Computing Resolutions Over Finite p-Groups

Johannes Grabmeier[1] and Larry A. Lambe[2]

[1] IBM Deutschland Informationssysteme GmbH, Postfach 103068, D-69020 Heidelberg, Germany, grabm@de.ibm.com
[2] Centre for Innovative Computation University of Wales, Bangor, Gwynedd, LL57 1UT U.K. l.lambe@bangor.ac.uk

Dedicated to Professor Adalbert Kerber at the occasion of his 60th birthday.

Abstract. A uniform and constructive approach for the computation of resolutions and for (co)homology computations for any finite p-group is detailed. The resolutions we construct ([32]) are, as vector spaces, as small as the minimal resolution of $\mathbb{F}_p$ over the elementary abelian p-group of the same order as the group under study. Our implementations are based on the development of sophisticated algebraic data structures. Applications to calculating functional cocycles are given and the possibility of constructing interesting codes using such methods is presented.

1 Introduction

In this paper, we present a uniform constructive approach to calculating relatively small resolutions over finite p-groups. The algorithm we use comes from [32, 8.1.8 and the penultimate paragraph of 9.4]. There has been a massive amount of work done on the structure of p-groups since the beginning of group theory. A good introduction is [22].

We combine mathematical and computer methods to construct the uniform resolutions in this paper. These resolutions are much smaller than the bar construction [34, Chapter IV, §5] (or Sect. 3.4), but in fact, actually use the bar construction in an essential way. As vector spaces over $\mathbb{F}_p$ (the field with p elements), the resolutions are as small as the minimal resolution of $\mathbb{F}_p$ over the elementary abelian p-group G_+ of the same order as the group G under study.

In low degrees, (e.g. less than or equal to 7 depending on the size of the group), these resolutions can be used to explicitly calculate not only homology and cohomology, but explicit cycle and (functional) cocycle representives of classes. This takes things a step further than one can go with, e.g. GAP or MAGMA where one can only get at a basis for first and second (co)homology. Having functional cocycles in hand allows one to examine interesting combinatorial properties. In

this way we mention briefly how certain codes arise from some explicit cocycles. Also see [9] and [10].

We note that, in general, in order to increase practicality further, one needs to devise "reduction strategies" along the lines of [27] to reduce the size of the resolution for a general group. Our moderate sized resolutions are a good starting point for these methods, but such reductions are not within the scope of the current paper and will be discussed elsewhere.

A wide variety of algebraic data structures were required for our implementations. We briefly recall the mathematical setting for all these algebraic data structures and we also discuss them from the viewpoint of implementation. In so doing, we realized that it is well worth and by no means trivial to design a GENERIC LANGUAGE (GL) for the description and natural implementations of sophisticated algebraic objects, algorithms and data structures. However, these fundamental considerations combining mathematical and computer science methods and techniques would be far beyond the scope of the current paper and hence will be developed and discussed elsewhere [5]. For now, we use the computer algebra system AXIOM [24] which consists of a language, compiler, interpreter and a user interface to accomplish our goals.

The algorithm we present in Sect. 5 is of a recursive nature and can be applied naturally to p-groups G given as a polynomial perturbation of the elementary abelian group G_+ (Sect. 2.4). If the group is not given this way, we use the theorems of Jennings and Birkhoff-Poincaré-Witt to construct an appropriate isomorphism $\Xi : \mathbb{F}_p G_+ \longrightarrow \mathbb{F}_p G$ as vector spaces. This is related to the mod-p lower central series of G.

2 Finite p-groups

In this section, we give an exposition of some well-known properties of finite p-groups needed to understand the algorithms presented for practical applications of the main theorems in Sect. 4.

2.1 The mod-p Lower Central Series

For each finite p-group G, an n-dimensional *mod-p restricted* Lie algebra $\mathrm{gr}_p G$ can be defined ([25]) using the *mod-p lower central* series $G = Z_1 \geq Z_2 \geq \ldots$. Here Z_i is defined by

$$Z_i = \langle (x_1, (x_2, (\ldots (x_{j-1}, x_j) \ldots) y^{p^k} | jp^k \geq i \rangle$$

or equivalently, for $i > 1$, $Z_i = (Z_{i-1}, Z_1)Z_j{}^p$ where j is the smallest integer greater than or equal to $\frac{i}{p}$.

As $(Z_i)_{i\geq 1}$ in (2.1) is a *p-filtration* of the group G, i.e. the commutator group (Z_i, Z_j) is contained in Z_{i+j} and $Z_i^p \subseteq Z_{pi}$, the groups Z_i/Z_{i+1} (of order say p^{d_i}) are abelian and elementary, and hence we are able to define the $\mathbb{F}_p$-vector space $\mathrm{gr}_p G = \oplus_{i\geq 1} Z_i/Z_{i+1}$ and recursively can choose elements $\{x_{i,k} | 1 \leq k \leq d_i\}$ in the group, use their remainder classes as an $\mathbb{F}_p$-basis and define the Lie algebra multiplication

$$[x_{i,k}, x_{j,l}] = \overline{(x_{i,k}, x_{j,l})}$$

by using the commutator $(x_{i,k}, x_{j,l})$ in the group and reducing modulo the next subgroup Z_{i+j+1} in the filtration. More precisely, let ρ_i denote the natural surjection $\rho_i : Z_i \longrightarrow Z_i/Z_{i+1}$. Then the Lie bracket for elements $\rho_i(x) \in Z_i/Z_{i+1}$ and $\rho_j(y) \in Z_j/Z_{j+1}$ is defined by

$$[\rho_i(x), \rho_j(y)] = \rho_{i+j}((x, y))$$

while the definition

$$\rho_i(x)^p = \rho_{ip}(x^p)$$

satisfies the identities given on pages 91−93 in [25] (also see [23]), hence it is a *p-restriction* and this indeed yields a *p-restricted* Lie algebra. Let $\epsilon : \mathbb{F}_p G \longrightarrow \mathbb{F}_p, \sum_g a_g g \mapsto \sum_g a_g$ (cf. 3.3) be the augmentation of the group algebra $\mathbb{F}_p G$ and $I = \ker(\epsilon)$ the augmentation ideal. Then note also the important relation used below (2.3):

$$Z_i = \{g \in G | (g-1) \in I^i\}$$

holds ([25]).

2.2 The Universal Enveloping Algebra of a *p*-Restricted Lie Algebra and the Theorem of Birkhoff-Poincaré-Witt

Let L be an ordinary Lie algebra over $\mathbb{F}_p$. One has the universal enveloping algebra $T(L)/J$, where $T(L)$ is the tensor algebra of the underlying vector space structure of L and J is the ideal generated by $\{x \otimes y - y \otimes x - [x, y] \,|\, x, y \in L\}$ (see e.g. [37]) – there is a universal enveloping algebra for p-restricted Lie algebras. One adds the additional relations $\underbrace{x \otimes \ldots \otimes x}_{p} - x^p$ for $x \in L$ to the ideal J. We denote this algebra by $\mathcal{V}(L)$ for a p-restricted Lie algebra L. Similar to the

ordinary case, if A is any associative algebra over $\mathbb{F}_p$, there is a functor $\mathcal{L}$ such that $\mathcal{L}(A)$ is a p-restricted Lie algebra. The underlying vector space structure is that of A, the bracket is $[x, y] = xy - yx$, and the restriction is given by the p-th power in A. Furthermore, the universal property for $\mathcal{V}(L)$ holds for any p-restricted Lie algebra, viz. any map $f : L \longrightarrow \mathcal{L}(A)$ extends to a unique algebra map $\mathcal{V}(f) : \mathcal{V}(L) \longrightarrow A$ (see [23]).

$\mathcal{V}(L)$ has a natural grading induced by the filtering by length, and hence we can form the associated graded algebra, i.e. $E_0(\mathcal{V}(L)) = \sum_{i\geq 0} V_i/V_{i-1}$, where V_i for $i \geq 0$ is the submodule consisting of all elements, which are images of elements in the tensor algebra of (total) length less or equal to i and $V_{-1} = 0$. This construction is important for computation in the universal enveloping algebra $\mathcal{V}(\mathrm{gr}_p G)$ of the p-restricted Lie algebra $\mathrm{gr}_p G$ as one can make use of the p-modular version of the Birkhoff-Poincaré-Witt theorem.

Theorem 1. *Let L be a p-restricted Lie algebra with basis $\{e_1, \ldots, e_n\}$. The associated graded algebra $E_0(\mathcal{V}(L))$ of the universal enveloping algebra $\mathcal{V}(L)$ given by the length filtration is isomorphic as a graded $\mathbb{F}_p$-algebra to the algebra $\mathbb{F}_p[e_1, \ldots, e_n]/({e_1}^p, \ldots, {e_n}^p)$ of truncated polynomials.*

The proof is given in [23].

2.3 The Theorem of Jennings

The theorem of Jennings, [25], is the final link between these constructions.

Theorem 2. *Let $(Z_i)_{i>0}$ be the $\mathrm{mod}-p$ lower central series of a finite p-group G and let $I = \ker(\epsilon)$ be the augmentation ideal in the group ring $\mathbb{F}_p G$. Let $E_0(\mathbb{F}_p G) = \sum_{i\geq 0} I^i/I^{i+1}$ be the associated graded algebra with respect to the filtration given by powers of the augmentation ideal I. Let ρ_i denote both the canonical surjections $Z_i \longrightarrow Z_i/Z_{i+1}$ and $I^i \longrightarrow I^i/I^{i+1}$. Then $\rho_i(x) \mapsto \rho_i(x-1)$ induces an homomorphism of p-restricted Lie algebras $\mathrm{gr}_p G \longrightarrow \mathcal{L}(E_0(\mathbb{F}_p G))$ and its extension to a map*

$$\mathcal{V}(\mathrm{gr}_p G) \longrightarrow E_0(\mathbb{F}_p G)$$

is a graded algebra isomorphism.

Note that from the construction of $x_{i,k}$ in 2.1 it is clear that for each element g of the group there is a unique sequence of exponents $0 \leq \epsilon_{i,j} < p$ such that $g = \prod_{i \geq 1} \prod_{k=1}^{d_i} x_{i,k}^{\epsilon_{i,k}}$, where the order of the multiplied elements is lexicographic w.r.to (i,k). The crucial step in Jennings' work and particularly important for implementations is his proof that

$$x^{\epsilon} = \prod_{i \geq 1} \prod_{k=1}^{d_i} (x_{i,k} - 1)^{\epsilon_{i,k}}$$

in $E_0(\mathbb{F}_p G)$ for $0 \leq \epsilon_{i,k} < p$ (same ordering) is an $\mathbb{F}_p$-Basis. Note that all these basis elements are homogeneous, where the degree is computed by $\deg(x^{\epsilon}) = \sum_{i \geq 1} \sum_{k=1}^{d_i} i\epsilon_{i,k}$, and all the basis elements of degree i form a basis of I^i/I^{i+1} ([25]).

This theorem gives a close connection between restricted Lie algebras and finite p-groups. It is an important relation. It was generalized by Quillen to all groups – not just p-groups, see [36].

2.4 Polynomial Group Laws

It is not hard to see that any group G of order p^n is isomorphic to a group of the form $(\mathbb{F}_p^n, \rho)$, where the group law $\rho : \mathbb{F}_p^n \times \mathbb{F}_p^n \longrightarrow \mathbb{F}_p^n$ is a polynomial function. That is, the i-th component function $\rho_i(a,b)$ is given by a polynomial in the coordinates $(a_1, \ldots, a_n, b_1, \ldots, b_n)$ with coefficients in $\mathbb{F}_p$. Furthermore, the ρ_i may be chosen to satisfy

$$\rho_i(a,b) = a_i + b_i + \mu_i(a_1, \ldots, a_{i-1}, b_1, \ldots, b_{i-1}) \tag{1}$$

where μ_i is zero if any argument is zero. It is clear that the identity element is the zero vector and that μ_i cannot have a constant term.

Letting $e_n = (0, \ldots, 0, 1)$, it is clear that e_n has order p, $e_n^j = (0, \ldots, 0, j)$ and all these elements are in the center of the group.

The proof is by induction. Clearly the result is true for an elementary abelian p-group since its operation is just $+$. Assume inductively that the result is true for all finite p-groups G of order p^n. Given a finite p-group $\widetilde{G}$ of order p^{n+1}, as is well-known, there is a non-trivial element in the center and it can be chosen to have order p. We therefore have a central extension

$$0 \longrightarrow (\mathbb{F}_p, +) \xrightarrow{\alpha} \widetilde{G} \xrightarrow{\beta} G \longrightarrow 1.$$

As is also well-known ([34]), a 2-cocycle μ arises by choosing a right inverse $u : G \longrightarrow \widetilde{G}$ for β with $u(1) = 1$ and taking $G \times G \xrightarrow{\mu} \mathbb{F}_p$ to be $\mu(a, b) = u(a)u(b)u(ab)^{-1}$.

Assuming inductively that G and the group law ρ of G are of the desired form and noting that the group $\widetilde{G}$ is isomorphic to the group given by $G \times \mathbb{F}_p$ with group law

$$\widetilde{\rho}((a, a_{n+1}), (b, b_{n+1})) = (\rho(a, b), a_{n+1} + b_{n+1} + \mu_{n+1}(a, b)),$$

the result follows.

A natural class of examples is given by the upper triangular $n \times n$ matrix groups over $\mathbb{F}_p$, where n is any positive integer.

$$UT_n(p) = \left\{ \begin{pmatrix} 1 & a_1 & a_n & \dots & a_m \\ 0 & 1 & a_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{2n-3} \\ 0 & 0 & \ddots & \ddots & a_{n-1} \\ 0 & 0 & 0 & \ddots & 1 \end{pmatrix} \mid a_{i,j} \in \mathbb{F}_p \right\}. \tag{2}$$

Clearly, $UT_n(p) \cong (\mathbb{F}_p^m, \rho)$ where $m = \binom{n}{2}$ and the group law (matrix multiplication) is a polynomial function of the required form. Note also that any finite p-group can be embedded in $UT_n(p)$ for some n (see [22]).

For the class of cyclic groups C_{2^n} of order 2^n it is also easy to write down a polynomial group law. Define $c_1 = 0$ and for any positive integer i, let $\rho_i = a_i + b_i + c_i$ where

$$c_i(a_1, \dots, a_{i-1}, b_1, \dots, b_{i-1}) = \sum_{\gamma=1}^{i-1} a_\gamma b_\gamma \prod_{\kappa=\gamma+1}^{i-1} (a_\kappa + b_\kappa). \tag{3}$$

Then for all positive integers n, $\rho = (\rho_1, \dots, \rho_n)$ is a polynomial group law and in fact, the group $(\mathbb{F}_2^n, \rho)$ determined by ρ is a cyclic group of order 2^n. Note that this group is generated by $(1, 0, \dots, 0)$. More generally, if e_i is the i^{th} standard basis element vector, then $e_i e_i = e_{i+1}$ and e_i generates a subgroup of order 2^{n+1-i}. Moreover,

$$(e_1)^j = (j_0, j_1, \dots, j_{n-2}, j_{n-1})$$

for $0 \leq j < 2^n$ and $j = \sum_{\nu=0}^{n-1} j_\nu 2^\nu$ is the representation of j as a binary number.

Note also that $c_0 = 0$ and $c_{i+1} = c_i(a_i + b_i) + a_i b_i$. It is easily seen, that the polynomial group law precisely describes the addition of binary numbers. Contrary to the usual we have to reverse the order of the digits in this situation. A proof consists in verifying that adding three natural numbers a_i, b_i and a carry c_i from the position before from $\{0, 1\}$ to get $(c_{i+1}, \rho_i)_2$ can be realized recursively by $\rho_i = a_i + b_i + c_i$ (sum bits of two half adders) and the carry bit $c_{i+1} = a_i b_i + (a_i + b_i)c_i = a_i b_i \vee (a_i + b_i)c_i$ as $a_i b_i$ and $(a_i + b_i)c_i$ are never both equal to 1. This shows that we indeed have the usual implementation of a full adder.

