Abstract.

In this paper, we describe a set of APL programs prepared by the author for computation of homology groups of simplicial complexes. These programs may be used in conjunction with a first course in algebraic topology in several ways: to help the student gain an intuitive feel for homology theory, to suggest plausible conjectures in algebraic topology, and to aid proofs.

I. Introduction.

Use of the computer as an aid to the discovery and proof of mathematical theorems has increased greatly in the past ten years. One example that immediately comes to mind is the machine-aided proof of the four-color conjecture by Haken and Appel (see [1]). Computers have also been used extensively in group theory [2 3] and number theory [4 5].

In this paper, we describe a set of APL programs prepared by the author for computation of homology groups of simplicial complexes. The entire set of programs takes up little more than one sheet of paper, and most are written in the direct definition (a-oJ) formalism (see the appendix). These programs may be used in conjunction with a first course in algebraic topology in several ways: to help the student gain an intuitive feel for homology theory, to suggest plausible conjectures in algebraic topology, and to aid proofs.

II. Algebraic Topology.

An n-simplex is the n-dimensional generalization of a triangle in 2-space, i.e. a set homeomorphic to

\[ \{(t_0, t_1, \ldots, t_n) \mid t_i \geq 0, t_0 + t_1 + \cdots + t_n = 1\} \]

the standard simplex. The points \((1,0,0,\ldots,0)\), \((0,1,0,\ldots,0)\), etc. are called the vertices of the standard simplex.

The programs exemplify the use of the computer to compute, not as a tutor or instructor. Use of APL in this way has been encouraged for more than ten years; for example, see [6].

In many ways, APL is the ideal language to express algebraic concepts. First, its function-oriented structure applies naturally to algebraic systems. Second, its powerful array processing allows easy manipulation of matrices. Finally, APL's conciseness and power make development and testing easy.
A simplex \( A \) is said to be a face of a simplex \( B \) if \( A \) is a subset of \( B \) and every vertex of \( A \) is also a vertex of \( B \). For example, the 3-simplex above has 15 faces: the four points, the six lines, the four triangles, and the tetrahedron making up the 3-simplex itself. The dimension of a simplex is one less than the number of vertices defining it.

Two simplices \( A \) and \( B \) are said to be properly joined provided that either \( A \cap B = \emptyset \) or \( A \cap B \) is a face of both \( A \) and \( B \). See Figure 2.

![Proper Joining and Improper Joining](image)

Figure 2

A simplicial complex \( K \) is a finite collection of properly joined simplices such that each face of a member of \( K \) is also a member of \( K \).

Through the process of triangulation, common topological objects may be represented by simplicial complexes. For example, the figure below is a triangulation of the Moebius strip, a one-sided surface.

![Triangulation of Moebius Strip](image)

Figure 3: Triangulation of Moebius Strip

Associated with each simplicial complex is a set of groups called homology groups. These groups partially describe the structure of the complex, and methods for their computation form a large part of elementary algebraic topology.

For more on these concepts, see [7 8].

The APL workspace HOMOLOGY contains a set of functions and variables to manipulation representations of simplicial complexes, including the computation of their homology groups.

### III. Representations

An important question is: How should simplicial complexes be represented? We want a representation that is both efficient (in terms of storage requirements) and easily manipulated.

One representation (which we will call B-representation) is through Boolean matrices, where each row represents a face of a complex. For example, the Moebius strip could be represented by the following matrix of size 24 x 5:

```
BFVVFP MOEBIUS
1  1  0  0  0  0
1  1  0  0  0  0
1  1  0  1  0  0
1  0  0  1  0  0
1  0  0  1  0  1
1  0  0  0  0  1
0  1  0  0  0  0
0  1  0  0  0  0
0  1  0  1  0  0
0  1  0  1  0  0
0  1  0  1  1  0
0  1  0  1  1  0
0  0  1  0  0  0
0  0  1  1  0  0
0  0  0  1  1  0
0  0  0  1  1  0
0  0  0  0  1  0
```

The presence of a 1 in column \( N \) indicates that vertex \( N \) is included in the face represented by the given row.

