The function is based on Theorem 6.)

17. `Exterior(GhomQ, HhomQ, p) =`  
`(GexteriorH, delta, C, ChomG, phi, act)`  
`Map, Map, RngIntElt → EnumGrp,`  
`Map, EnumGrp, Map, Map, Map`  
 =====

Given three finite fp-groups  $G, H, Q$  and normal inclusions  $GhomQ: G \hookrightarrow Q, HhomQ: H \hookrightarrow Q$ , this function yields:

- i) an enumerated version `GexteriorH` of the exterior square  $G \wedge H$ ;
- ii) the homomorphism `delta: GexteriorH → G` which sends an exterior form to the corresponding ‘commutator’ in  $G$ ;
- iii) a subgroup `C` of (an enumerated version of)  $G$  consisting of those elements  $g$  for which  $g \wedge h$  is trivial in  $G \wedge H$  for all  $h \in H$ ;
- iv) the inclusion homomorphism `ChomG: C → G`;
- v) the mapping `phi: G × H → GexteriorH` from the cartesian product into the exterior product which sends a pair to the corresponding exterior form;
- vi) the  $G$ -action `act: G × GexteriorH → GexteriorH` from the cartesian product to the exterior product. The groups  $G, H, Q$  can be accessed by the commands `G:=Domain(GhomQ); H:=Domain(HhomQ); Q:=Codomain(GhomQ)`.

18. `BaerInvariant(G, c, q, p) = (BG, C)`  
`FpGrp, RngIntElt, RngIntElt, RngIntElt → EnumGrp, EnumGrp`  
 =====

Given a finite fp-group  $G$ , a positive integer  $c$ , and  $q$  a positive multiple of the exponent of the  $c$ -th Baer invariant  $BG$  of  $G$ , this function returns  $BG$  as an enumerated group. The function also returns the  $c$ th term of the upper epicentral series of  $G$  as an enumerated group  $C$ . (Theorem 7 and Proposition 8 can help with the choice of a suitable value for  $q$ .)

19. `CoveringGroup(G, p) = (DG, delta, Htwo)`  
`FpGrp, RngIntElt → FpGrp, Map, EnumGrp`  
 =====

Given a finite fp-group  $G$ , this function returns a finite fp-group  $DG$  which is a central extension  $DG \rightarrow G$  of  $G$  with the property that the Schur multiplier of  $G$  is isomorphic to

$$[DG, DG] \cap \ker(DG \rightarrow G).$$

It also returns the homomorphism  $\text{delta}: eDG \rightarrow eG$  given on enumerated groups, as well as the Schur multiplier  $\text{Htwo}$  as an enumerated group. (The fp-groups  $G$  and  $DG$  have the same generating set, but the order of  $DG$  is not minimal. The groups  $eDG$  and  $eG$  can be accessed by the commands `eDG:=Domain(delta);` `eG:=Codomain(delta);`)

20. `IsCapable(G,p) = B`  
`FpGrp, RngIntElt → Boolean`  
 =====

Given a finite fp-group  $G$  this function returns the boolean value `TRUE` if  $G$  is the inner automorphism group of some group, and returns the boolean value `FALSE` otherwise. (This function is based on Proposition 10.)

21. `IsTerminal(G,p) = B`  
`FpGrp, RngIntElt → Boolean`  
 =====

Given a finitely presented finite nilpotent group  $G$  this function returns the boolean value `TRUE` if  $G$  is a terminal group, and returns the boolean value `FALSE` otherwise. (This function is based on [15].)

**Example 1.** To find the order and nilpotency class of the tensor square of the Burnside group  $G = B(3, 3)$  of exponent 3 on three generators, enter the following:

```
F:=FreeGroup(3);
a:=F.1; b:=F.2; c:=F.3;
x1:=(c,a);
x1:=(a,b);
x2:=(a,c);
x3:=(b,c);
x4:=(x1,c);
rels:=[ a^3, b^3, c^3, x1^3, x2^3, x3^3, x4^3,
(b,a)*x1^-2, (c,a)*x2^-2, (x1,a), (x2,a), (x3,a)*x4^-1,
(x4,a), (c,b)*x3^-2, (x1,b), (x2,b)*x4^-2, (x3,b),
(x4,b), (x1,c)*x4^-1, (x2,c), (x3,c), (x4,c),
(x2,x1), (x3,x1), (x4,x1), (x3,x2), (x4,x2), (x4,x3) ];
G:=quo<F|rels>;
T:=TensorSquare(G,3);
print Order(T);
print NilpotencyClass(T);
```

This tensor square turns out to have order  $3^{20}$  and nilpotency class 2.

Now let  $N$  denote the centre of the Burnside group  $G$ . In order to determine the relative Schur multiplier  $M(G, N)$ , enter the following:

```
eG, GhomeG, eGhomG:=EnumeratedGroup(G,3);  
N:=Center(eG);  
N:=eGhomG(N);  
M:=RelativeSchurMultiplier(G,N,3);  
print M;
```

It turns out that the relative multiplier  $M(G, N)$  is elementary abelian of order 27.