For another natural example, let i be a positive integer and define $\rho_i = a_i + b_i + c_{i-1,2} + \delta_i a_1$, where $c_{i-1,2}$ is the cyclic cocycle for the positions $2, \ldots, i$ and $\delta_i = \sum_{\kappa=1}^{i} p_\kappa$, where p_κ is recursively defined by

$$p_1 = 0,\ p_2 = 0,\ p_{\kappa+1} = (b_\kappa + a_\kappa)\delta_\kappa + c_{\kappa-1,2} + b_\kappa.$$

Then for all positive integers n, $\rho = (\rho_1, \ldots, \rho_n)$ is a polynomial group law and the group $(\mathbb{F}_2^n, \rho)$ is a dihedral group D_{2^n} of order 2^n. Its cyclic subgroup of order 2^{n-1} can be generated by $c = (0, 1, 0, \ldots, 0)$, while $x = (1, 0, 0, \ldots, 0)$ is a reflection.

This can be verified by direct computation for the products $xa = (1 + a_1, \overline{a_2}, \ldots, \overline{a_{n-2}})$, $ax = e_1 + a$ for $a \in D_{2^n}$ where, as above, $e_1 = (1, 0, \ldots, 0)$ and $(x(c^i))(x(c^i)) = 1$ as well relating the recursion of p and ρ to the computation of a^{-1}. Note that for $i = i_2 + i_3 2 + \ldots + i_n 2^{n-2}$ we denoted by $\overline{i_2} + \overline{i_3} 2 + \ldots + \overline{i_n} 2^{n-2}$ the element $-i$ modulo 2^{n-1}.

3 Some Homological Algebra

We recall some basic facts from homological algebra needed to understand the purpose of the algorithms presented in Sect. 5. Throughout this section, R will denote a commutative ring with unit.

3.1 Chain/Cochain Complexes

A *chain complex* over R is a sequence of R-modules and R-linear maps

$$\ldots \xrightarrow{d_{n+1}} X_n \xrightarrow{d_n} X_{n-1} \longrightarrow \ldots$$

such that for all n, $d_n d_{n+1} = 0$. Following the usual conventions, such a chain complex will be denoted by (X, d) or simply X when the context

is clear. The map d is called the *differential*. If it needs to be stressed, we will write the differential in X as d_X. Elements of X_n are said to have degree n and if $x \in X_n$, we write $|x|$ for its degree. Since d lowers the degree by one, we say it has degree -1 and we write $|d| = -1$.

The n^{th} homology module of X, denoted by $H_n(X)$ is, by definition, the quotient module $\ker(d_n)/\mathrm{im}(d_{n+1})$, the homology of X is $H_*(X) = \oplus_n H_n(X)$.

A *cochain complex* over R is a sequence of R-modules and R-linear maps

$$\ldots \xleftarrow{d_n} X_n \xleftarrow{d_{n-1}} X_{n-1} \longleftarrow \ldots$$

such that for all n, $d_n d_{n-1} = 0$. The n^{th} cohomology module of X, denoted by $H^n(X)$ is, by definition, the quotient module $\ker(d_n)/\mathrm{im}(d_{n-1})$. The cohomology of X is defined to be $H^*(X) = \oplus_n H^n(X)$.

Note that if X is chain complex, then the linear dual $X^* = \mathrm{Hom}_R(X, R)$ is a cochain complex in the obvious way.

3.1.1 Chain Maps and Homotopies A *chain map* $f : X \longrightarrow Y$ is a sequence of R-linear maps making the diagram

$$\begin{array}{ccc} X_n & \xrightarrow{f_n} & Y_n \\ {\scriptstyle d_n}\downarrow & & \downarrow{\scriptstyle d_n} \\ X_{n-1} & \xrightarrow[f_{n-1}]{} & Y_{n-1} \end{array}$$

commute. Its easy to see that this condition causes any chain map to induce an R-linear map on homology $H_*(f) : H_*(X) \longrightarrow H_*(Y)$ in the obvious way.

Note that the identity map on X which we will denote by 1_X is a chain map.

Two chain maps $f, g : X \longrightarrow Y$ are said to be *chain homotopic* (by ϕ) if there is an R-linear map $\phi_n : X_n \longrightarrow Y_{n+1}$ such that $\phi_{n-1} d_n + d_{n+1}\phi_n = f_n - g_n$ for all n. Following conventions, this condition is simply written $d\phi + \phi d = f - g$. The (degree $+1$) map ϕ is called a *chain homotopy* between f and g. Its easy to see that if f and g are chain homotopic, then they induce the same map in homology.

Note that these notions clearly have analogues for cochain complexes.

3.1.2 Strong Deformation Retracts Let X and Y be chain complexes, $\nabla : X \longrightarrow Y$, $f : Y \longrightarrow X$ be chain maps and let $\phi : Y \longrightarrow Y$ be a degree $+1$ R-linear map such that $f\nabla = 1_X$ and $d\phi + \phi d = 1_Y - \nabla f$. Thus, f and ∇ compose to the identity, but the composition the other way around is only chain homotopic to the identity. When these conditions hold, we say that this collection of data forms a *strong deformation retraction* (SDR) and we write

$$X \underset{f}{\overset{\nabla}{\rightleftarrows}} (Y, \phi). \tag{4}$$

Crucial to computations are the *side conditions* ([30])

$$\phi^2 = 0, \quad \phi\nabla = 0, \quad \text{and}, \quad f\phi = 0.$$

In fact, it can be shown with a bit of computation that these may always be assumed to hold: if the last two do not hold, replace ϕ by $\phi' = D(\phi)\phi D(\phi)$ where $D(\phi) = \phi d + d\phi$ and the last two conditions will now hold with respect to ϕ'. If the first condition does not hold for ϕ', replace it by $\phi'' = \phi' d \phi'$ and all three conditions will hold for the chain homotopy ϕ''.

3.2 The Perturbation Lemma

Given the SDR (4) and in addition a second differential d'_Y on Y, let $t = d'_Y - d_Y$. The map t is called the *initiator* ([1]). The *perturbation lemma*, [2], [12], [1] states that if we set $t_n = (t\phi)^{n-1}t, \quad n \geq 1$ and, for each n, define new maps on X:

$$\partial_n = d + f(t_1 + t_2 + \ldots + t_{n-1})\nabla \tag{5}$$
$$\nabla_n = \nabla + \phi(t_1 + t_2 + \ldots + t_{n-1})\nabla. \tag{6}$$

On Y:

$$f_n = f + f(t_1 + t_2 + \ldots + t_{n-1})\phi \tag{7}$$
$$\phi_n = \phi + \phi(t_1 + t_2 + \ldots + t_{n-1})\phi. \tag{8}$$

then in the limits (provided they exist), we have new SDR data

$$(X, \partial_\infty) \underset{f_\infty}{\overset{\nabla_\infty}{\rightleftarrows}} ((Y, d'_Y), \phi_\infty). \tag{9}$$

Note that the limits will certainly exist if $t\phi$ is nilpotent in each degree.

Examples are given in [2], [12], [28], [33], [31], [29], [30], [13], [14], [21], [19], [20], for example. In particular, this paper discusses an implementation of the algorithm given at the end of section (9.4) in [32].

3.3 Homology and Cohomology of Algebras

Let A be an algebra over R. For a left A-module M, a projective resolution of M over A is an exact sequence of projective A-modules

$$\ldots \xrightarrow{d_2} X_1 \xrightarrow{d_1} X_0 \xrightarrow{\epsilon} M \longrightarrow 0. \tag{10}$$

In particular, there is an associated chain complex X:

$$\ldots \xrightarrow{d_2} X_1 \xrightarrow{d_1} X_0 \longrightarrow 0$$

such that $H_i(X) = 0$ if $i > 0$ and $H_0(X) = M$. Note that we will often write (10) as $X \xrightarrow{\epsilon} M \longrightarrow 0$

If N is another A-module and X is a projective resolution, $X \otimes_A N$ is a chain complex whose homology is $\mathrm{Tor}^A(M, N)$ and the (co)homology of the chain complex $\mathrm{Hom}_A(M, N)$ is $\mathrm{Ext}_A(M, N)$. In the special case that A is *augmented* over R, i.e. comes equipped with a ring homomorphism $A \xrightarrow{\epsilon} R \longrightarrow 0$, we can make R a left A-module via the action $ar = \epsilon(a)r$, for $a \in A$, $r \in R$. In this special case, it is conventional to denote $\mathrm{Tor}^A(A, R)$ by $H_*(A)$ and call it the homology of A and in addition, denote $\mathrm{Ext}_A(A, R)$ by $H^*(A)$ and call it the cohomology of A.

3.3.1 Homology and Cohomology of Groups If G is any group, let RG be the group ring over R, i.e. the free R-module over R with basis G and multiplication given by the bilinear extension of the product in G to all of RG. This algebra is augmented over R by $\epsilon(\sum r_g g) = \sum r_g$. The homology and cohomology of RG (in the sense above) are simply denoted by $H_*(G)$ and $H^*(G)$ in this case. In this paper, we are concerned with the case that G is a finite p-group and $R = \mathbb{F}_p$.

3.4 The Standard Resolution (Bar Construction)

Let A be an R-Algebra with augmentation ϵ as above. The *bar construction* $B(A)$ ([34], [3]) is a particular A-free resolution of R. It is of

the form $B(A) = A \otimes_R \bar{B}(A)$ where

$$\bar{B}(A) = \sum_{n \geq 0} \otimes_R^n \bar{A},$$

and $\bar{A} = A/R$ (thinking of R as a submodule of A via the unit). Following convention, we write $a[a_1|\dots|a_n] = a \otimes a_1 \otimes \dots \otimes a_n$ for an element of $A \otimes_R \bar{B}(A)$ and we think of the a_i as coming from A with the convention that $[a_1|\dots|a_n] = 0$ if one of the a_i is in R. Also, the class of the identity element of A in $\bar{B}_0(A) = \bar{A}$ is denoted by $[]$. Elements of $\bar{B}_n(A) = \otimes_R^n \bar{A}$ are called *reduced elements*.

Define an R-linear map $B(A) \xrightarrow{s} B(A)$ by

$$s(a[a_1|\dots|a_n]) = [a|a_1|\dots|a_n]$$

and extend the augmentation map to all of $B(A)$ by taking $\epsilon(a[]) = \epsilon(a)$ and $\epsilon(a[a_1|\dots|a_n]) = 0$ if $n \geq 1$. Let $R \xrightarrow{\sigma} A$ denote the unit map. Now consider the equation

$$\partial s + s\partial = 1_{B(A)} \tag{11}$$

in ∂ for chain maps. Let $\partial_0(a[]) = []\epsilon(a)$, and $s_{-1} : R \longrightarrow B_0(A) = A$ be given by $s_{-1}(r) = [\,]\sigma(r)$. The formula (11) then inductively determines ∂_n for all $n \geq 1$. It is straightforward to derive the formula

$$\begin{aligned} \partial([a_1|\dots|a_n]) = a_1[a_2|\dots|a_n] + \sum_{j=1}^{n-1}(-1)^j[a_1|\dots|a_ja_{j+1}|\dots|a_n] \\ + (-1)^n[a_1|\dots|a_{n-1}]\epsilon(a_n) \qquad (12) \\ (13) \end{aligned}$$

(extend A-linearly over all of $B(A)$). In fact, it is not hard to prove that $\partial^2 = 0$ so that we have a free resolution of the A-module R (via augmentation). In fact, it is not hard to see from the definitions that we actually have an explicit SDR

$$R \underset{\epsilon}{\overset{s_{-1}}{\rightleftarrows}} (B(A), s). \tag{14}$$

3.4.1 Functional Cocycles Let G be a group and $A = RG$ be the group ring. The dual of the bar construction

$$C(G) = \mathrm{Hom}_A(B(A), A) \tag{15}$$

is a complex whose cohomology is the cohomology of G (as defined in section (3.3). It is not hard to see that

$$C^n(G) = \mathrm{Hom}_R(\bar{B}_n(A), R) \cong F^n \tag{16}$$

where $F^n = \{G^n \xrightarrow{f} R \mid f(g_1, \ldots, g_n) = 0, \text{ if } g_\iota \in R \text{ for some } \iota\}$. See [34] for details. We shall identify $C^n(G)$ with such functions from G^n to R. Note that the differential in this context is given as follows. If $f : G^n \longrightarrow R$, then $\delta(f) : G^{n+1} \longrightarrow R$ is the function

$$\begin{aligned}\delta(f)(g_1, \ldots, g_{n+1}) = \\ f(g_2, \ldots, g_{n+1}) + \sum (-1)^k f(g_1, \ldots, g_k g_{k+1}, \ldots, g_{n+1}) \\ + (-1)^{n+1} f(g_1, \ldots, g_n)\end{aligned} \tag{17}$$

The algorithms given in the next sections can be used to explicitly compute such functional cochain representatives for the cohomology of any finite p-group. In fact, as we will see, the algorithm actually produces polynomials that represent cocycles.

3.5 The Comparison Theorem

The comparison theorem in homological algebra [34], [3] states that if X and Y are and two projective resolutions of M over A, they are chain homotopy equivalent, i.e. there are chain maps $f : X \longrightarrow Y$ and $g : Y \longrightarrow X$ such that fg and gf are both chain homotopy equivalent to the identity map. We will need a constructive version of this for free resolutions that essentially goes back to [4]. In fact, we are interested only in the case when $Y = B(A)$ and $fg = 1_X$, so that we actually obtain an SDR. The explicit formulae and discussion were given in [33]. We will simply repeat the formulas here for the reader's convenience.

Given a resolution of the form $X = A \otimes_R \bar{X} \longrightarrow R \longrightarrow 0$ with an explicit contracting homotopy $\psi : X \longrightarrow X$, construct maps $\nabla : X \longrightarrow B(A)$, $f : B(A) \longrightarrow X$, and $\phi : B(A) \longrightarrow B(A)$ inductively as follows:

$$\nabla_0 = 1_A, \quad f_0 = 1_A, \quad \phi_0 = 0, \tag{18}$$

and for $n > 0$, extend A-linearly the map defined on $\bar{X}$ given by

$$\nabla_n = s_{n-1}\nabla_{n-1}d_n \tag{19}$$

and extend A-linearly the maps defined on $\bar{B}(A)$ given by

$$f_n = \psi_{n-1}f_{n-1}\partial_n, \tag{20}$$

$$\phi_n = s_n(1_{\bar{B}_n(A)} - \nabla_n f_n - \phi_{n-1}\partial_n). \tag{21}$$

Conditions under which ∇ constructed this way is one-one are given in [35]. When they are satisfied, the maps above produce an SDR as is discussed in [33]. In this paper, it will be clear that the maps ∇ we define below are one-one by construction (and hence SDR data results).

The formulae for f and ϕ can easily be worked out ([33]) as follows. The A-linear map f is given recursively by

$$f([\,]) = 1,\ f([b_1|\ldots|b_n]) = \psi(b_1 f([b_2|\ldots|b_n])). \tag{22}$$

The A-linear map ϕ is given recursively by

$$\phi([\,]) = 0,\ \phi([b_1|\ldots|b_n]) = s\nabla f([b_1|\ldots|b_n]) + s(b_1\phi([b_2|\ldots|b_n])). \tag{23}$$

3.6 The Minimal Resolution of $\mathbb{F}_p^n$ Over $\mathbb{F}_p$

Let G_+ be the underlying abelian group of $\mathbb{F}_p$. H. Cartan [4] gave a resolution of $\mathbb{F}_p$ over $\mathbb{F}_p G_+^n$ and an explicit contracting homotopy which we will recall in this subsection. First, we will need to recall some standard algebras. The ordinary polynomial algebra is denoted by $\mathbb{F}_p[t_1, \ldots, t_n]$.

3.6.1 The Divided Power Algebra The divided power algebra $\Gamma_p[y]$, for an even degree generator y has $\mathbb{F}_p$-basis $\{\gamma_i(y)|i = 0, \ldots\}$. The multiplication is determined by extending

$$\gamma_i(y)\gamma_j(y) = \binom{i+j}{j}_p \gamma_{i+j}(y)$$

bilinearly over all of $\Gamma_p[y]$ where $\binom{i+j}{j}_p$ is the binomial coefficient mod p. Note that $\gamma_0(y) = 1$ and, by convention, we write $y = \gamma_1(y)$. For even degree generators, we define

$$\Gamma_p[y_1, \ldots, y_n] = \Gamma_p[y_1] \otimes \ldots \otimes \Gamma_p[y_n]$$

(tensor product algebra). We omit writing the tensor sign for elements when convenient. The degree is given by $|\gamma_{i_1}(y_1)\dots\gamma_{i_n}(y_n)| = \sum i_\nu|y_\nu|$.