Another representation (which we will call V-representation) is to again use rows to represent faces, but give the vertices explicitly (numbered from 1 to \( N \), padded with zeros on the right, if necessary). The order in which rows appear is unimportant, but within each row the non-zero entries should appear in ascending order. The Moebius strip could be represented by the following matrix:

```
VFP MOEBIUS
1  0  0
1  2  0
1  2  4
1  4  0
1  4  6
1  6  0
2  0  0
2  3  0
2  3  5
2  4  0
2  4  5
2  5  0
3  0  0
3  4  0
3  4  6
3  5  0
3  5  6
3  6  0
4  0  0
4  5  0
4  6  0
5  0  0
5  6  0
6  0  0
```

Another alternative, which is more efficient in terms of storage requirements, consists of representing complexes by listing only the principal simplices, i.e. those maximal with respect to inclusion. We call this the P-representation. To continue with our example of the Moebius strip, a P-representation might be
The last possibility is a cross between P- and B-representations, where only principal simplices are given, but they are listed as rows in a Boolean matrix. This is the representation given in [9 10]. We will not use it in this paper.

We also want functions to convert between the various representations. These are given as follows:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFV</td>
<td>Convert to B-rep from V-rep</td>
</tr>
<tr>
<td>VFB</td>
<td>Convert to V-rep from B-rep</td>
</tr>
<tr>
<td>VFP</td>
<td>Convert to V-rep from P-rep</td>
</tr>
<tr>
<td>PFV</td>
<td>Convert to P-rep from V-rep</td>
</tr>
</tbody>
</table>

Following the suggestion of Berry [11], these functions are designed to link together easily: for example, \(PFV \cdot VFB \cdot M\) converts \(M\) from Boolean to principal simplex representation.

Unless stated otherwise, most functions in the HOMOLOGY workspace assume arguments in V-representation. Although this representation is less efficient in terms of space needed, the ease of manipulation more than compensates for this deficiency.

IV. Homology.

Suppose we are given a simplicial complex \(A\) in V-representation. The faces of \(A\) can be classified according to their dimension; for example, the Moebius strip has 6 faces of dimension 2.

Suppose \(A\) has \(N\) faces of dimension \(P\) and \(M\) faces of dimension \(P-1\). We can then form a matrix of size \((N,M)\), say \(P \cdot \text{INCID} \cdot A\), which contains a 1 in row \(I\) and column \(J\) iff the \(J\)-th subsimplex of dimension \(P-1\) is a face of the \(I\)-th subsimplex of dimension \(P\). All other entries are 0. The signs of the non-zero elements are chosen to alternate across rows. These matrices are called incidence matrices. For example,

The rows of these incidence matrices can be considered to be generators of an abelian group; algebraically speaking, the rows span a submodule of \(\mathbb{Z}^n\). The rows of the incidence matrix may not form a basis, however, and it is important later to have a minimal spanning set. A matrix with integer entries can be reduced to one in upper triangular form that spans the same submodule by a type of Euclidean algorithm; this is done by the APL function \(UT\). \(UT \cdot M\) produces a matrix such that \((UT \cdot M)^+ \cdot x\) is in upper triangular form. We can then delete rows containing all zeroes with the function \(REDUCE\), which generates a minimal spanning set. Note that only integer operations are used.

For example:

\[
\begin{align*}
\text{REDUCE:} & \quad \text{RED TR} \cdot w = \text{MIN SPANNING ROW SET} \\
\text{TRI:} & \quad (\text{UT} \cdot w)^+ = \text{CHANCE TO UPPER TRIANG FORM} \\
\text{REDUCE:} & \quad (UT \cdot M) + \cdot x \text{ REDUCE} \; \text{INCID STRIP}
\end{align*}
\]

In a similar fashion, it is possible to obtain a basis for the null space of a given matrix; in fact, this is nothing more than the rows of \(UT \cdot M\) corresponding to the zero rows of \((UT \cdot M)^+ \cdot x\). For example:

\[
\begin{align*}
\text{REDUCE:} & \quad \text{RED TR} \cdot w = \text{MIN SPANNING ROW SET} \\
\text{TRI:} & \quad (\text{UT} \cdot w)^+ = \text{CHANCE TO UPPER TRIANG FORM} \\
\text{REDUCE:} & \quad (UT \cdot M) + \cdot x \text{ REDUCE} \; \text{INCID STRIP}
\end{align*}
\]

Note that the matrices given by \(Z\) and \(B\) above have the same number of columns. The group (or \(Z\)-module) represented by \(Z\) is called the cycle group and that represented by \(B\) is called the boundary group. We can find a function that maps \(B\) into \(Z\); this amounts to expressing the rows of \(B\) as integer linear combinations of rows of \(Z\) and is neatly given by the \(I^+\) function. In theory, the result will be a matrix with integer entries, but round-off error occasionally obscures this.

