Calculating cohomology

Jon F. Carlson

University of Georgia

http://www.math.uga.edu/~jfc

What I mean by "computing cohomology"

By "computing cohomology", I mean computing $\mathrm{Ext}^*_A(M,N)$ where A is an algebra and M and N are A-modules. The answer might be returned as a module over $\mathrm{Ext}^*_A(M,M)$ or over some object that you know. Or if you are doing something like $\mathrm{Ext}^*_A(M,M)$, it might be given by generators and relations (a presentation).

I do NOT mean computing $\mathrm{Ext}^1_A(M,N)$ or $\mathrm{Ext}^2_A(M,N)$. These might be very interesting and important, but the methods involved in the computation of low dimensional cohomology, in general, are very different. It is not my intention to discuss them here.

Basic definitions

Suppose that A is an algebra over a field k, which for computing purposes will be assumed to be finite. Let M and N be A-modules. Let

$$\cdots \longrightarrow P_2 \longrightarrow P_1 \longrightarrow P_0 \longrightarrow M \longrightarrow 0$$

be a projective resolution of k. So then for any kG-module N,

$$\mathrm{Ext}^n_A(M,N) \cong \mathrm{H}^n(\mathrm{Hom}_A(P_*,N))$$

In the case that $A = kG$ for G a finite group and $M = k$, we have that

$$\mathrm{H}^*(G,N) = \mathrm{Ext}^n_{kG}(k,N) \cong \mathrm{H}^n(\mathrm{Hom}_{kG}(P_*,N))$$

A cohomology element $\zeta \in \mathrm{Ext}^*_A(M,N)$ is represented by a chain map

$$\begin{array}{ccccccccc}
\cdots \longrightarrow & P_{n+1} & \longrightarrow & P_n & \longrightarrow & P_{n-1} & & \cdots & \\
 & \downarrow \zeta_1 & & \downarrow \zeta_0 & \searrow^{\zeta} & & & & \\
\cdots \longrightarrow & Q_1 & \longrightarrow & Q_0 & \longrightarrow & N & \longrightarrow & 0. &
\end{array}$$

Cup product can be computed as composition of chain maps.

$$\mathrm{Ext}^*_A(M,N) \otimes \mathrm{Ext}^*_A(L,M) \longrightarrow \mathrm{Ext}^*_A(L,N)$$

Trick No. 1: First reduce to the basic algebra

The point is that if you have an algebra that is small enough that you can think about computing cohomology in some sort of direct way, then it is small enough that you can compute the basic algebra. If A is a finite dimensional algebra, then the basic algebra of A is $\mathrm{End}_A(P)$ where $P = \sum P_i$ is a direct sum of the indecomposable projective modules (one of each isomorphism class). The basic algebra B is also isomorphic to eAe where e is a sum of primitive idempotents, one for each isomorphism class of irreducible modules.

The point is that B is Morita equivalent to A. So they have the same module theory and the same cohomology. Hence computing in the basic algebra is equivalent to computing in the algebra.

Trick No. 2: Only compute cohomology of irreducible modules

If M is irreducible, then we can compute a minimal projective resolution (P_*, ε) of M. Now if N is also irreducible we have that

$$\mathrm{Hom}_A(P_n, N) \cong \mathrm{Ext}_A^n(M, N).$$

Thus we don't have the problems of computing coboundaries and taking the quotients of cocycles by coboundaries.

Cohomology rings of p-groups

A little history – what's available:

Larry Lambe (stand alone – perturbation method – this is the oldest implementation that I know about.)
Me and my students (available in MAGMA – uses linear algebra)
David Green (stand alone – uses noncommutative Groebner bases)
Marcus Bishop (available GAP – uses linear algebra)
Graham Ellis (available in GAP – perturbation method)

So suppose that G is a p-group. Then kG is a local ring and is isomorphic to its own basic algebra. So we can

(1) compute a minimal projective resolution of the trivial module k,
(2) get chain maps for the generators of the cohomology and
(3) compute the relations among the generators.