3.6.2 The Exterior Algebra The exterior algebra $\Lambda_p[x]$, for an odd degree generator x, is the quotient $\mathbb{F}_p[x]/(x^2)$. For odd degree generators, $x_1,\dots,x_n$, we take

$$\Lambda_p[x_1,\dots,x_n] = \Lambda_p[x_1]\otimes\dots\otimes\Lambda_p[x_n].$$

We also omit writing tensor signs for elements when convenient. Note that every element of $\Lambda_p[x_1,\dots,x_n]$ can be written uniquely as an $\mathbb{F}_p$-linear combination of elements of the form $x_1^{i_1}\dots x_n^{i_n}$ where $i_\nu\in\{0,1\}$ and $|x_1^{i_1}\dots x_n^{i_n}| = \sum i_\nu|x_\nu|$.

3.6.3 Cartan's "Little Resolution" in Any Characteristic Note first of all that as an algebra, $\mathbb{F}_pG_+ \cong \mathbb{F}_p[t]/(t^p-1)$. We assign all elements the degree zero.

Let $\mathcal{C} = \mathbb{F}_pG_+\otimes\Gamma_p[y]\otimes\Lambda_p[x]$ (as an algebra) where $|y| = 2$ and $|x| = 1$. Again we omit tensor signs in writing elements when convenient. Extend the grading to all of $\mathcal{C}$ in the usual way, i.e. $|cc'| = |c|+|c'|$.

$\mathcal{C}$ is augmented by the map ϵ given by extending the assignments

$$\epsilon(t) = 1,\ \epsilon(x) = 0,\ \text{and } \epsilon(\gamma_i(y)) = 0$$

to an algebra map to $\mathbb{F}_p$. Note that the unit map is $\sigma(r) = r\otimes 1\otimes 1$.

The differential d on $\mathcal{C}$ is the unique $\mathbb{F}_pG_+$-linear graded derivation determined by

$$dt = 0,\quad dx = t-1,\quad d\gamma_i(y_1) = t^{[p]}\otimes\gamma_{i-1}(y)\otimes x \tag{24}$$

where we use the notation $t^{[k]} = \frac{t^k-1}{t-1}$ in general.

A contracting homotopy ψ for $\mathcal{C}$ is the $\mathbb{F}_p$-linear map given by

$$\begin{aligned}\psi(t^k\otimes\gamma_i(y)\otimes x^\eta) = {}& [k>0][\eta=0]\ t^{[k]}\otimes\gamma_i(y)\otimes x\\ &+ [k=p-1][\eta=1]\ 1\otimes\gamma_{i+1}(y)\otimes 1,\end{aligned} \tag{25}$$

where we use the Kronecker-Iverson notation $[b]$, which evaluates to 1 for Boolean expressions b having value true, and to 0 for those having value false, see [11].

Note that $\mathbb{F}_{\mathrm{p}}[t_1,\ldots,t_n]/(t_1^p-1,\ldots,t_n^p-1) \cong \mathbb{F}_{\mathrm{p}}[t_1](t_1^p-1)\otimes\ldots\otimes \mathbb{F}_{\mathrm{p}}[t_n]/(t_n^p-1) \cong \mathbb{F}_{\mathrm{p}}G_+^n$. Using tensor product formulae, we get a resolution $\mathcal{C}^{(n)}$ over $\mathbb{F}_{\mathrm{p}}G_+^n$ of the form

$$(\mathbb{F}_{\mathrm{p}}G_+^n \otimes \mathit{\Gamma}[y_1,\ldots,y_n]\otimes \mathit{\Lambda}_p[x_1,\ldots,x_n],\, d^{(n)}) \xrightarrow{\epsilon^{(n)}} \mathbb{F}_{\mathrm{p}} \longrightarrow 0$$

using the fact that

$$\mathbb{F}_{\mathrm{p}}G_+^n \otimes \mathit{\Gamma}_p[y_1,\ldots,y_n]\otimes \mathit{\Lambda}_p[x_1,\ldots,x_n] \cong \otimes^n\mathcal{C}.$$

Here $\epsilon^{(n)} = \otimes^n\epsilon$ and $d^{(n)}$ is the pullback of the tensor product differential

$$d^{\otimes^n} = \sum_{\nu=1}^{n} \underbrace{1\otimes\ldots\otimes 1}_{\nu-1}\otimes d\otimes \underbrace{1\otimes\ldots\otimes 1}_{n-\nu}. \tag{26}$$

and $\psi^{(n)}$ is the pullback of the tensor product homotopy given by

$$\psi^{\otimes^n} = \sum_{\nu=1}^{n} \underbrace{\pi\otimes\ldots\otimes \pi}_{\nu-1}\otimes \psi\otimes \underbrace{1\otimes\ldots\otimes 1}_{n-\nu}, \tag{27}$$

or by the reverse variant

$$\psi^{\otimes^n} = \sum_{\nu=1}^{n} \underbrace{1\otimes\ldots\otimes 1}_{\nu-1}\otimes \psi\otimes \underbrace{\pi\otimes\ldots\otimes \pi}_{n-\nu} \tag{28}$$

(or in fact, by any of the other permutations possible) where $\pi = \sigma\epsilon : \mathcal{C} \longrightarrow \mathcal{C}$. See [33, 2.3.2].

We will give a compact formula for the contracting homotopy $\psi^{(n)}$. Note first, that after fixing the position ν of the occurrence of ψ, we get a number of conditions for the appearances of non-zero terms by considering the positions $1 \le \mu < \nu$ where π occurs: $\pi(t_\mu^{a_\mu}\gamma_{i_\mu}(y_\mu)x_\mu^{\eta_\mu}) = [i_\mu = 0][\eta_\mu = 0]$ and hence it *kills* the element from the group algebra. Then we have to impose the conditions of ψ from (25) at position ν which can be associated to two sums. Hence, we immediately get

$$\psi^{(n)}(t^a\gamma_i(y)x^\eta) = \mathit{\Sigma}_0 + \mathit{\Sigma}_1 \tag{29}$$

where

$$\mathit{\Sigma}_0 = \sum_{\nu=1}^{n}\left(\prod_{\mu=1}^{\nu-1}[i_\mu=0][\eta_\mu=0]\right)[a_\nu>0][\eta_\nu=0]\sum_{\kappa=0}^{a_\nu-1} t^{\kappa e_\nu+\lambda_\nu(a)}\gamma_{\lambda_{\nu-1}(i)}(y)x^{e_\nu+\lambda_\nu(\eta)},$$

$$\mathit{\Sigma}_1 = \sum_{\nu=1}^{n}\left(\prod_{\mu=1}^{\nu-1}[i_\mu=0][\eta_\mu=0]\right)[a_\nu=p-1][\eta_\nu=1]\; t^{\lambda_\nu(a)}\gamma_{e_\nu+\lambda_{\nu-1}(i)}(y)x^{\lambda_\nu(\eta)}$$

and where we define

$$\lambda_\nu(a_1,\ldots,a_n) = (0,\ldots,0,a_{\nu+1},\ldots,a_n)$$

for $1 \leq \nu \leq n$ and we recall that e_ν is the ν^{th} standard basis vector in dimension n. Our programs use exactly these formulae and definitions we have just given.

Summing up we have a resolution $\mathcal{C}^{(n)}$ of $\mathbb{F}_{\mathrm{p}}$ over $\mathbb{F}_{\mathrm{p}}G_+^n$ and in fact, an SDR

$$\mathbb{F}_{\mathrm{p}} \underset{\epsilon}{\overset{\sigma}{\rightleftarrows}} (\mathcal{C}^{(n)}, \psi^{(n)}).$$

Note that, in fact, $\mathcal{C}^{(n)}$ is a *minimal* resolution, i.e. the differential in $\mathbb{F}_{\mathrm{p}} \otimes_{\mathbb{F}_{\mathrm{p}}G_+^n} \mathcal{C}^{(n)}$ vanishes. This is easy to see from the explicit form of the differential. Thus, in fact,

$$H_*(\mathbb{F}_{\mathrm{p}}G_+^n) \cong \mathbb{F}_{\mathrm{p}} \otimes_{\mathbb{F}_{\mathrm{p}}G_+^n} \mathcal{C}^{(n)} \cong \Gamma_p[y_1,\ldots,y_n] \otimes \Lambda_p[x_1,\ldots,x_n] \quad (30)$$

(as is well-known).

In the case of $p = 2$ this construction can be simplified.

3.6.4 Cartan's Little Resolution in Characteristic 2 In characteristic 2, we can define $\Gamma_2[x]$ for elements of any degree using the same multiplication formula as in the case of odd primes. Note then that if $|y| = 2$ and $|x| = 1$, we have

$$\Gamma_2[y] \otimes \Lambda_2[x] \cong \Gamma_2[x]. \quad (31)$$

An explicit algebra isomorphism is given by $\gamma_i(y) \otimes x^\eta \longmapsto \gamma_{2i+\eta}(x)$ as is easily verified. By tensoring, this generalizes to the n-variable case for $n \geq 1$. Thus, the underlying vector space for the minimal resolution of $\mathbb{F}_2$ over $\mathbb{F}_2(G_+)$ is $\mathcal{C} = \mathbb{F}_2(G_+) \otimes \Gamma_2[x]$ (G_+ is the underlying group structure of $\mathbb{F}_2$ in this case). The differential transfers over as

$$da = 0,\ d\gamma_i(x) = (t+1)\gamma_{i-1}(x) \quad (32)$$

for $a \in \mathbb{F}_2(G_+)$ and $\gamma_i(x)$ for $i \geq 1$. This follows from (24) and the fact that $t + 1 = t - 1 = t^{[2]}$.

As before, we obtain the resolution $\mathcal{C}^{(n)} = \mathbb{F}_2G_+^n \otimes \Gamma[x_1,\ldots,x_n]$ of $\mathbb{F}_2$ over $\mathbb{F}_2(G_+)[n]$ by tensoring $\mathcal{C}$ with itself n times. Thus, we have the usual tensor product formula for the differential and we have the

tensor product homotopy $\psi^{(n)}$ by formula (28). Note that in this case, we have $a_k = [a_k > 0]$. Since we will also have an explicit need of the formula in Sect. 6, we present it here.

$$\psi^{(n)}(t_1^{a_1} \dots t_n^{a_n} \gamma_{j_1}(x_1) \dots \gamma_{j_n}(x_n)) =$$

$$\sum_{k=1}^{n} \left(\prod_{p=k+1}^{n} [j_p = 0] \right) a_k t_1^{a_1} \dots t_{k-1}^{a_{k-1}} \gamma_{j_1}(x_1) \dots \gamma_{j_{k-1}}(x_{k-1}) \gamma_{j_k+1}(x_k). \tag{33}$$

3.6.5 Splitting Off of the Bar Construction Let $A = \mathbb{F}_p G_+^n$ in this section. The minimal resolutions given above split off of the bar construction in the terminology used in [33]. In other words, we have an SDR

$$\mathcal{C}^{(n)} \underset{f^{(n)}}{\overset{\nabla^{(n)}}{\rightleftarrows}} (B(A), \phi^{(n)}) \tag{34}$$

for each $n \geq 1$. The maps involved are given by the general formulae (18–23). But more can be said. Its not hard to see that $B(A)$ can be given an algebra structure $*$. The definition is inductive. In degree 0, $B_0(A)$ is just A and we take multiplication $*$ to be from this algebra. Inductively, on reduced elements $x, y \in \bar{B}(A)$, we take $x * y = s(\partial(x) * y + (-1)^{|x|} x * \partial(y))$. With this, $B(A)$ is a differential graded commutative algebra (this is true for any commutative algebra A). The recursive definition of $\nabla^{(n)}$ given by (19) makes it into a one-one map of differential graded algebras (see [4]). Thus, $\nabla^{(n)}$ is completely determined on the exterior algebra part of the minimal resolution by $\nabla^{(n)}(x_i) = [t_i - 1]$.

Still more can be said. Again, following Cartan in [4], one can define a *divided power* structure in $B(A)$ as follows. For p an odd prime and $x \in B(A)$, an *even degree reduced* element, define $\gamma_0(x) = x$ and inductively $\gamma_i(x) = s(\gamma_{i-1}(x) * \partial(x))$. With this definition, it is also not hard to see that the map $\nabla^{(n)}$ preserves divided powers, i.e. that $\nabla^{(n)}(\gamma_i(y_j)) = \gamma_i(\nabla^{(n)}(y_j))$.

When $p = 2$, the formula $\gamma_i(x) = s(\gamma_{i-1}(x) * \partial(x))$ is valid for x of any degree and it follows immediately that

$$A \otimes \Gamma_2[x_1, \dots, x_n] \xrightarrow{\nabla^{(n)}} B(A)$$

is the unique multiplicative extension of the map

$$\nabla^{(n)}(\gamma_i(x_j)) = [t_j | \dots | t_j] \ (i \text{ times}). \tag{35}$$

4 Main Theorems

Theorem 3. *Let G be a finite p-group and let A be its augmented group ring over $\mathbb{F}_p$. There is a free resolution*

$$(A \otimes \Gamma_p[y_1, \ldots, y_n] \otimes \Lambda_p[x_1, \ldots, x_n], d) \longrightarrow \mathbb{F}_p \longrightarrow 0 \tag{36}$$

of $\mathbb{F}_p$ over A. Furthermore, there is an SDR

$$(A \otimes \Gamma_p[y_1, \ldots, y_n] \otimes \Lambda_p[x_1, \ldots, x_n], d) \underset{f}{\overset{\nabla}{\rightleftarrows}} ((B(A), \partial), \phi). \tag{37}$$

If $p = 2$, this simplifies. In this case, there is a resolution

$$(A \otimes \Gamma_2[x_1, \ldots, x_n], d) \longrightarrow \mathbb{F}_2 \longrightarrow 0 \tag{38}$$

of $\mathbb{F}_2$ over A and there is an SDR

$$(A \otimes \Gamma_2[x_1, \ldots, x_n], d) \underset{f}{\overset{\nabla}{\rightleftarrows}} ((B(A), \partial), \phi) \tag{39}$$

These resolutions and strong deformation retracts can be constructed with algorithm 5.4.

Note that we do not claim that the differential is a derivation of the *obvious* algebra structure in either case.

One can iterate Wall's construction of twisted tensor product resolutions [38] to see why such resolutions exist, but that procedure requires making choices at various stages as explained below. In Sect. 7.1, we will prove however that up to conjugation by a chain isomorphism, any resolution obtained this way is given by our algorithm.

4.1 Twisted Tensor Product Resolutions

Given a group extension

$$1 \longrightarrow K \longrightarrow \widetilde{G} \longrightarrow G \longrightarrow 1$$

and two resolutions

$$\mathbb{F}_p K \otimes X \longrightarrow \mathbb{F}_p \longrightarrow 0, \ \mathbb{F}_p G \otimes Y \longrightarrow \mathbb{F}_p \longrightarrow 0,$$

the procedure in [38] constructs a differential on $\mathbb{F}_p\widetilde{G} \otimes X \otimes Y$, giving a resolution of $\mathbb{F}_p$ over $\mathbb{F}_p\widetilde{G}$, crucially using the fact that, as vector spaces, $\mathbb{F}_p\widetilde{G} \cong \mathbb{F}_p(K \times G)$. The construction is done in stages and several choices (which are abstractly known to be possible) have to be made. Now one can see how to get resolutions as above by induction. For $K = (\mathbb{F}_p, +)$, take the minimal resolution. If the theorem is true for all groups of order p^n and K' is of order p^{n+1}, then since K' is an extension of a group K of order p^n by $\mathbb{F}_p$, we can use the Wall construction on the resolution of the theorem over K and the minimal resolution over $\mathbb{F}_p$. Clearly the result is isomorphic to a resolution of the form of the theorem for K'.

Since iterating the construction above (essentially up through the lower central series) compounds the number of choices that have to by made considerably, it would be quite nice if there were a uniform procedure, suitable for programming, that could be given. In fact, the algorithm we mentioned in the introduction from [32] and which we implement in this paper does exactly that. We will discuss the relationship with the iterated twisted tensor product resolution in Sect. 7.1.