\[
\begin{align*}
\text{I^+:} & \quad (\text{UT} \cdot M)^+ + \cdot x \text{ REDUCE} \; \text{INCID STRIP}
\end{align*}
\]
The matrix $H$ given by the matrix division "imbeds" $B$ into $Z$; a well-known theorem says that we can change $H$ into a diagonal matrix through elementary row and column operations. This is done by the function $\text{DIAG}$.

$$\text{DIAG} : \text{TRI} \rightarrow \text{TRI} \rightarrow A \rightarrow \text{REDUCE TO DIAGONAL FORM}$$

$$\text{LAH} = 1$$

$$\begin{array}{ccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}$$

The interpretation of this matrix is as follows: The quotient group represented by $H$ is isomorphic to a sum of cyclic groups:

$$\mathbb{Z}/a_1\mathbb{Z} + \mathbb{Z}/a_2\mathbb{Z} + \cdots + \mathbb{Z}/a_n\mathbb{Z}$$

where the $a_n$ are entries on the diagonal of $H$. Hence the number of rows containing all zeros correspond to copies of $\mathbb{Z}$, and other non-unit rows indicate the torsion (finite order) subgroups. In this case, $H$ is isomorphic to $\mathbb{Z}$.

The entire procedure is automated by the function $\text{CH}$, which takes a left argument of the order of the homology group to be computed and a right argument of a simplicial complex in vertex form. The result is the non-unit entries on the diagonal of the $H$ matrix.

Usually it is more amenable to employ the function $\text{CHOMP}$ (which stands for "Compute Homology, Please"); this has a left argument identical to that of $\text{CH}$ and a right argument of the name of a complex in P-representation. The result is a character matrix describing the homology groups.

$$\text{CHOMP 'MOEBIUS'}$$

$$H(0) = \mathbb{Z}$$

$$H(1) = \mathbb{Z}$$

$$H(2) = 0$$

V. Discovering Theorems.

The HOMOLOGY workspace provides P-representations of triangulations of some common topological objects:

- **KLEIN** Klein Bottle
- **PROJPLANE** Projective Plane
- **TORUS** Torus
- **MOEBIUS** Moebius Strip
- **CYL** Cylinder

Also provided are the following functions, which generate sequences of topological objects:

- **SPHERE N** The N-sphere
- **K N** Complete graph on N points

It is useful to be able to join these objects in various ways. The following "conjunctive" functions are provided:

- **A DU B** Disjoint union
- **A CS B** Connected sum
- **A CP B** Cartesian product

Direct sum corresponds to placing A beside B; connected sum cuts a "hole" corresponding to the first simplex in A and B and gluing the complexes together at the "hole". Cartesian product forms a new complex by a method similar to APL outer product.

For example,

$$\text{CHOMP 'KLEIN'}$$

$$H(0) = \mathbb{Z}$$

$$H(1) = \mathbb{Z}/2\mathbb{Z} + \mathbb{Z}$$

$$H(2) = 0$$

$$\text{CHOMP 'PROJPLANE CS PROJPLANE'}$$

$$H(0) = \mathbb{Z}$$

$$H(1) = \mathbb{Z}/2\mathbb{Z} + \mathbb{Z}$$

$$H(2) = 0$$

Even if he doesn't know the theorem that 2-pseudomanifolds are isomorphic iff they have the same homology groups, the student might conjecture: The Klein bottle is isomorphic to the connected sum of two projective planes.

In a similar fashion, the student could compare the homology groups of the torus with those of the Cartesian product of two circles (i.e. 1-spheres):

$$\text{CHOMP 'TORUS'}$$

$$H(0) = \mathbb{Z}$$

$$H(1) = \mathbb{Z} + \mathbb{Z}$$

$$H(2) = \mathbb{Z}$$

$$\text{CHOMP 'SPHERE 1) CP SPHERE 1'}$$

$$H(0) = \mathbb{Z}$$

$$H(1) = \mathbb{Z} + \mathbb{Z}$$

$$H(2) = \mathbb{Z}$$

There are many similar examples.

Comparing the groups of the the Klein bottle, projective plane, and objects like the torus, another plausible (in fact, true) conjecture would be: A 2-pseudomanifold is non-orientable iff its second homology group is $\{0\}$.