#### Appendix A. MAGMA programs

The MAGMA programs discussed in this paper are provided as text files, forming a special electronic appendix to this paper (see Section 4 for installation instructions). This appendix is available to journal subscribers at:

<http://www.lms.ac.uk/jcm/1/lms97004/appendix-a/>

#### References

1. M. BACON, L.C. KAPPE and R.F. MORSE, 'On the nonabelian tensor square of a 2-Engel group', *Archiv der Mathematik* (Basel) 69 (1997) 353–364. 25, 34
2. F.R. BEYL and U. TAPPE, *Group extensions, representations, and the Schur multiplier*, Lecture Notes in Mathematics 958 (Springer, Berlin, 1982). 29, 29
3. W. BOSMA and J.J. CANNON, *Handbook of MAGMA functions*, Third edition (University of Sydney, 1994). 25, 32
4. W. BOSMA, J.J. CANNON and C.E. PLAYOUST, 'The MAGMA algebra system I: the user language', *J. Symbolic Computation*, 24, 3/4 (1997) 235–265. 25
5. R. BROWN, D.L. JOHNSON and E.F. ROBERTSON, 'Some computations of nonabelian tensor products of groups', *J. Algebra* 111 (1987) 177–202. 25, 29, 33
6. R. BROWN and J.-L. LODAY, 'Van Kampen theorems for diagrams of spaces', *Topology* 26 (1987) 311–335. 26, 27, 27, 27, 27, 28, 28, 31
7. J. BURNS and G. ELLIS, 'On the nilpotent multipliers of a group', *Mathematische Zeitschrift* 226 (1997) 405–428. 26, 26, 29, 29, 29, 31, 32, 32, 33, 35
8. J. BURNS and G. ELLIS, 'Inequalities for Baer invariants of finite groups', *Canadian Mathematical Bulletin*, to appear. 29, 29, 29
9. J.J. CANNON and C.E. PLAYOUST, 'MAGMA: A new computer algebra system', *Euro-math Bulletin*, 2, 1 (1996) 113–144. 25
10. R.K. DENNIS, 'In search of new homology functors having a close relationship to K-theory', preprint, Cornell University, 1976. 26
11. G. ELLIS, 'Tensor products and  $q$ -crossed modules', *J. London Math. Soc.* (2) 51 (1995) 243–258. 27, 27, 31
12. G. ELLIS, 'Capability, homology, and central series of a pair of groups', *J. Algebra* 179 (1996) 31–46. 28
13. G. ELLIS, 'The Schur multiplier of a pair of groups', *Applied Categorical Structures*, to appear (1997). 25, 28
14. G. ELLIS and F. LEONARD, 'Computing Schur multipliers and tensor products of finite groups', *Proc. Royal Irish Academy* (95A) 2 (1995) 137–147. 30, 30



15. L. EVENS, 'Terminal  $p$ -groups', *Illinois J. Math.* 12 (1968) 682–699. 29, 39
16. N.D. GILBERT and P.J. HIGGINS, 'The nonabelian tensor product of groups and related constructions', *Glasgow Mathematical J.* 31 (1989) 17–29. 30, 30
17. D. GUIN, 'Cohomologie non abéliennes des groupes', *J. Pure Applied Algebra* 50 (1988) 109–138. 28, 28
18. P. HALL, 'The classification of prime-power groups', *J. Reine Angew. Math.* 182 (1940) 130–141. 29
19. P.J. HILTON and U. STAMMBACH, *A course in homological algebra*, Graduate Texts in Mathematics 4 (Springer, New York, 1971). 28
20. L.-C. KAPPE, 'The nonabelian tensor product of groups', *Groups in St. Andrews/Bath 1997*, LMS Lecture Notes, (ed. C. CAMPBELL, E.F. ROBERTSON *et al.*, Cambridge University Press, Cambridge, 1997). 25
21. G. KARPILOVSKY, *The Schur multiplier*, LMS monographs New Series 2, (Oxford University Press, 1987).
22. C.R. LEEDHAM-GREEN and S. MCKAY, 'Baer invariants, isologism, varietal laws and homology', *Acta Mathematica* 137 (1976) 99–150. 26
23. J.-L. LODAY, 'Cohomologie et groupe de Steinberg relatif', *J. Algebra* 54 (1978) 178–202. 28, 28
24. A.S.-T. LUE, 'The Ganea map for nilpotent groups', *J. London Math. Soc.* (2) 14 (1976) 309–312. 26
25. A. MCDERMOTT, 'Tensor products of groups and related theory', PhD Thesis, National University of Ireland Galway, May 1998. 30, 30, 33
26. C. MILLER, 'The second homology group of a group', *Proc. Amer. Math. Soc.* 3 (1952) 588–595. 26, 28, 28

Graham Ellis [graham.ellis@ucg.ie](mailto:graham.ellis@ucg.ie)

Department of Mathematics  
National University of Ireland  
Galway  
Ireland