Finally, we mention that using the methods of this paper a stronger theorem actually holds. We have

Theorem 4. *Let G be a finite p-group and let A be its augmented group ring over $\mathbb{F}_p$. Let M be an A-module. There is a free resolution*

$$(A \otimes \Gamma_p[y_1, \ldots, y_n] \otimes \Lambda_p[x_1, \ldots, x_n] \otimes M, d) \longrightarrow \mathbb{F}_p \longrightarrow 0. \quad (40)$$

of M over A. Furthermore, there is an SDR

$$(A \otimes \Gamma_p[y_1, \ldots, y_n] \otimes \Lambda_p[x_1, \ldots, x_n] \otimes M, d) \underset{f}{\overset{\nabla}{\rightleftarrows}} ((B(A, M), \partial), \phi).$$

Note that this uses the two-sided bar construction [32, §3]. There is, of course, the obvious simplification for $p = 2$. We will however, only discuss the details of the version for $M = \mathbb{F}_p$ in this paper.

4.2 Complexes for Cohomology

By taking the A-linear dual of the onto map f from the theorems, we obtain an embedding of the linear dual over A of the small resolution into the dual $C(G)$ of the bar construction $B(A)$ (3.4.1).

Now note that for any A-module X, we have

$$X^* = \mathrm{Hom}_A(A \otimes X, \mathbb{F}_p) \cong \mathrm{Hom}_{\mathbb{F}_p}(X, \mathbb{F}_p).$$

Furthermore, it is well-known (e.g. [4]) that

$$\begin{aligned}&\mathrm{Hom}_{\mathbb{F}_p}(\Gamma_p[y_1, \ldots, y_n] \otimes \Lambda_p[x_1, \ldots, x_n], \mathbb{F}_p)\\ &\cong \mathbb{F}_p[w_1, \ldots, w_n] \otimes \Lambda_p[z_1, \ldots, z_n]\end{aligned}$$

and

$$\mathrm{Hom}_{\mathbb{F}_2}(\Gamma_2[x_1, \ldots, x_n], \mathbb{F}_2) \cong \mathbb{F}_2[z_1, \ldots, z_n].$$

as algebras (in fact, as Hopf-algebras), where $\{w_1, \ldots, w_n\}$ is dual to $\{y_1, \ldots, y_n\}$ and $\{z_1, \ldots, z_n\}$ is dual to $\{x_1, \ldots, x_n\}$. We do not claim that the corresponding differential δ_∞ is a derivation. In fact, generally this is false. There is however an algebra structure for which δ_∞ *is* a derivation. The discussion of this and its consequences is beyond the scope of the current paper however. The interested reader should see [26], [14]. We have the immediate

Corollary 5. *Let G be a finite p-group. If $p = 2$, let $X^* = \mathbb{F}_2[z_1, \ldots, z_n]$ otherwise let $X^* = \mathbb{F}_p[z_1, \ldots, z_n] \otimes \Lambda_p[w_1, \ldots, w_n]$. There is a differential δ_∞ on X and an embedding $X^* \hookrightarrow C(G)$ which is a chain homotopy equivalence. Hence $H^*(G) \cong H^*(X^*, \delta_\infty)$.*

As a consequence, given explicit cocycles in X^*, we can produce explicit functional cochains on the group G. Examples will be given in Sect. 8 below.

5 Explicit Algorithms

Let G be a finite p-group of order p^n and A be its augmented group ring over $\mathbb{F}_p$. We outline and detail the algorithms from [28], [33], [31], and [32] for constructing free resolutions. We then look specifically at the case of finite p-groups.

5.1 Perturbation Principle

The idea behind the algorithms is the following. Let A be an algebra over a field k and suppose there is free A resolution $\mathcal{B}(A, M) \xrightarrow{\epsilon} 0$ for all A-modules M where $\mathcal{B}(A, M) = A \otimes \bar{\mathcal{B}}(A, M)$ for some vector space

$\bar{\mathcal{B}}(A, M)$. For example one can take $\mathcal{B}$ to be the bar construction. We will say that an algebra A is a *perturbation* of an algebra A_0 if there is a k-linear isomorphism $A \cong A_0$. We suppose that if A is a perturbation of A_0, there is a vector space isomorphism $\mathcal{B}(A, M) \cong \mathcal{B}(A_0, M)$, as is the case with the bar construction. Thus, $\mathcal{B}(A_0, M)$ supports two differentials. The first is the differential d, corresponding to A_0 and the second, d' is the pullback to $\mathcal{B}(A_0, M)$ of the differential on $\mathcal{B}(A, M)$. Thus, if there is an SDR

$$(X, d_X) \underset{f}{\overset{\nabla}{\rightleftarrows}} (\mathcal{B}(A_0, M), \phi),$$

we can use $t = d' - d$ as an initiator and see if the perturbation formulae (3.2) converge. If they do, we obtain a resolution over A as small as the given one over A_0. There are various contexts in which a given algebra may be realized as a perturbation of another algebra and we give two examples below.

5.2 Polynomial Groups Laws Revisited

Suppose that G is given in the form of Sect. 2.4, i.e. we have $G = {\mathbb{F}_{\mathrm{p}}}^n$ (as sets) with groups law ρ that satisfies (1). In this case, we have generators $\{t_1, \ldots, t_n\}$ for G such that every element of G has a unique "normal form" $t_1^{a_1} \ldots t_n^{a_n}$ and the multiplication in G is given by

$$t_1^{a_1} \ldots t_n^{a_n} \cdot t_1^{b_1} \ldots t_n^{b_n} = t_1^{a_1+b_1+\mu_1} \ldots t_n^{a_n+b_n+\mu_n}$$

where $\mu_i = \mu_i(a_1, \ldots, a_{i-1}, b_1, \ldots, b_{i-1})$. On the same underlying set, we have the elementary abelian group law which we write as

$$t_1^{a_1} \ldots t_n^{a_n} + t_1^{b_1} \ldots t_n^{b_n} = t_1^{a_1+b_1} \ldots t_n^{a_n+b_n}.$$

As before write G_+^n for this group. We thus have two group laws on the same underlying set and so we have two different differentials (12) on $B(\mathbb{F}_{\mathrm{p}} G_+^n)$, viz., the differential ∂ for G and the differential ∂^+ for G_+^n. We set $\mathcal{T} = \partial - \partial^+$. Thus, using the notation $t^a = t_1^{a_1} \ldots, t_n^{a_n}$, we have an initiator (3.2)

$$\mathcal{T}[t^{a_1}|\ldots|t^{a_k}] = \sum_{i=1}^{k-1} ([t^{a_1}|\ldots|t^{a_i} \cdot t^{a_{i+1}}|\ldots|t^{a_k}] - [t^{a_1}|\ldots|t^{a_i} + t^{a_{i+1}}|\ldots|t^{a_k}]). \quad (41)$$

Provided the maps given in the perturbation lemma converge, we obtain a resolution of $\mathbb{F}_{\mathrm{p}}$ over A. If G is not presented in the form above, it is more complicated to obtain an initiator. We describe this next.

5.3 Using the Isomorphisms of Sect. 2 for an Initiator

In general, since we are over a field, we have that for any filtration of the algebra A, A is a perturbation of $E_0(A)$ in the sense defined above. Let G be a finite p-group and $A = \mathbb{F}_{\mathrm{p}}G$. By definition, we have $\mathbb{F}_{\mathrm{p}}G_+^n \cong \mathbb{F}_{\mathrm{p}}[t_1, \ldots, t_n]/(t_1^p - 1, \ldots, t_n^p - 1)$. But the latter algebra is clearly isomorphic to $\mathbb{F}_{\mathrm{p}}[t_1, \ldots, t_n]/(t_1^p, \ldots, t_n^p)$ since $(t-1)^p = t^p - 1 \bmod p$. We have a sequence of vector space-isomorphisms

$$E_0(\mathcal{V}(\mathrm{gr}_p G)) \cong \mathcal{V}(\mathrm{gr}_p G) \cong E_0(\mathbb{F}_{\mathrm{p}}G) \cong \mathbb{F}_{\mathrm{p}}G \tag{42}$$

using the observation above and the isomorphism of Theorem 2. But we also have the p-modular Birkhoff-Poincaré-Witt theorem 1 that gives an isomorphism

$$\mathbb{F}_{\mathrm{p}}[t_1, \ldots, t_n]/({t_1}^p, \ldots, {t_n}^p) \cong E_0(\mathcal{V}(\mathrm{gr}_p G)) \tag{43}$$

Putting all these maps together, we have an explicit realization of $A = \mathbb{F}_{\mathrm{p}}G$ as a perturbation of $\mathbb{F}_{\mathrm{p}}G_+^n$ by

$$\Xi : \mathbb{F}_{\mathrm{p}}G_+^n \longrightarrow \mathbb{F}_{\mathrm{p}}G \tag{44}$$

and we can form an intitator as described above. Again, provided the maps given in the perturbation lemma 3.2 converge, we obtain a resolution of $\mathbb{F}_{\mathrm{p}}$ over A.

We will give an indication of why the maps in the perturbation lemma converge in these cases only for the case $p = 2$. The general case is similar. Also note that the isomorphism just given is quite similar to the situation in Sect. 5.2. In essence, by refining the mod-p lower central series to have cyclic factors and pulling back the generators to G, we obtain a set of generators $\{t_1, \ldots, t_n\}$ for the group G so that every element has the unique (normal) form $t_1^{a_1} \ldots t_n^{a_n}$ and clearly, the normal form of the product of two such elements satisfies (1).

5.4 The Algorithm

Note that in what follows, improvements for the special case $p = 2$ can easily derived by using 3.6.4.

INPUT: – A group G of prime power order p^n given in some computationally accessible form and its group algebra $A = \mathbb{F}_{\mathrm{p}}G$ over the prime field $\mathbb{F}_{\mathrm{p}}$ with p elements.

– The elementary abelian group $G_+^n = (\mathbb{F}_p^n, +)$ of order p^n, and its group algebra $A_+ = \mathbb{F}_p G_+^n$.

step 1. Construct an $\mathbb{F}_p$-isomorphism $\Xi : A_+ \longrightarrow A$. If G is given as a perturbation of G_+^n by a polynomial group law ρ as in 2.4, then set Ξ to be the identity on the underlying sets, which are equal and by means of 5.2 proceed with step 2. Otherwise by means of 5.3 construct Ξ as by determining the p-modular lower central series

$$G = Z_1 \geq Z_2 \geq \ldots \geq Z_m \geq \ Z_{m+1} = 1,$$

according to 2.1, refining it to have cyclic factors, exhibit a generator of each factor and pull them back to get a sequence $t_1, \ldots, t_n$ of generators for G. Ξ then maps the generator corresponding to the unit vector e_i in $G_+^n \cong (\mathbb{F}_p^n, +)$ to t_i.

step 2. Construct the bar constructions $(B(A_+), \partial_+)$ and $(B(A), \partial)$ and according to 3.4, form the $\mathbb{F}_p$-vector space isomorphism

$$\Theta : (B(A_+), \partial_+) \longrightarrow (B(A), \partial)$$

induced by Ξ and transfer the differential ∂ by $\partial' := \Theta^{-1}\partial\Theta$ as a second differential on $B(A_+)$. Define the initiator $\mathcal{T} := \partial_+ - \partial' : B(A_+) \longrightarrow B(A_+)$ for the transfer process in the perturbation lemma 3.2. Furthermore, construct the SDR

$$\mathbb{F}_p \underset{\epsilon}{\overset{s_{-1}}{\rightleftarrows}} (B(A_+), s).$$

step 3. Construct Cartan's little resolution

$$\mathcal{C} = (A_+ \otimes \Gamma_p[y_1, \ldots, y_n] \otimes \Lambda_p[x_1, \ldots, x_n], d)$$

according to Sect. 3.6.3 for the elementary abelian group G_+^n by constructing the divided power algebra $\Gamma_p[y]$ with n generators (3.6.1) and the exterior power algebra $\Lambda_p(x)$ generated by n generators, see 3.6.2.

step 4. Form the SDR (see 3.6.5)

$$\mathcal{C}^{(n)} \underset{f^{(n)}}{\overset{\nabla^{(n)}}{\rightleftarrows}} (B(A_+), \phi^{(n)}).$$

step 5. Use (5)–(8) in 3.2 to recursively define maps

$$\partial_k^+,\ \nabla_k^{(n)},\ f_k^{(n)},\ \text{and}\ \phi_k^{(n)}$$

for $k = 1, \ldots$

OUTPUT: A resolution $(\mathcal{C}, \partial_\infty)$ of $\mathbb{F}_\mathrm{p}$ over the group algebra $\mathbb{F}_\mathrm{p}G$ and an SDR

$$(\mathcal{C}, \partial_\infty) \underset{f}{\overset{\nabla}{\rightleftarrows}} ((B(A_+), \partial), \phi)$$

as the limit of the constructions in step 4.

Note that the tensor products over $\mathbb{F}_\mathrm{p}$ can be realized algorithmically by designing a data type for divided power algebras and exterior algebras, which allow arbitrary commutative rings as coefficient domains.

6 Details of the Perturbation for p = 2

As before, let $G_+^n = \mathbb{F}_2^n$ be the elementary abelian group 2^n and let $A_+ = \mathbb{F}_2 G_+^n$. We want to examine the SDR of the bar construction and the minimal resolution in more detail. By (31–33), we may write this as

$$(A_+ \otimes \Gamma_2[x_1, \ldots, x_n], d) \underset{f}{\overset{\nabla}{\rightleftarrows}} ((B(A_+), \partial), \phi) \tag{45}$$

where

$$\nabla(t_1^{a_1} \ldots t_n^{a_n} \gamma_{j_1}(x_1) \ldots \gamma_{j_n}(x_n)) = t_1^{a_1} \ldots t_n^{a_n} \gamma_{j_1}([t_1]) * \ldots * \gamma_{j_n}([t_n])) \tag{46}$$

and f and ϕ are generally given by (22) and (23).

Let $\mathcal{T}$ be the initiator from the last section. We need to see that $\mathcal{T}\phi$ is nilpotent in each degree. We will indicate why this is true in degree one. The higher degrees are similar.

In degree one we have

$$f([t_1^{a_1} \ldots t_n^{a_n}]) = \sum_{i=1}^{n} a_i t_{i+1}^{a_{i+1}} \ldots t_{n+1}^{a_{n+1}} \gamma_1(x_i) \tag{47}$$

Note that for a uniform formula, we have taken $t_{n+1} = 1$ here. See (33) above. Using this formula for f and (23) we have

$$\phi([t_1^{a_1}| \ldots |t_n^{a_n}]) = \sum_{i=1}^{n-1} a_i [t_{i+1}^{a_{i+1}} \ldots t_n^{a_n} \mid t_i]. \tag{48}$$

Assume now that we have a finite 2-group G and let $A = \mathbb{F}_2 G$. We assume that G is of the form in Sect. 2.4. So the group law ρ is a polynomial function that satisfies (1). We want to examine the perturbation formulae more closely in this situation.

Thus by (41) and (48), we immediately have

$$\mathcal{T}\phi([t_1^{a_1} \dots t_n^{a_n}]) = \sum_{i=1}^{n-2} a_i([t_{i+1}^{a_{i+1}} \dots t_n^{a_n} \cdot t_i] + [t_i t_{i+1}^{a_{i+1}} \dots t_n^{a_n}]). \qquad (49)$$

Note that the last term in the sum vanished since t_n is central as follows from (1). Now note that in the terms above,

$$t_{i+1}^{a_{i+1}} \dots t_k^{a_n} \cdot t_i = t_i t_i^{\mu_i} \dots t_n^{\mu_n}$$

where $i \leq n-2$ which again follows from (1). Thus, every term in the sum above is of the form $t_i t_i^{b_{i+1}} \dots t_n^{b_n}$ for some $b_j \in \{0, 1\}$. We will say that such elements in G have *rank* i. Thus, it suffices to see that $\mathcal{T}\phi$ is nilpotent on elements of rank i for $i \leq n-2$. The proof is by downward induction.