Let's now look at the homology groups of N-dimensional spheres. For example:

$$\text{CHOMP 'SPHERE 1'}$$

$$H(0) = \mathbb{Z}$$

$$H(1) = \mathbb{Z}$$

$$\text{CHOMP 'SPHERE 1'}$$

$$H(0) = \mathbb{Z}$$

$$H(1) = 0$$

$$H(2) = \mathbb{Z}$$

$$\text{CHOMP 'SPHERE 1'}$$

$$H(0) = \mathbb{Z}$$

$$H(1) = 0$$

$$H(2) = \mathbb{Z}$$
Judging from these examples, a plausible conjecture might be: The 0-th and N-th homology groups of the N-sphere are both \( \mathbb{Z} \); all others are \( \{0\} \).

These are just a few of the theorems that can be discovered easily by an inquiring student.

VI. Interface with Graphics.

It is worthwhile to mention a simple experimental interface of the HOMOLOGY workspace with graphics workspaces suitable for use with a Tektronix 4013 or 4015 terminal. This allows interactive graphics input of simplicial complexes, where the user sketches a triangulation on the screen and the groups are computed automatically.

For example, below is the sample input from the graphical homology program for the Moebius strip.

![Graphical Homology Program Input](image)

\[
\begin{align*}
H_0(\text{MSTRIP}) &= \mathbb{Z} \\
H_1(\text{MSTRIP}) &= \mathbb{Z} \\
H_2(\text{MSTRIP}) &= 0
\end{align*}
\]

VII. Topics for Further Development.

The HOMOLOGY workspace has been used successfully in conjunction with an elementary course in algebraic topology at the University of California, Berkeley. However, there are many ways in which the workspace could be improved. We list just a few:

A. Rewrite the functions that compute with incidence matrices to use sparse representation methods.

B. Develop good interactive methods for input and display of higher-dimensional complexes on graphics terminals.

C. Find upper bounds on the computation time and space needed to determine homology groups based on the number of simplexes.

VIII. Acknowledgements.

The author acknowledges with thanks conversations with John Hughes.

IX. References.


Appendix: Function Listings

A. Functions to Compute and Display Homology Groups.

BBASE: REDUCE (q+1) INCID w
CHANGE: \(-/0\) DIM w APPLY 'VPP VFP ',w = RULED-POINCARE CHARACTERISTIC
CHOM: \(0,1\) DIM w APPLY 'CHOM ',w = COMPUTE HOMOLOGY PLEASE
CHOM: (\(0,1\) DIM w) APPLY 'CHOM ',w = COMPUTE HOMOLOGY PLEASE

B. Functions to Convert Representations.

BFV: +/*[2] \(=0\) B-REP FROM V-REP
BSUB: \(=0\) BOOLEAN MATRIX REPRESENTING ALL NON-NUL SUBSETS
SORT: \(=0\) SORT SUBSETS OF \(=0\), PADDED WITH 0'S
VFB: RTF (\(=0\) V-REP FROM P-REP

C. Functions to Generate and Join Objects.

K: \((1,1,1)\) COMPLETE GRAPH ON \(w+1\) POINTS, P-REP
SPHERE: \(=0\) PRINCIPAL SIMPLICES FOR N-SPHERE
CS: \(=0\) DISJOINT UNION
ECF: \((P,J)\) ELEMENTARY CART PROD

D. Utility Functions.

APPF: \((1,1)\) APPLY \(=0\) MAT w = CATEenate TWO ARRAYS, TREATED AS MATRICES
COL: \((\text{MAT } w)\) JOIN MAT w = CATEGorize TWO ARRAYS, TREATED AS MATRICES
CONV: \(=0\) CONV TO COLUMN
MAT: \((\text{MAT } w)\) ADD ROW AND COLUMN TO MATRIX w
FILL: \(=0\) FILL CHARACTER FOR w
ID: \(=0\) SQUARE IDENTITI MATRIX
INDEX: \((\text{MAT } w)\) INDEXES (\(=0\) w)
JOIN: \(=0\) JOIN TWO MATRICES
MAP: \((\text{MAT } w)\) PAIRS (\(=0\) w) = PAIR GENERATOR
MERGE: \((\text{MAT } w)\) REMOVE DUPLICATE ROWS
REDUCE: \((\text{MAT } w)\) REDUCE TO DIAGONAL FORM
UMB: \((\text{MAT w})\) ID NUMBERS IN V-REP FROM P-REP
REDUCE: \((\text{MAT w})\) REDUCE TO DIAGONAL FORM
E. Variables

TORUS

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>

KLEIN

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>

MOEBIUS

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>

CYL

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>