Trick No. 3: Never let the computer know that you are computing an A-resolution.

The machine should always think that you are only doing linear algebra.

Now a demonstration

Stable elements

Suppose that P is a Sylow p-subgroup of G and then somehow you have computed $\mathrm{H}^*(P, k)$. Then $\mathrm{H}^*(G, k) \subseteq \mathrm{H}^*(P, k)$ is the subring of stable elements. It consists of all $\zeta \in \mathrm{H}^*(P, k)$ with the property that

$$c_g(\mathrm{res}_{P,Q}(\zeta)) = \mathrm{res}_{P,{}^gQ}(\zeta)$$

for all $g \in G$ and $Q \subseteq P$. That is, the diagrams

$$\begin{array}{ccc} & \mathrm{H}^*(P,k) & \\ {\scriptstyle\mathrm{res}}\swarrow & & \searrow{\scriptstyle\mathrm{res}} \\ \mathrm{H}^*(Q,k) & \xrightarrow{\;c_g\;} & \mathrm{H}^*({}^gQ,k) \end{array}$$

must all commute.

This technique has been used many times but has never been well implemented.

Basic algebras and Ext algebras

For the more general *Ext* functor, there is a basic algebras package in MAGMA that is optimized for homological algebra operations like

(1) projective resolution,
(2) module complexes,
(3) chain maps,
(4) homology of complexes.

Klaus Lux has written a similar thing for GAP (?), and in fact, his package will do things like the *Ext* algebra, meaning $\mathrm{Ext}_A^*(M, M)$ where M is the direct sum of the simple modules. The MAGMA package could easily be adapted to compute the *Ext* algebra and it is on my list of things to do.

There are two problems associated with this.

(1.) How do you get the basic algebra from the algebra that you are studying?

If the algebra is given as a matrix algebra over a finite field, then there is a package in MAGMA that will give you a condensed algebra which can be easily made into a basic algebra. For other algebras there are methods, but nothing that is well implemented.

(2.) When do you know that you are done?

For $\mathrm{H}^*(G, k)$ there is an elaborate scheme for knowing when the computation is done (recently refined by a theorem of Dave Benson). For *Ext* algebras this is more of a guess.

Varieties and Quillen categories

Assume that k is algebraically closed. The cohomology ring $\mathrm{H}^*(G,k)$ is a finitely generated k-algebra and has a maximal ideal spectrum $V_G(k)$.

For example, suppose that $G \cong (\mathbb{Z}/p\mathbb{Z})^n$ is an elementary abelian group of order p^n. Then

$$\mathrm{H}^*(G,k) \cong k[\zeta_1,\dots,\zeta_n] \otimes \Lambda(\eta_1,\dots,\eta_n)$$

if $p > 2$, and

$$\mathrm{H}^*(G,k) \cong k[\zeta_1,\dots,\zeta_n]$$

if $p = 2$. Here $deg(\zeta_i) = 2$, $deg(\eta_i) = 1$ when $p > 2$ and $deg(\zeta_i) = 1$ when $p = 2$.

In both cases, we have that $V_G(k) \cong k^n$.

For another example, suppose that $p = 2$ and G is a dihedral group of order 8. We have that .

$$\mathrm{H}^*(G, k) \;\cong\; k[z, y, x]/(zy)$$

where $deg(z) = deg(y) = 1$ and $deg(x) = 2$. So in this case $V_G(k)$ is a union of two (twisted) planes.

More generally we have the following. Let $\mathcal{EA}$ denote the collection of all elementary abelian p-subgroups.

Theorem 1. *(Quillen) For any finite group G we have that*

$$V_G(k) \ = \ \cup_{E \in \mathcal{EA}} \mathrm{res}^*_{G,E}(V_E(k))$$

In fact, $\mathcal{EA}$ is a category with the inclusion and conjugation maps being the morphisms. Quillen's theorem really says that $V_G(k)$ is the limit over this category.

What is the Quillen category for a finite group of Lie type? Can we describe it in any generality?