If $z = t_{n-2} t_{n-1}^{b_{n-1}} t_n^{b_n}$ is any rank $n-2$ element, then we have

$$\mathcal{T}\phi([z]) = [t_{n-1}^{b_{n-1}} t_n^{b_n} \cdot t_{n-2}] + [t_{n-2} t_{n-1}^{b_{n-1}} t_n^{b_n}].$$

but $t_{n-1}^{b_{n-1}} t_n^{b_n} \cdot t_{n-2} = t_{n-2} t_{n-1}^{b_{n-1}} t_n^{b_n+q}$ where $q = q(b_{n-1})$ and q is a polynomial over $\mathbb{F}_2$ with no constant term. In fact, from (1), it is clear that

$$q(b_{n-1}) = \mu_n(0, \dots, 0, b_{n-1}, 0, \dots, 0, 1, 0).$$

We thus have $\mathcal{T}\phi(z) = [t_{n-2} t_{n-1}^{b_{n-1}} t_n^{b_n+q}] + [z]$ and we need to consider cases.

If $b_{n-1} = 0$, then clearly $q = 0$ and the terms above cancel. If $b_{n-1} = 1$, then if $q = 0$, the terms cancel once more. If $q = 1$ in this case, we have $z = t_{n-2} t_{n-1} t_n^{b_n}$ and $\mathcal{T}\phi([z]) = [w] + [z]$ where $w = t_{n-2} t_{n-1} t_n^{b_n+1}$, but then $(\mathcal{T}\phi)^2([z]) = [t_{n-2} t_{n-1} t_n^{b_n+1+1}] + [w] + [w] + [z] = 0$. The proof for the inductive step is similar.

7 Some Observations and Consequences

7.1 Relationship to Twisted Tensor Product Resolutions

Consider again the twisted tensor product construction of Sect. 4.1. We have

Theorem 6. *Let G be a finite p-group and let $(X, d) \longrightarrow \mathbb{F}_p \longrightarrow 0$ be the twisted tensor product resolution over G from Sect. 4.1. We have that (X, d) is chain isomorphic to the resolution given by the perturbation lemma using the algorithm from Sect. 5.*

Proof. We use the uniqueness theorem from [1]. We have set up an explicit *transference problem* (see [1] for terminology), in this case, in Sect. 5. Since the twisted tensor product differential is a solution to this problem, the main theorem of [1, §5] shows that we have a chain homotopy *isomorphism* so that the differential obtained by iterating the construction in [38] is conjugate to the differential obtained via the perturbation formulae.

Note that, of course, by the comparison theorem, all such resolutions must be chain homotopy equivalent. This theorem gives a much stronger statement. In essence, up to the choices made in [38], the construction is exactly the same as the one given uniformly by the perturbation lemma.

7.2 An Explicit SDR and an Explicit Contracting Homotopy

Using the algorithms from Sect. 5, we obtain an SDR

$$(\mathcal{C}^{(n)}, \partial_\infty) \underset{f_\infty^{(n)}}{\overset{\nabla_\infty^{(n)}}{\rightleftarrows}} ((B(A), \partial), \phi_\infty^{(n)}). \tag{50}$$

but more can be said.

It turns out that this construction actually provides an explicit contracting homotopy for $(\mathcal{C}^{(n)}, \partial_\infty)$. This can be seen from the general

Lemma 7. *Let X be a resolution of R over A and suppose that we have an SDR*

$$X \underset{f}{\overset{\nabla}{\rightleftarrows}} (B(A), \phi).$$

Then $fs\nabla$ is a contracting homotopy for X.

7.3 2-Cocyles and Codes

For any group G, the special case of the cochain differential (17) for $i = 2$ is of some interest. A function $f : G \times G \longrightarrow \mathbb{F}_p$ that satisfies $\delta(f) = 0$ is said to be a 2-cocycle. I.e. f is a 2-cocycle if and only if

$$f(y,z) - f(xy,z) + f(x,yz) - f(x,y) = 0 \tag{51}$$

for all $x, y, z \in G$. As was pointed out in section 2, every p-group is built up inductively from $\mathbb{F}_p$ by a sequence of 2-cocycles.

Using classical methods involving the universal coefficient theorem, Schur multipliers, and transgression, methods for finding 2-cocycles representing 2-dimensional cohomology classes can be worked out in some cases. See [7], [8] and [17]. Also see [6]. Connections between combinatorial design theory and 2-cocycles has been pointed out in [15] (also see [16] for errata). Connections between coding theory and 2-cocycles have also been made in [18].

Evaluating functional 2-cocycles on all pairs of the group elements yields a matrix with entries from $\mathbb{F}_2$ in case of $p = 2$. In light of the connection between cocycles and combinatorics just mentioned, it seems useful to have a means to generate whole families of functional cocycles to examine for various properties. Our algorithm above actually can calculate explicit representative cocycles (and hence codes) directly for any finite p-group. An example is given in 8.2.4.

7.4 Universal Cochain

Let G be a finite p-group for which the group law is given by a polynomial as in (1). We want to examine Cor. 5 in more detail. First of all, we have an onto map $\bar{f}_\infty : \bar{B}(\mathbb{F}_p G) \longrightarrow X$ where $X = \Gamma_p[y_1, \ldots, y_n] \otimes \Lambda_p[x_1, \ldots, x_n]$ in case p is odd and $X = \Gamma_2[x_1, \ldots, x_n]$ when $p = 2$ by "reducing" the SDR from Theorem 3, i.e., by tensoring the objects and maps over $\mathbb{F}_p G$ with $\mathbb{F}_p$. We claim the

Lemma 8.

$$\bar{f}_\infty[t^{a_1}|\ldots|t^{a_k}] = \sum_{i,\epsilon} \lambda_{i,\epsilon}\gamma_i(y)x^\epsilon \tag{52}$$

where $a_\kappa = (a_{1,\kappa}, \ldots, a_{n,\kappa})$, $t^{a_\kappa} = t_1^{a_{1,\kappa}} \ldots t_n^{a_{n,\kappa}}$, $i = (i_1, \ldots, i_n)$, $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$, $\gamma_i(y) = \gamma_{i_1}(y_1) \ldots \gamma_n(y_n)$, $x^\epsilon = x_1^{\epsilon_1} \ldots x_n^{\epsilon_n}$, *and*

$$\lambda_{i,\epsilon} = \lambda_{i,\epsilon}(a_1, \ldots, a_k) \tag{53}$$

is a polynomial in the coordinates $a_{\nu,\kappa}$. As such, we may think of the expression (52) as lying in $\mathbb{F}_p[a_{1,1},\ldots,a_{n,i}]/(a_{\nu,\kappa}^p - a_{\nu,\kappa}) \otimes X$.

The proof is quite easy once one realizes that f_∞ is constructed from the maps f and ϕ in the SDR (34) and the initiator $\mathcal{T}$ from Sect. 5. using only arithmetic operations. Its clear that f, $\mathcal{T}$, and ϕ can be expressed in terms of polynomials, so the lemma follows. We will denote this polynomial expression by $f_\infty^{(i)}$ and call it *the universal i-cochain*. As an immediate consequence of all this, we have

Corollary 9. *Let $\alpha = \sum_{j,\mu} \alpha_{j,\mu}\gamma_j(z)w^\mu$ be any i-cochain in $X^* = \mathbb{F}_p[z_1,\ldots,z_n]\otimes\Lambda_p[w_1,\ldots,w_n]$. The corresponding functional i-cochain*

$$f_\infty^{(i)}(\alpha) : G^i \to \mathbb{F}_p$$

is given by

$$f_\infty^{(i)}(\alpha)(t^{a_1},\ldots,t^{a_k}) = \sum \alpha_{i,\epsilon}\lambda_{i,\epsilon}(a_1,\ldots,a_k).$$

Obviously, if α is a cocycle, so is $f_\infty^{(i)}(\alpha)$, and in this way, we obtain polynomial representatives for cohomology classes.

Explicit examples of such universal cochains will be given in the next section.

8 Implementations and Computations

8.1 Implementations in AXIOM

The broad variety of algebraic data types required to realize and implement the the extensions of the algorithm from 5.4 seen in [32] in its most general setting is a topic of interest itself (see [5]). For the case of a given polynomial group law of the form (2.4) we have implemented the algorithm in the computer algebra system AXIOM (in arbitrary characteristic) and with focus on $p = 2$ and applications to coding theory. For the general case when the map Θ has to be considered, we also have written a program in the group theory system GAP as a preprocessing step when the finite p-group is given by power-commutator relations. The source code and some examples can be downloaded from the authors' homepages

`http://www.bangor.ac.uk/~mas019/`

or

`http://www.hd.shuttle.de/grabm/jg-top.html`.

8.1.1 Implementations for Arbitrary Characteristic The necessary structure for elementary abelian groups of order p^n, written multiplicatively, is given by the domain MWEA MultiplicativelyWrittenElementaryAbelian, and p-groups given by a polynomial group law are realized in PPGP PolynomialPGroup (see 2.4).[1] All kinds of graded, differential and augmented structures are provided, e.g. GRALALG GradedAugmentedLeftAlgebra. [2] The multiplicative structure of a monoid ring is given by the AXIOM domain constructor MRING MonoidRing, which was enhanced. [3] The bar construction (see 3.4) is implemented in the domain BAR BarConstruction [4]. Its basis over the group (monoid) ring has the infinite basis consisting of all lists of group elements.

We have constructed exterior algebras using the category EXTALGC ExteriorAlgebraCategory, and the domain EXTALG ExteriorAlgebra which requires the package EXMER ExMerge. [5] The algebra of divided powers is implemented in the domain DIVPOW DividedPowerAlgebra. [6] The full power and beauty of AXIOM is used to construct a domain for Cartan's little resolution (see 3.6) by combined use of the domain constructors ExteriorAlgebra MonoidRing, PrimeField, DividedPowerAlgebra and yields the constructor CLR CartanLittleResolution. [7] The functions (see 3.1.2) of the strong deformation retract for a p-group given by a polynomial perturbation of the elementary abelian group are given in the package SDRPG StrongDeformationRetractionPGroup. [8]

8.1.2 Implementations in Characteristic 2 In this case, the bar construction BarConstruction[9] is implemented using the constructor FreeModule[10] with coefficient ring R of category AssociativeAlgebra SingleIntegerMod 2 and with basis the type List ElementaryCommutativeGroup(2, n, R). The dimension n: SingleInteger is the first argument to the bar constructor, while R is either simply SingleIntegerMod 2 itself or an extension of it. This allows symbolic computation with generic

[1] The code is in `ppgp.spad`
[2] The code is in `graded.spad.`
[3] The code is in `mring.spad.`
[4] The code is in `bar.spad`
[5] The code is in `extalg.spad.`
[6] The code is in `divp.spad`
[7] The code is in `cartan.spad`
[8] The code is in `sdrpg.spad`
[9] This is coded in file `twococ2.as`.
[10] This is coded in file `freemod.as`.

group elements, e.g. SymbolicExponents[11]. The last argument to the bar constructor is the perturbation group law of type (Array R, Array R) ⟶ Array R, which describes the group of order 2^n as a perturbation of the elementary abelian group. The implemented functions include t2 and t3 which implement the initiator $\mathcal{T}$ (49) in degrees 2 and 3, both on the basis type and on the bar construction. Similarly the contracting homotopy ϕ_+ of the strong deformation retract of Cartan's little resolution in its characteristic 2 variant is realized in degrees 1 and 2 by phi1 and phi2. Finally, the perturbation lemma iteration is done for degree 1 and 2 in the code for t_phi_iteration1 and t_phi_iteration2. A function basisOfDegree returns lists of basis elements as elements of the bar construction.

The necessary functions for

$$\mathcal{C}^{(n)} \underset{f^{(n)}}{\overset{\nabla^{(n)}}{\rightleftarrows}} (B(A), \phi^{(n)})$$

for the given 2-group (see (3.6.5)) are implemented in StrongDeformationRetraction2Group[12], which has the same first 3 arguments as the implementation of the corresponding bar construction. Moreover, the fourth argument is the group algebra and has to be given as S: AssociativeAlgebra R[13]. The embedding $\nabla^{(n)}$ is implemented as nabla for degrees 1, 2, and 3 or for exactly one variable in arbitrary degree. The projection $f^{(n)}$ is implemented as f1 and f2 for the degrees 1 and 2 and on basis elements as well as linearly extended on arbitrary bar elements. The new differential is implemented as d and realized by 5 functions: dOnBasisOfDegree2 and dOnBasisOfDegree3 are constants which store the results on basis elements of degree 2 and 3, setTableForDegree sets up the constant hashtable (a data structure set up for efficient storage and retrevial) dTable for these degrees, while d extracts the values from the hashtable. The new projection is implemented as f and gained as the result of the transfer problem. It is implemented for degree 1 and 2 in f_infinity1 and f_infinity2, while the function lambda is an internal help function.

The package DualComputations[14] has the same arguments as StrongDeformationRetraction2Group. Here Cartan's Little Resolution C is im-

[11] This is coded in file `pgroups.as`.
[12] This is coded in file `twococ2.as`.
[13] This is coded in file `algebra.as`.
[14] This is coded in file `twococ2.as`.

plemented by DividedPowerBasisChar2(n, S, vx)[15]. The package exports the constant lists d1_dual and d2_dual with entries from C. They contain the values of the differential on the the (dual) basis of the cochain complex $\mathrm{Hom}_{\mathbb{F}_p}(\Gamma_2[x_1, ..., x_n], \mathbb{F}_p)$ (see 4.2 and 3.4.1). In addition the function d_dual implements its differential both for C or the type of its canonical basis CB, defined to be DividedPowerBasisChar2(n, S, vx)[16]. To be able to use linear algebra to determine the kernels and images, the constants matrix_d1_dual and matrix_d2_dual return objects of Matrix S[17], where each row consists of the coefficients of the differential applied to a dual basis element.

The package CocycleComputations[18] has two arguments, the dimension n and the polynomial group law rho: (ASE, ASE) ⟶ ASE, where ASE abbreviates Array SymbolicExponents n[19]. It conveniently puts all pieces of computations together and exports cocycleRepresentatives and secondCohomologyGroup. The second function takes the polynomial group law (ASE, ASE) ⟶ ASE as its argument and returns the a 2-tupel. The first component is the kernel of the differential, the second component is the image of the differential for degree 2. This can be conveniently used as argument for the function cocycleRepresentatives, which returns a basis consisting of 2-cocycle representatives as elements of the type SE.

8.2 Upper Triangular Matrices

We present results of executing the algorithm in Sect. 5 first for the matrix groups $UT_3(\mathbb{F}_2)$ and $UT_4(\mathbb{F}_2)$ and then the group $UT_3(\mathbb{F}_5)$.

Using spectral sequence methods, one could work out the ranks of the cohomology of the groups given in this section. Indeed, this provides a check of our results. In fact, even though we obtain a complex for computing the (co)homology of a group directly, it can be advantageous to use the spectral sequence associated to it when the number of generators is large and it is possible to interpret our results as giving a closed form expression for the corresponding differentials [32]. Since we obtain resolutions, we will however produce more than just cohomology. In addition, when we want to obtain results about (co)homology,

[15] This is coded in file `divpow.as`.
[16] This is coded in file `divpow.as`.
[17] This is coded in file `matrix.as`.
[18] This is coded in file `twococ2.as`.
[19] This is coded in file `pgroups.as`.

we can easily get (co)cycle representatives from our complexes using the universal cochain from Sect. 7.4. This is illustrated in the last example.

8.2.1 $\mathbf{UT_3(\mathbb{F}_2)}$ This group G is dihedral and the group law may be written

$$(a_1, a_2, a_3)(b_1, b_2, b_3) = (a_1 + b_1, a_2 + b_2, a_3 + b_3 + a_1 b_2).$$

Of course, it is straightforward to compute $H^*(G, \mathbb{F}_2)$ in this case using the Lyndon-Hochschild-Serre spectral sequence. The result is $H^*(G, \mathbb{F}_2) = R[z]$, the polynomial algebra in one variable z of degree 2 over the ring $R = \mathbb{F}_2[x, y]/(xy)$. Thus, H^2 has dimension 3.

It is also quite easy to compute $\bar{\partial}^*_{\infty,i}$ for $i = 1, 2$ in this case. We give only the non-zero differentials on the dual of the canonical basis:

$$x_3^* \longmapsto (x_1x_2)^* \tag{54}$$
$$(x_2x_3)^* \longmapsto (x_1\gamma_1(x_2))^* \tag{55}$$
$$(x_1x_3)^* \longmapsto (x_2\gamma_1(x_1))^* \tag{56}$$
$$(\gamma_2(x_3))^* \longmapsto (x_1\gamma_1(x_2))^* + (x_2\gamma_1(x_1))^*. \tag{57}$$

Thus, representatives of the three cohomology classes in dimension two are given by

$$\alpha_1 = (\gamma_2(x_1))^*, \quad \alpha_2 = (\gamma_2(x_2))^*, \quad \alpha_3 = (\gamma_2(x_3))^* + (x_1x_3)^* + (x_2x_3)^*.$$

The universal polynomial cochain is also easily computed. It is

$$\begin{aligned} f_\infty(a, b) = {} & (a_3b_3 + a_1a_3b_2 + a_1b_2b_3)\gamma_2(x_3) \\ & + a_2b_2\gamma_2(x_2) + a_1b_1\gamma_2(x_1) \\ & + a_1b_2x_1x_2 + a_1b_3x_1x_3 \end{aligned}$$

so three cocycle representatives $c_i : G \times G \longrightarrow \mathbb{F}_2$ are given by

$$\begin{aligned} c_1(a, b) &= a_1b_1 \\ c_2(a, b) &= a_2b_2 \\ c_3(a, b) &= a_3b_3 + a_1a_3b_2 + a_1b_2b_3 + a_1b_2 + a_1b_3. \end{aligned}$$

8.2.2 The universal Cochain for $\mathrm{UT}_4(\mathbb{F}_2)$ The interested reader can find further results for the example above as well as results for $\mathrm{UT}_4(\mathbb{F}_2)$ at the author's web sites (see Sect. 8.1), however we will list the universal cochain for $\mathrm{UT}_4(\mathbb{F}_2)$ in degree 2 here. It is

$$
\begin{aligned}
&a_1b_1\gamma_2(x_1) + a_1b_2\gamma_1(x_1)\gamma_1(x_2) + a_1b_3\gamma_1(x_1)\gamma_1(x_3) + a_1b_4\gamma_1(x_1)\gamma_1(x_4) + \\
&a_1b_5\gamma_1(x_1)\gamma_1(x_5) + a_1b_6\gamma_1(x_1)\gamma_1(x_6) + a_2b_2\gamma_2(x_2) + a_2b_3\gamma_1(x_2)\gamma_1(x_3) + \\
&(a_2b_4 + a_1a_2b_2)\gamma_1(x_2)\gamma_1(x_4) + a_2b_5\gamma_1(x_2)\gamma_1(x_5) + \\
&(a_2b_6 + a_1a_2b_5)\gamma_1(x_2)\gamma_1(x_6) + a_3b_3\gamma_2(x_3) + \\
&(a_3b_4 + a_1b_2b_3 + a_1a_3b_2)\gamma_1(x_3)\gamma_1(x_4) + (a_3b_5 + a_2a_3b_3)\gamma_1(x_3)\gamma_1(x_5) + \\
&(a_3b_6 + a_1a_3b_5)\gamma_1(x_3)\gamma_1(x_6) + (a_1b_2b_4 + a_4b_4 + a_1a_4b_2)\gamma_2(x_4) + \\
&(a_4b_5 + a_2b_3b_4 + a_2a_4b_3)\gamma_1(x_4)\gamma_1(x_5) + (a_4b_6 + a_1a_4b_5 + a_1b_2b_3b_4 + \\
&a_1a_3b_2b_4 + a_3a_4b_4 + a_1a_4b_2b_3 + a_1a_3a_4b_2)\gamma_1(x_4)\gamma_1(x_6) + (a_2b_3b_5 + a_5b_5 + a_2a_5b_3)\gamma_2(x_5) + \\
&(a_5b_6 + a_3b_4b_5 + a_1b_2b_3b_5 + a_1a_3b_2b_5 + a_1a_5b_5 + a_3a_5b_4 + a_1a_5b_2b_3 + a_1a_2a_3b_2b_3 + \\
&a_1a_2b_2b_3 + a_1a_3a_5b_2)\gamma_1(x_5)\gamma_1(x_6) + (a_1b_5b_6 + a_3b_4b_6 + a_1b_2b_3b_6 + a_1a_3b_2b_6 + \\
&a_6b_6 + a_3a_4b_6 + a_1a_3b_4b_5 + a_1b_2b_3b_5 + a_1a_3b_2b_5 + a_1a_6b_5 + a_1a_3a_4b_5 + a_1b_2b_3b_4 + \\
&a_1a_3b_2b_4 + a_3a_6b_4 + a_3a_4b_4 + a_1a_6b_2b_3 + a_1a_3a_4b_2b_3 + a_1a_3a_6b_2 + a_1a_3a_4b_2)\gamma_2(x_6)
\end{aligned}
$$

8.2.3 $\mathrm{UT}_3(\mathbb{F}_5)$ The AXIOM program given in App. A can be used to output the resolution in section 8.2.3. These procedures rely heavily upon our constructed libraries described in Sect. 8.1.1.

The Differential in the Resolution up to Degree 4 We will give the resolution only up to degree 4; only non-zero differentials are shown. In degree 1, we have

$$\partial_\infty(x_1) = 4 + t_1, \quad \partial_\infty(x_2) = 4 + t_2, \quad \partial_\infty(x_3) = 4 + t_3,$$

in degree 2,

$$
\begin{aligned}
\partial_\infty(x_1x_2) &= 4t_1t_2x_3 + (4 + t_1)x_2 + (1 + 4t_2)x_1, \\
\partial_\infty(x_1x_3) &= (4 + t_1)x_3 + (1 + 4t_3)x_1, \\
\partial_\infty(x_2x_3) &= (4 + t_2)x_3 + (1 + 4t_3)x_2,
\end{aligned}
$$

$$\partial_\infty(\gamma_1(y_1)) = {t_1}^{[5]}x_1, \quad \partial_\infty(\gamma_1(y_2)) = {t_2}^{[5]}x_2, \quad \partial_\infty(\gamma_1(y_3)) = {t_3}^{[5]}x_3,$$

in degree 3,

$$
\begin{aligned}
\partial_\infty(x_1x_2x_3) &= (4 + t_1)x_2x_3 + (1 + 4t_2)x_1x_3 + (4 + t_3)x_1x_2, \\
\partial_\infty(\gamma_1(y_1)x_1) &= (t_1 + 4)\gamma_1(y_1), \\
\partial_\infty(\gamma_1(y_1)x_2) &= (t_2 + 4)\gamma_1(y_1) + t_2\gamma_1(y_3) + p_1x_1x_3 + {t_1}^{[5]}x_1x_2, \\
\partial_\infty(\gamma_1(y_1)x_3) &= (t_3 + 4)\gamma_1(y_1) + {t_1}^{[5]}x_1x_3, \\
\partial_\infty(\gamma_1(y_2)x_1) &= (t_1 + 4)\gamma_1(y_2) + 4t_1\gamma_1(y_3) + p_2x_2x_3 + 4{t_2}^{[5]}x_1x_2,
\end{aligned}
$$

$$
\begin{aligned}
\partial_\infty(\gamma_1(y_2)x_2) &= (t_2+4)\gamma_1(y_2),\\
\partial_\infty(\gamma_1(y_2)x_3) &= (t_3+4)\gamma_1(y_2)+{t_2}^{[5]}x_2x_3,\\
\partial_\infty(\gamma_1(y_3)x_1) &= (t_1+4)\gamma_1(y_3)+4{t_3}^{[5]}x_1x_3,\\
\partial_\infty(\gamma_1(y_3)x_2) &= (t_2+4)\gamma_1(y_3)+4{t_3}^{[5]}x_2x_3,\\
\partial_\infty(\gamma_1(y_3)x_3) &= (t_3+4)\gamma_1(y_3),
\end{aligned}
$$

where

$$
\begin{aligned}
p_1 &= 4t_1t_2+4t_1^3t_2+4t_1^3t_2t_3+4t_1^3t_2t_3^2+4t_1^2t_2+4t_1^2t_2t_3+4t_1^4t_2t_3^3\\
&\quad +4t_1^4t_2t_3^2+4t_1^4t_2t_3+4t_1^4t_2\\
p_2 &= t_1t_2^3t_3^2+t_1t_2^3t_3+t_1t_2^3+t_1t_2^4t_3^3+t_1t_2^4t_3^2+t_1t_2^4t_3+t_1t_2^4\\
&\quad +t_1t_2^2t_3+t_1t_2^2+t_1t_2
\end{aligned}
$$

and in degree 4,

$$
\begin{aligned}
\partial_\infty(\gamma_1(y_1)x_1x_2) &= (4t_1t_2\gamma_1(y_1)+4t_1t_2\gamma_1(y_3))x_3+(t_1+4)\gamma_1(y_1)x_2\\
&\quad +((4t_2+1)\gamma_1(y_1)+4t_2\gamma_1(y_3))x_1\\
\partial_\infty(\gamma_1(y_2)x_1x_2) &= (4t_1t_2\gamma_1(y_2)+t_1t_2\gamma_1(y_3))x_3+((t_1+4)\gamma_1(y_2)+4t_1\gamma_1(y_3))x_2\\
&\quad +(4t_2+1)\gamma_1(y_2)x_1\\
\partial_\infty(\gamma_1(y_3)x_1x_2) &= 4t_1t_2\gamma_1(y_3)x_3+(t_1+4)\gamma_1(y_3)x_2+(4t_2+1)\gamma_1(y_3)x_1+t_3^{[5]}x_1x_2x_3\\
\partial_\infty(\gamma_1(y_1)x_1x_3) &= (t_1+4)\gamma_1(y_1)x_3+(4t_3+1)\gamma_1(y_1)x_1\\
\partial_\infty(\gamma_1(y_2)x_1x_3) &= ((t_1+4)\gamma_1(y_2)+4t_1\gamma_1(y_3))x_3+(4t_3+1)\gamma_1(y_2)x_1+4t_2^{[5]}x_1x_2x_3\\
\partial_\infty(\gamma_1(y_3)x_1x_3) &= (t_1+4)\gamma_1(y_3)x_3+(4t_3+1)\gamma_1(y_3)x_1\\
\partial_\infty(\gamma_1(y_1)x_2x_3) &= ((t_2+4)\gamma_1(y_1)+t_2\gamma_1(y_3))x_3+(4t_3+1)\gamma_1(y_1)x_2+t_1^{[5]}x_1x_2x_3\\
\partial_\infty(\gamma_1(y_2)x_2x_3) &= (t_2+4)\gamma_1(y_2)x_3+(4t_3+1)\gamma_1(y_2)x_2\\
\partial_\infty(\gamma_1(y_3)x_2x_3) &= (t_2+4)\gamma_1(y_3)x_3+(4t_3+1)\gamma_1(y_3)x_2\\
\partial_\infty(\gamma_2(y_1)) &= t_1^{[5]}\gamma_1(y_1)x_1\\
\partial_\infty(\gamma_1(y_1)\gamma_1(y_2)) &= q_1\gamma_1(y_3)x_3+(t_2^{[5]}\gamma_1(y_1)+q_2\gamma_1(y_3))x_2+(t_1^{[5]}\gamma_1(y_2)+q_3)\gamma_1(y_3)x_1\\
&\quad +q_4x_1x_2x_3\\
\partial_\infty(\gamma_1(y_1)\gamma_1(y_3)) &= t_3^{[5]}\gamma_1(y_1)x_3+t_1^{[5]}\gamma_1(y_3)x_1\\
\partial_\infty(\gamma_2(y_2)) &= t_2^{[5]}\gamma_1(y_2)x_2\\
\partial_\infty(\gamma_1(y_2)\gamma_1(y_3)) &= t_3^{[5]}\gamma_1(y_2)x_3+t_2^{[5]}\gamma_1(y_3)x_2\\
\partial_\infty(\gamma_2(y_3)) &= t_3^{[5]}\gamma_1(y_3)x_3,
\end{aligned}
$$

where

$$
\begin{aligned}
q_1 &= 4t_1^4+4t_1^4t_3+4t_1^4t_3^2+4t_1^4t_3^3+4t_1^3t_2^3+4t_1^3t_2^3t_3+4t_1^3t_2^3t_3^2+4t_1^2t_2^4+4t_1^2t_2^4t_3\\
&\quad +4t_1^3+4t_1^3t_3+4t_1^3t_3^2+4t_1^2t_3+4t_1^2+4t_1^4t_2^4t_3^3+4t_1^4t_2^4t_3^2+4t_1^4t_2^4t_3+4t_1^4t_2^4\\
q_2 &= t_1^3t_2^4+t_1^2t_2^4+t_1^4t_2^4+t_2^4+t_2^3+t_2+t_1^3t_2^2+t_1^4t_2^3+t_1^2t_2^3+t_2^2\\
q_3 &= 4t_1^4t_2^4+4t_1^3+4t_1^4t_2^2+4t_1^4+4t_1^3t_2^4+4t_1^3t_2^2+4t_1^2t_2^3+4t_1+4t_1^2+4t_1^4t_2^3\\
q_4 &= t_1^2t_2^4t_3^2+t_1^2t_2^4t_3+t_1^2t_2^4+t_1^2t_2t_3+t_1^2t_2+t_1^4t_2^2t_3^2+t_1^4t_2^2t_3+t_1^4t_2^2+t_1^4t_2^3t_3\\
&\quad +t_1^4t_2^3+t_1^2t_2^3+t_1^2t_2^2t_3^3+t_1^2t_2^2t_3^2+t_1^2t_2^2t_3+t_1^2t_2^2+t_1^4t_2t_3^3+t_1^4t_2t_3^2+t_1^4t_2t_3\\
&\quad +t_1^4t_2+t_1^4t_2^4+t_1t_2^4t_3^3+t_1t_2^4t_3^2+t_1t_2^4t_3+t_1t_2^4+t_1t_2+t_1^3t_2^2+t_1^3t_2^3+t_1^3t_2^3t_3\\
&\quad +t_1^3t_2^3t_3^2+t_1^3t_2^3t_3^3+t_1t_2^3+t_1t_2^3t_3+t_1t_2^3t_3^2+t_1t_2^2+t_1t_2^2t_3+t_1^3t_2+t_1^3t_2t_3+t_1^3t_2t_3^2\\
&\quad +t_1^3t_2^4+t_1^3t_2^4t_3.
\end{aligned}
$$

The Contracting Homotopy for the Resolution Only non-zero values are shown. Furthermore, we give the contracting homotopy only on elements that are needed to prove that the differential given above is indeed a resolution up to the given degree. This has been computed using Lemma 7.

$$\phi_\infty(t_1) = x_1, \quad \phi_\infty(t_2) = x_2, \quad \phi_\infty(t_3) = x_3$$

$$\begin{array}{ccc} \phi_\infty(t_2x_1) = 4x_1x_2, & \phi_\infty(t_3x_1) = 4x_1x_3, & \phi_\infty(t_3x_2) = 4x_2x_3, \\ \phi_\infty(t_1^4x_1) = \gamma_1(y_1), & \phi_\infty(t_2^4x_2) = \gamma_1(y_2), & \phi_\infty(t_3^4x_3) = \gamma_1(y_3) \end{array}$$

$$\begin{array}{c} \phi_\infty(t_3x_1x_2) = x_1x_2x_3, \\ \phi_\infty(t_1\gamma_1(y_1)) = \gamma_1(y_1)x_1,\ \phi_\infty(t_2\gamma_1(y_1)) = \gamma_1(y_1)x_2,\ \phi_\infty(t_3\gamma_1(y_1)) = \gamma_1(y_1)x_3, \\ \phi_\infty(t_2^4x_1x_2) = 4\gamma_1(y_2)x_1, \phi_\infty(t_2\gamma_1(y_2)) = \gamma_1(y_2)x_2,\ \phi_\infty(t_3\gamma_1(y_2)) = \gamma_1(y_2)x_3, \\ \phi_\infty(t_3^4x_1x_3) = 4\gamma_1(y_3)x_1,\ \phi_\infty(t_3^4x_2x_3) = 4\gamma_1(y_3)x_2, \phi_\infty(t_3\gamma_1(y_3)) = \gamma_1(y_3)x_3 \end{array}$$

$$\begin{array}{ll} \phi_\infty(t_2\gamma_1(y_1)x_1) = 4\gamma_1(y_1)x_1x_2, & \phi_\infty(t_2\gamma_1(y_2)x_1) = 4\gamma_1(y_2)x_1x_2, \\ \phi_\infty(t_3^4x_1x_2x_3) = \gamma_1(y_3)x_1x_2, & \phi_\infty(t_3\gamma_1(y_1)x_1) = 4\gamma_1(y_1)x_1x_3, \\ \phi_\infty(t_3\gamma_1(y_2)x_1) = 4\gamma_1(y_2)x_1x_3, & \phi_\infty(t_3\gamma_1(y_3)x_1) = 4\gamma_1(y_3)x_1x_3, \\ \phi_\infty(t_3\gamma_1(y_1)x_2) = 4\gamma_1(y_1)x_2x_3, & \phi_\infty(t_3\gamma_1(y_2)x_2) = 4\gamma_1(y_2)x_2x_3, \\ \phi_\infty(t_3\gamma_1(y_3)x_2) = 4\gamma_1(y_3)x_2x_3, & \phi_\infty(t_1^4\gamma_1(y_1)x_1) = \gamma_2(y_1), \\ \phi_\infty(t_2^4\gamma_1(y_1)x_2) = \gamma_1(y_1)\gamma_1(y_2), & \phi_\infty(t_3^4\gamma_1(y_1)x_3) = \gamma_1(y_1)\gamma_1(y_3), \\ \phi_\infty(t_2^4\gamma_1(y_2)x_2) = \gamma_2(y_2), & \phi_\infty(t_3^4\gamma_1(y_2)x_3) = \gamma_1(y_2)\gamma_1(y_3), \\ \phi_\infty(t_3^4\gamma_1(y_3)x_3) = \gamma_2(y_3). & \end{array}$$

The Differential in the Reduced Complex up to Degree 4 By definition, the reduced complex is the one obtained by tensoring the resolution with $\mathbb{F}_5$ over the group ring $\mathbb{F}_5\mathrm{UT}_3(\mathbb{F}_5)$. It is a suitable complex for computing the homology of $\mathrm{UT}_3(\mathbb{F}_5)$. Again, only non-zero differentials are shown.

$$\partial_\infty(x_1x_2) = 4x_3$$

$$\partial_\infty(\gamma_1(y_1)x_2) = \gamma_1(y_3) \quad \partial_\infty(\gamma_1(y_2)x_1) = 4\gamma_1(y_3)$$

$$\begin{aligned} \partial_\infty(\gamma_1(y_1)x_1x_2) &= (4\gamma_1(y_1) + 4\gamma_1(y_3))x_3 + 4\gamma_1(y_3)x_1 \\ \partial_\infty(\gamma_1(y_2)x_1x_2) &= (4\gamma_1(y_2) + \gamma_1(y_3))x_3 + 4\gamma_1(y_3)x_2 \\ \partial_\infty(\gamma_1(y_3)x_1x_2) &= 4\gamma_1(y_3)x_3 \\ \partial_\infty(\gamma_1(y_2)x_1x_3) &= 4\gamma_1(y_3)x_3 \\ \partial_\infty(\gamma_1(y_1)x_2x_3) &= \gamma_1(y_3)x_3 \\ \partial_\infty(\gamma_1(y_1)\gamma_1(y_2)) &= 2\gamma_1(y_3)x_3 \end{aligned}$$

8.2.4 A Cocyclic Code with a Hadamard Property Continuing the computation we find as representatives of a basis of the cohomology group $H^2(\mathrm{UT}_4(\mathbb{F}_2))$ the following 7 (abstract) cocycles.

$$\gamma_2(x_1)^*,\ \gamma_2(x_2)^*,\ \gamma_2(x_3)^*,\ (\gamma_1(x_1)\gamma_1(x_3))^*,\ \gamma_2(x_4)^* + (\gamma_1(x_1)\gamma_1(x_4))^* + (\gamma_1(x_2)\gamma_1(x_4))^*$$

$$\gamma_2(x_5)^* + (\gamma_1(x_3)\gamma_1(x_5))^* + (\gamma_1(x_2)\gamma_1(x_5))^*, (\gamma_1(x_2)\gamma_1(x_6))^* + (\gamma_1(x_3)\gamma_1(x_4))^* + (\gamma_1(x_4)\gamma_1(x_5))^*$$

Applying these functions to the universal cochain from Sect. 8.2.2 results in the following functional cocycle representatives of the second cohomology group:

$$\mu_1 = a_1b_1, \mu_2 = a_2b_2, \mu_3 = a_3b_3, \mu_4 = a_1b_3,$$
$$\mu_5 = a_1b_2b_4 + a_4b_4 + a_2b_4 + a_1b_4 + a_1a_4b_2 + a_1a_2b_2,$$
$$\mu_6 = a_2b_3b_5 + a_5b_5 + a_3b_5 + a_2b_5 + a_2a_5b_3 + a_2a_3b_3,$$
$$\mu_7 = a_2b_6 + a_4b_5 + a_1a_2b_5 + a_2b_3b_4 + a_3b_4 + a_1b_2b_3 + a_2a_4b_3 + a_1a_3b_2.$$

Considering the cocycle $\mu = \sum_{i=1}^{7} \mu_i$ and evaluating this function on all pairs of group elements results in a 0-1-matrix H which satisfies the following combinatorial property

$$HH^t = (64([i = j] + [i = 32j]))_{1 \le i,j \le 64}$$

– a generalization of the Hadamard property (cf. [18]). In particular the first 32 rows without column 1 determine a non-linear $(63, 32, 32)$ code.

8.3 The 2-Sylow Subgroup of $\mathrm{Sp}_4(F_{2^2})$

The symplectic group $\mathrm{Sp}_4(F_{2^2})$ is a sporadic simple group of order $979200 = 2^8 3^2 5^2 17$. The power-commutator presentation of its 2-Sylow subgroup was computed with the system GAP. [20] It is given by

$$\langle t_1, t_2, \ldots, t_8 | t_1^2 = t_8, (t_2, t_1) = t_5, (t_2, t_1) = t_6, (t_3, t_1) = t_8, (t_3, t_2) = t_7, (t_4, t_2) = t_5 \rangle.$$

Its mod-2 lower central series is $Z_1 = \mathrm{Sp}_4(F_{2^2}) > Z_2 = \langle t_5, t_6, t_7, t_8 \rangle > Z_3 = \{1\}$. This was refined to cyclic factors using GAP's functions `RightCoset` and `CanonicalRightCosetElement`.[21] This gave t_1t_8 and did not change $t_i, i \ge 2$. Writing t_1 for t_1t_8, which does not change the given relations, we have found the images under the vector space isomorphims Θ of the canonical generators $e_1, \ldots, e_n$ of the corresponding elementary abelian group of order 2^n. Using a *symbolic* collecting algorithm, which we have implemented[22], we can multiply two generic elements t^a and t^b to get the polynomial group law. We find that $t^a t^b = t^c$ where c is equal to

$$(b_1+a_1, b_2+a_2, b_3+a_3, b_4+a_4, b_5+a_4b_2+a_2b_1+a_5, b_6+a_3b_1+a_6, b_7+a_3b_2+a_7, b_8+a_4b_1+a_1b_1+a_8).$$

[20] `http://www-history-mcs.st-and.ac.uk/~gap/`
[21] See `sp4-4.gap and corrplie.gap.`
[22] The code is in `binstr.spad.`

For a group of nilpotency class 2 the power-commutator relations are directly reflected. The results for the reduced differential computed with our algorithm are as follows. In degree 1 $\bar{\partial}_\infty$ is 0. The non-zero images of the canonical basis in degree 2 are

$$\begin{aligned}
\bar{\partial}_\infty(x_1x_2) &= x_5,\\
\bar{\partial}_\infty(x_1x_3) &= x_6,\\
\bar{\partial}_\infty(x_1x_4) &= x_8,\\
\bar{\partial}_\infty(x_2x_3) &= x_7,\\
\bar{\partial}_\infty(x_2x_4) &= x_5,\\
\bar{\partial}_\infty(\gamma_1(y_1)) &= x_8,
\end{aligned}$$

while in degree 3 we have,

$$\begin{aligned}
\bar{\partial}_\infty(x_1x_2x_3) &= x_6x_7 + x_5x_7 + x_5x_6 + x_3x_5 + x_2x_6 + x_1x_7,\\
\bar{\partial}_\infty(x_1x_2x_4) &= x_4x_5 + x_2x_8 + x_1x_5,\\
\bar{\partial}_\infty(x_1x_2x_6) &= x_5x_6,\\
\bar{\partial}_\infty(x_1x_2x_7) &= x_5x_7,\\
\bar{\partial}_\infty(x_1x_2x_8) &= x_5x_8,\\
\bar{\partial}_\infty(x_1x_3x_4) &= x_6x_8 + x_4x_6 + x_3x_8,\\
\bar{\partial}_\infty(x_1x_3x_5) &= x_5x_6,\\
\bar{\partial}_\infty(x_1x_3x_7) &= x_6x_7,\\
\bar{\partial}_\infty(x_1x_3x_8) &= x_6x_8,\\
\bar{\partial}_\infty(x_1x_4x_5) &= x_5x_8,\\
\bar{\partial}_\infty(x_1x_4x_6) &= x_6x_8,\\
\bar{\partial}_\infty(x_1x_4x_7) &= x_7x_8,\\
\bar{\partial}_\infty(x_2x_3x_4) &= x_5x_7 + x_4x_7 + x_3x_5,\\
\bar{\partial}_\infty(x_2x_3x_5) &= x_5x_7,\\
\bar{\partial}_\infty(x_2x_3x_6) &= x_6x_7,\\
\bar{\partial}_\infty(x_2x_3x_8) &= x_7x_8,\\
\bar{\partial}_\infty(x_2x_4x_6) &= x_5x_6,\\
\bar{\partial}_\infty(x_2x_4x_7) &= x_5x_7,\\
\bar{\partial}_\infty(x_2x_4x_8) &= x_5x_8,\\
\bar{\partial}_\infty(\gamma_1(y_1)x_1) &= x_1x_8,\\
\bar{\partial}_\infty(\gamma_1(y_2)x_1) &= \gamma_1(y_5) + x_2x_5,\\
\bar{\partial}_\infty(\gamma_1(y_3)x_1) &= \gamma_1(y_6) + x_3x_6,\\
\bar{\partial}_\infty(\gamma_1(y_4)x_1) &= \gamma_1(y_8) + x_4x_8,\\
\bar{\partial}_\infty(\gamma_1(y_1)x_2) &= \gamma_1(y_5) + x_2x_8 + x_1x_5,\\
\bar{\partial}_\infty(\gamma_1(y_3)x_2) &= \gamma_1(y_7) + x_3x_7,\\
\bar{\partial}_\infty(\gamma_1(y_4)x_2) &= \gamma_1(y_5) + x_4x_5,\\
\bar{\partial}_\infty(\gamma_1(y_1)x_3) &= \gamma_1(y_6) + x_3x_8 + x_1x_6,
\end{aligned}$$

$$\begin{aligned}
\bar{\partial}_\infty(\gamma_1(y_2)x_3) &= \gamma_1(y_7) + x_2x_7,\\
\bar{\partial}_\infty(\gamma_1(y_1)x_4) &= \gamma_1(y_8) + x_4x_8 + x_1x_8,\\
\bar{\partial}_\infty(\gamma_1(y_2)x_4) &= \gamma_1(y_5) + x_2x_5,\\
\bar{\partial}_\infty(\gamma_1(y_1)x_5) &= x_5x_8,\\
\bar{\partial}_\infty(\gamma_1(y_1)x_6) &= x_6x_8,\\
\bar{\partial}_\infty(\gamma_1(y_1)x_7) &= x_7x_8.
\end{aligned}$$

A An AXIOM-Program for $\mathrm{UT}_3(\mathbb{F}_5)$

The first part defines the ingredients for this particular group. From the line starting with The data structures on, the program is generally applicable for other p-groups as well. It computes ∂_∞ (dNew) and ∇_∞ (nablaNew) as well as $\bar{\partial}_\infty$ (dNewReduced) and $\bar{\nabla}_\infty$ (nablaNewReduced) for the degrees 1 up to 4. For brevity, obvious parts are left out. Similar programs for computing the projection f_∞ and the contracting homotopy ϕ_∞ can be found on the internet (see 8.1).

```
)clear all
)spool ut3-5.out

-- load all the necessary code
)r loadall

-- the prime
p := 5
-- the field with p elements
F := PrimeField p
-- the dimension
n := 3

-- sets of variables
ly := [subscript('y,[i]) for i in 1..n]
lt := [subscript('t,[i]) for i in 1..n]
lx := [subscript('x,[i]) for i in 1..n]

-- the group law for the 3x3 upper triangular matrices
-- with 1's along the diagonal

rho:(List PF 5,List PF 5) -> List PF 5
rho(x,y) == [x.1+y.1,x.2+y.2,x.3+y.3+x.1*y.2]

------------------------------------------------------------------------------------------
-- The data structures
------------------------------------------------------------------------------------------
-- the group
G := PPGP(p, n, rho)

-- the elementary abelian group of order p^n
Gp := MultiplicativelyWrittenElementaryAbelian(p, n)

-- the group algebra of the elementary abelian group Gp
-- written multiplicatively
Ap := MonoidRing(F, Gp)

-- The divided power algera over Ap.  This is the algebra
-- Fp( (Z/pZ)^n ) x Gamma(y1,..,yn) where "x" denotes tensor
-- product over F, the prime field with p elements,
-- and Gamma is the algebra with infinitely
-- many generators g_i(j_j), i = 0,1,..., j = 1,..,n and
```

```
-- multiplication g_i(y_j)g_k(y_j) = [i+j,i]g_{i+j} where
-- [i+j,i] denotes the binomial coefficient mod p.
DP := DIVPOW(Ap, n, ly)

-- This is the Cartan "little resolution" over the group
-- ring of (Z/pZ)^n.  It is the differential graded augmented
-- algebra Z/pZ( (Z/pZ)^n ) x Gamma(y1,..,yn) x Lambda[x1,..,xn]
-- where Lambda denoted the exterior algebra and d is the classical
-- differential.  This includes the contracting homotopy.
C := CLR(p, n, lt, ly, lx)

--  This is the bar construction of Eilenberg and MacLane
--  for (Z/pZ)^n.
Bp := BAR(F, Gp)

--  This is the bar construction of Eilenberg and MacLane
--  for G.
B   := BAR(F, G)

--  This is the strong deformation retraction of the
--  Cartan little resolution into the bar construction.
--  It includes the inclusion, the retraction and the homotopy.
SDR := SDRPG(p,n,lt,ly,lx)

------------------------------------------------------------------------------------------
-- The initiator
------------------------------------------------------------------------------------------
--  The package PerturbationUtilites provides conversion functions (::, coerce)
--  to accomplish the isomorphisms Xi and Theta from the paper.
--  The actual initiator T:
tee : Bp -> Bp
tee(b) ==
  d1 :=  d(b :: B )$B :: Bp
  d2 :=  d(b)$Bp
  d1-d2

-- t composed with phi = homot
tphi : Bp -> Bp
tphi(b) == tee homot(b)$SDR

------------------------------------------------------------------------------------------
-- The iterated transference process for the chain maps on Cartan's little resolution
------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------
-- The basis elements in Cartan's little resolution
------------------------------------------------------------------------------------------
--  Now we go about constructing the degree 1, 2, 3, 4 components in the Cartan little
--  resolution.  The canonical bases of Cartan's little resolutions for degrees 1,2,3,4
Cdegree1 := [monomial(gamma(r.divpow)$DP, r.extalg)$C for r in basisOfDegree(1)$CARTUTS(p,n)]
Cdegree2 := [monomial(gamma(r.divpow)$DP, r.extalg)$C for r in basisOfDegree(2)$CARTUTS(p,n)]
Cdegree3 := [monomial(gamma(r.divpow)$DP, r.extalg)$C for r in basisOfDegree(3)$CARTUTS(p,n)]
Cdegree4 := [monomial(gamma(r.divpow)$DP, r.extalg)$C for r in basisOfDegree(4)$CARTUTS(p,n)]


------------------------------------------------------------------------------------------
-- degree 1
------------------------------------------------------------------------------------------
tphiListCdegree1 : List List Bp := [];
zerosBp1 : List Bp := [0$Bp for i in 1..#Cdegree1];
nablaCdegree1 := [inc(c)$SDR for c in Cdegree1];

-- apply initiator to images of basis elements of degree 1
tnablaCdegree1 := [tee bp for bp in nablaCdegree1];
zerosBpCheck: Boolean := (nablaCdegree1 = zerosBp1);
if not zerosBpCheck then tphiListCdegree1 := [tnablaCdegree1];
while (not zerosBpCheck) repeat
   tphiIteration := [tphi bp for bp in tphiListCdegree1.1]
   zerosBpCheck := (tphiIteration = zerosBp1)
   if (not zerosBpCheck) then tphiListCdegree1 := cons(tphiIteration,tphiListCdegree1)

------------------------------------------------------------------------------------------
-- degree 2
------------------------------------------------------------------------------------------

...
```

```
-----------------------------------------------------------------------------------------
-- degree 4
-----------------------------------------------------------------------------------------

...

-----------------------------------------------------------------------------------------
-- the new limit differential on Cartan's Little Resolution
-----------------------------------------------------------------------------------------

-- the lists tphiListCdegree4 contain all the non-zero powers of tphi
-- now project the summands back (f = proj) to Cartan's Little Resolution
ftphiListCdegree1 := [ [proj(bp)$SDR for bp in LL] for LL in reverse tphiListCdegree1];
ftphiListCdegree2 := [ [proj(bp)$SDR for bp in LL] for LL in reverse tphiListCdegree2];
ftphiListCdegree3 := [ [proj(bp)$SDR for bp in LL] for LL in reverse tphiListCdegree3];
ftphiListCdegree4 := [ [proj(bp)$SDR for bp in LL] for LL in reverse tphiListCdegree4];

-- the lists dCdegree contain the applications of the given differential on C to the basis
-- elements, for uniform handling, also put them into ftphiListCdegree
dCdegree1 := [d c for c in Cdegree1];
dCdegree2 := [d c for c in Cdegree2];
dCdegree3 := [d c for c in Cdegree3];
dCdegree4 := [d c for c in Cdegree4];

ftphiListCdegree1 := cons(dCdegree1, ftphiListCdegree1);
ftphiListCdegree2 := cons(dCdegree2, ftphiListCdegree2);
ftphiListCdegree3 := cons(dCdegree3, ftphiListCdegree3);
ftphiListCdegree4 := cons(dCdegree4, ftphiListCdegree4);

-- the new differential is the sum of the given differential and the result of
-- the perturbation process (perturbation lemma)
sumListsC(lc: List C, lc': List C): List C == [c+c' for c in lc for c' in lc']
-- summing up all lists componentwise
dnewCdegree1 := reduce(sumListsC, ftphiListCdegree1, [0$C for i in 1..#Cdegree1])
dnewCdegree2 := reduce(sumListsC, ftphiListCdegree2, [0$C for i in 1..#Cdegree2])
dnewCdegree3 := reduce(sumListsC, ftphiListCdegree3, [0$C for i in 1..#Cdegree3])
dnewCdegree4 := reduce(sumListsC, ftphiListCdegree4, [0$C for i in 1..#Cdegree4])

-- some function for pretty output
--)r homolprt

-- function for pretty printing
O := OutputForm
say(str) == messagePrint(str)$OutputForm
form(c,yy) == print(hconcat [message("d ")$O,c::O,message(" = ")$O,yy::O])$O
form(f, x, fx) == print(hconcat [message(concat(f,"("))$O,x::O,message(") = ")$O,fx::O])$O

printChainMap(f, degree, basisList, fbasisList) ==
  say "---------------------------------------------------------------------------"
  print(center ["  The chain map "::O, f::O])$O
  print(center [" on the canonical basis elements of degree "::O, degree :: O])$O
  say "---------------------------------------------------------------------------"
  for x in basisList for fx in fbasisList repeat form(f, x, fx)


printChainMap("dNew", 1, Cdegree1, dnewCdegree1)
printChainMap("dNew", 2, Cdegree2, dnewCdegree2)
printChainMap("dNew", 3, Cdegree3, dnewCdegree3)
printChainMap("dNew", 4, Cdegree4, dnewCdegree4)

-----------------------------------------------------------------------------------------
-- the new limit inclusion nabla from Cartan's Little Resolution to Bar Construction
-----------------------------------------------------------------------------------------

-- the lists tphiListCdegree contain all the non-zero powers of tphi
-- now once again apply homotopy phi
phitphiListCdegree1 := [ [homot(bp)$SDR for bp in LL] for LL in reverse tphiListCdegree1];
phitphiListCdegree2 := [ [homot(bp)$SDR for bp in LL] for LL in reverse tphiListCdegree2];
phitphiListCdegree3 := [ [homot(bp)$SDR for bp in LL] for LL in reverse tphiListCdegree3];
phitphiListCdegree4 := [ [homot(bp)$SDR for bp in LL] for LL in reverse tphiListCdegree4];

-- the lists nablaCdegree are already constructed,
-- for uniform handling, also put them into phitphiListCdegree

phitphiListCdegree1 := cons(nablaCdegree1, phitphiListCdegree1);
```

```
phitphiListCdegree2 := cons(nablaCdegree2, phitphiListCdegree2);
phitphiListCdegree3 := cons(nablaCdegree3, phitphiListCdegree3);
phitphiListCdegree4 := cons(nablaCdegree4, phitphiListCdegree4);

-- the new limit inclusion nablanew is the sum of the given inclusion and the result of
-- the perturbation process (perturbation lemma)
sumListsBp(lbp: List Bp, lbp': List Bp): List Bp== [bp+bp' for bp in lbp for bp' in lbp'];
-- summing up all lists componentwise
nablanewCdegree1 := reduce(sumListsBp, phitphiListCdegree1, [0$Bp for i in 1..#Cdegree1]);
nablanewCdegree2 := reduce(sumListsBp, phitphiListCdegree2, [0$Bp for i in 1..#Cdegree2]);
nablanewCdegree3 := reduce(sumListsBp, phitphiListCdegree3, [0$Bp for i in 1..#Cdegree3]);
nablanewCdegree4 := reduce(sumListsBp, phitphiListCdegree4, [0$Bp for i in 1..#Cdegree4]);

printChainMap("nablaNew", 1, Cdegree1, nablanewCdegree1)
printChainMap("nablaNew", 2, Cdegree2, nablanewCdegree2)
printChainMap("nablaNew", 3, Cdegree3, nablanewCdegree3)
printChainMap("nablaNew", 4, Cdegree4, nablanewCdegree4)


-----------------------------------------------------------------------------------------
-- the new limit inclusion nabla from Cartan's Little Resolution to Bar Construction
-----------------------------------------------------------------------------------------

-- the lists tphiListCdegree4 contain all the non-zero powers of tphi
-- now once again apply homotopy phi
phitphiListCdegree1 := [ [homot(bp)$SDR for bp in LL] for LL in reverse tphiListCdegree1];
phitphiListCdegree2 := [ [homot(bp)$SDR for bp in LL] for LL in reverse tphiListCdegree2];
phitphiListCdegree3 := [ [homot(bp)$SDR for bp in LL] for LL in reverse tphiListCdegree3];
phitphiListCdegree4 := [ [homot(bp)$SDR for bp in LL] for LL in reverse tphiListCdegree4];

-- the lists nablaCdegree are already constructed,
-- for uniform handling, also put them into phitphiListCdegree

phitphiListCdegree1 := cons(nablaCdegree1, phitphiListCdegree1);
phitphiListCdegree2 := cons(nablaCdegree2, phitphiListCdegree2);
phitphiListCdegree3 := cons(nablaCdegree3, phitphiListCdegree3);
phitphiListCdegree4 := cons(nablaCdegree4, phitphiListCdegree4);

-- the new limit inclusion nablanew is the sum of the given inclusion and the result of
-- the perturbation process (perturbation lemma)
sumListsBp(lbp: List Bp, lbp': List Bp): List Bp== [bp+bp' for bp in lbp for bp' in lbp']
-- summing up all lists componentwise
nablanewCdegree1 := reduce(sumListsBp, phitphiListCdegree1, [0$Bp for i in 1..#Cdegree1]);
nablanewCdegree2 := reduce(sumListsBp, phitphiListCdegree2, [0$Bp for i in 1..#Cdegree2]);
nablanewCdegree3 := reduce(sumListsBp, phitphiListCdegree3, [0$Bp for i in 1..#Cdegree3]);
nablanewCdegree4 := reduce(sumListsBp, phitphiListCdegree4, [0$Bp for i in 1..#Cdegree4]);


-----------------------------------------------------------------------------------------
-- reduction, i.e. tensoring with the ground field, i.e. summing up coefficients
-----------------------------------------------------------------------------------------

-- Here we tensor the resolution with Z/pZ over the group ring:
reductionC : C -> C
reductionC c ==
  brFc := basisRepresentationOverF(c)$C
  ans : C := 0
  -- note that we do that by simply 'forgetting' the group elements
  -- i.e. summing up their coefficients for the augmentation
  for rc in brFc repeat ans := ans + rc.f * gamma(rc.y)$DP * (rc.x::EAB::C)
  ans

-- Here we tensor the bar construction with Z/pZ over the group ring:
reductionBp : Bp ->Bp
reductionBp bp == reduce(+, [monomial(epsilon(c)::Ap, s)$Bp for c in coefficients bp
    for s in support bp],0$Bp)

say("-------------------------------------------------------------------------")
say("  The reduced complex ")
say("-------------------------------------------------------------------------")


dnewredCdegree1 := [reductionC dnewc for  dnewc in dnewCdegree1];
nablanewredCdegree1 := [reductionBp nablanewc for  nablanewc in nablanewCdegree1];

printChainMap("dNewReduced", 1, Cdegree1, dnewredCdegree1)
```

```
printChainMap("nablaNewReduced", 1, Cdegree1, nablanewredCdegree1)

...

printChainMap("nablaNewReduced", 4, Cdegree4, nablanewredCdegree4)

say("--------------------------------------------------------------------------")
say "-- End of homological computation (differential, inclusion) of given Group"
say("--------------------------------------------------------------------------")


)spool
```

References

1. Donald W. Barnes and Larry A. Lambe. A fixed point approach to homological perturbation theory. *Proc. Amer. Math. Soc.*, 112(3):881–892, 1991.
2. R. Brown. The twisted Eilenberg-Zilber theorem. In *Simposio di Topologia (Messina, 1964)*, pages 33–37. Edizioni Oderisi, Gubbio, 1965.
3. Henri Cartan and Samuel Eilenberg. *Homological algebra*. Princeton University Press, Princeton, N. J., 1956.
4. Henri Cartan, J. C. Moore, R. Thom, and J. P. Serre. *Algèbras d'Eilenberg-MacLane et homotopie. 2ieme ed., revue et corrig ee.* Secretariat mathematique (Hektograph), Paris, 1956.
5. Torsten Ekedahl, Johannes Grabmeier, and Larry Lambe. A Generic Language for algebraic computations, 2000. In preparation.
6. G. Ellis and I. Kholodna. Second cohomology of finite groups with trivial coefficients. *Homology, Homotopy & Appl.*, 1:163–168, 1999.
7. D. L. Flannery. Transgression and the calculation of cocyclic matrices. *Australas. J. Combin.*, 11:67–78, 1995.
8. D. L. Flannery. Calculation of cocyclic matrices. *J. Pure Appl. Algebra*, 112(2):181–190, 1996.
9. D. L. Flannery, K. J. Horadam, and W. de Launey. Cocyclic hadamard matrices and difference sets. *Discrete Appl. Math.*, 102:47–61, 2000.
10. 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.
11. R.L. Graham, D.E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, Reading, Massachusetts, 1989.
12. V. K. A. M. Gugenheim. On the chain-complex of a fibration. *Illinois J. Math.*, 16:398–414, 1972.
13. V. K. A. M. Gugenheim and L. A. Lambe. Perturbation theory in differential homological algebra. I. *Illinois J. Math.*, 33(4):566–582, 1989.
14. V. K. A. M. Gugenheim, L. A. Lambe, and J. D. Stasheff. Perturbation theory in differential homological algebra. II. *Illinois J. Math.*, 35(3):357–373, 1991.
15. K. J. Horadam and W. de Launey. Cocyclic development of designs. *J. Algebraic Combin.*, 2(3):267–290, 1993.
16. K. J. Horadam and W. de Launey. Erratum: "Cocylic development of designs". *J. Algebraic Combin.*, 3(1):129, 1994.
17. K. J. Horadam and W. de Launey. Generation of cocyclic Hadamard matrices. In *Computational algebra and number theory (Sydney, 1992)*, volume 325 of *Math. Appl.*, pages 279–290. Kluwer Acad. Publ., Dordrecht, 1995.
18. K. J. Horadam and A. A. I. Perera. Codes from cocycles. In *Lecture Notes in Computer Science*, volume 1255, pages 151–163. Springer-Verlag, Berlin-Heidelberg-New York, 1997.
19. Johannes Huebschmann. The homotopy type of $F\psi^q$. The complex and symplectic cases. In *Applications of algebraic K-theory to algebraic geometry and number theory, Part I, II (Boulder, Colo., 1983)*, pages 487–518. Amer. Math. Soc., Providence, R.I., 1986.
20. Johannes Huebschmann. Perturbation theory and free resolutions for nilpotent groups of class 2. *J. Algebra*, 126(2):348–399, 1989.
21. Johannes Huebschmann and Tornike Kadeishvili. Small models for chain algebras. *Math. Z.*, 207(2):245–280, 1991.

22. Bertram Huppert and Norman Blackburn. *Finite groups. II.* Springer-Verlag, Berlin-New York, 1982. Grundlehren der Mathematischen Wissenschaften, Band 242.
23. N. Jacobson. Restricted Lie algebras of characteristic p. *Trans. Amer. Math. Soc.*, 50:15–25, 1941.
24. Richard D. Jenks and Robert S. Sutor. *Axiom. The scientific computation system.* Springer-Verlag, Berlin, Heidelberg, New York, 1992.
25. S. A. Jennings. The structure of the group ring of a p-group over a modular field. *Trans. Amer. Math. Soc.*, 50:175–185, 1941.
26. Leif Johansson and Larry Lambe. Transferring algebra structures up to homology equivalence. *Math. Scand.*, 88(2), 2001.
27. Leif Johansson, Larry Lambe, and Emil Sköldberg. On constructing resolutions over the polynomial algebra, 2000. Preprint.
28. Larry Lambe. Next generation computer algebra systems AXIOM and the scratchpad concept: applications to research in algebra. In *Analysis, algebra, and computers in mathematical research (Luleå, 1992)*, volume 156 of *Lecture Notes in Pure and Appl. Math.*, pages 201–222. Dekker, New York, 1994.
29. Larry Lambe. The 1996 Adams Lectures at Manchester University: New computational methods in algebra and topology, May 20 1996.
30. Larry Lambe and Jim Stasheff. Applications of perturbation theory to iterated fibrations. *Manuscripta Math.*, 58(3):363–376, 1987.
31. Larry A. Lambe. Resolutions via homological perturbation. *J. Pure Appl. Algebra*, 12:71–87, 1991.
32. Larry A. Lambe. Homological perturbation theory, Hochschild homology, and formal groups. In *Deformation theory and quantum groups with applications to mathematical physics (Amherst, MA, 1990)*, volume 134 of *Contemp. Math.*, pages 183–218. Amer. Math. Soc., Providence, RI, 1992.
33. Larry A. Lambe. Resolutions which split off of the bar construction. *J. Pure Appl. Algebra*, 84(3):311–329, 1993.
34. Saunders Mac Lane. *Homology.* Classics in Mathematics. Springer-Verlag, Berlin, 1995. Reprint of the 1975 edition.
35. J. Peter May. The cohomology of restricted Lie algebras and of Hopf algebras. *Bull. Amer. Math. Soc.*, 71:372–377, 1965.
36. D. Quillen. On the associated graded ring of a group ring. *J. Algebra*, 10:411–418, 1968.
37. Jean-Pierre Serre. *Lie algebras and Lie groups. 1964 lectures, given at Harvard University, 2nd ed.*, volume 1500 of *Lecture Notes in Mathematics.* Springer-Verlag, Berlin-Heidelberg-New York, 1992.
38. C.T.C. Wall. Resolutions for extensions of groups. *Proc. Phil. Soc.*, 57:251–255, 1961.