

Ext in the nineties

ROBERT R. BRUNER

ABSTRACT. We describe a package of programs to calculate minimal resolutions, chain maps, and null homotopies in the category of modules over a connected algebra over Z_2 and in the category of unstable modules over the mod 2 Steenrod algebra. They are available for free distribution and intended for use as an Adams spectral sequence 'pocket calculator'. We provide a sample of the results obtained from them.

1. Introduction

Ext is a collection of programs to calculate $\text{Ext}(M) = \text{Ext}_A(M, Z_2)$, through a range of dimensions, where A is a connected augmented algebra over Z_2 and M is a bounded below A -module. The motivation is to provide the E_2 term of the Adams spectral sequence for the homotopy of the space or spectrum whose cohomology is M . To this end, the package includes the routines needed when A is the Steenrod algebra. It also contains, as samples, the routines needed when M is the trivial module, Z_2 , or the cohomology of a stunted projective space, and it is relatively easy to add the routines needed for a new module M . There is also a version which computes Ext in the category of unstable modules over the Steenrod algebra, for use in calculating the unstable Adams spectral sequence.

Routines are included which produce Postscript (TM) displays of the results as well as simple listings of elements and relations in tabular form. Thus, the user can send the resulting files to a laser printer or, on systems which provide on-screen display of Postscript, directly to the console screen. These display programs are also configurable, so that by default they show the conventional charts which display multiplication by h_0 and h_1 , but can easily be made to display other multiplications or Massey products (e.g., periodicity operators). An interactive program to examine the results is also provided.

1991 *Mathematics Subject Classification*. Primary 55-04, 55T15, 16-04, 18G10, 18G15; Secondary 55Q40, 55Q45, 55Q52 .

The final version of this paper will be submitted for publication elsewhere.

This work was supported in part by NSF Grant #DMS-8601776.

©0000 American Mathematical Society
0000-0000/00 \$1.00 + \$.25 per page

The program has been used to calculate $\text{Ext}(Z_2)$ through internal degree 116, including the complete product structure and a substantial number of Massey products through degree 90. The Massey products include the periodicity operators wherever they are defined, not only in the range in which they are isomorphisms. We include here the traditional chart of this calculation showing the products by h_0 and h_1 . Note that it shows a number of products that were hidden by filtration shifts in previous calculations, such as the fact that $h_0^5 B_4$ in the 60 stem is nontrivial. (This is an example of the tendency for results to be discovered independently but simultaneously: Tangora found this extension at almost the same time.) The entire product structure contains an enormous number of relations that is too massive to include here. However, we include as a sample of these relations the complete description of the 60-stem, since this contains a number of previously unknown relations. Our calculations confirm the unpublished calculations due to Tangora in the stems between 70 and 80, and include numerous relations which are not visible in the May spectral sequence. We have also used the program to calculate stable Ext for stunted projective spaces, for the mod 2 Moore space, and for the cofibers of η and ν . We have calculated unstable Ext for S^3 , S^7 , and for the cofiber of $\nu : S^{10} \rightarrow S^7$. In the spirit of [CM88] we include here the charts for unstable Ext of S^3 and for stable Ext of the Moore space.

2. Implementation and Techniques

The technique is to calculate a minimal resolution for M , chain maps between resolutions, and null homotopies of composites of chain maps. As described in [RRB] the resolution yields the additive structure and products by elements of $\text{Ext}^1(Z_2)$. The chain map induced by an element $b \in \text{Ext}(M)$ yields products ab and Massey products $\langle h, a, b \rangle$, for all $a \in \text{Ext}(Z_2)$ and $h \in \text{Ext}^1(Z_2)$. A null homotopy of the composite of the chain maps induced by elements $b \in \text{Ext}(Z_2)$ and $c \in \text{Ext}(M)$ yields Massey products $\langle a, b, c \rangle$, and $\langle h, a, b, c \rangle$ for all $a \in \text{Ext}(Z_2)$ and $h \in \text{Ext}^1(Z_2)$. The four fold Massey products are not mentioned in [RRB], but follow by the techniques there. While the algorithms are the same as those presented in [RRB], the implementation is entirely new.

This is the third such package of programs that I have written, and is specifically designed to be portable, so that any machine with a C compiler can run it. Anyone who wants a copy may obtain it by electronic mail addressed to rrb@math.wayne.edu. Those without access to Internet mail can write to the author at Wayne State University to find suitable means of distribution.

The package as it is mailed is about 90 K bytes. When unpacked, it occupies about 200 K. After compilation and creation of the necessary data files for the first module M , about 800 K is required, and each additional module requires another 160 K to start with. For the modules I have calculated thus far, the essential data files defining the resolution occupy about 5 K for $t \leq 30$, and

about 15 K for $t \leq 45$. For $M = Z_2$ and $t \leq 90$ they occupy about 400 K. If intermediate data files are saved from the calculation of the resolution for use in calculating chain maps and null homotopies, the storage requirements go up considerably. For example, about 1.1 M bytes are used for $t \leq 35$, 1.6 M for $t \leq 45$, and 200 M for $t \leq 90$. Still, on modern machines, these requirements are quite modest.

The package as currently distributed is intended for use on BSD Unix systems. On such systems, the various tasks necessary to initiate the calculations for a new module M are encapsulated in a script called `Install`, so that the user need only create the files which define the module and invoke the script. On other systems, the script can be used as a guide to the necessary steps.

There are also two versions of the package, one for spectra and one for (nice) spaces. The package for spectra is the more general in the sense that it simply calculates minimal resolutions (hence `Ext` and `Tor`) in the category of modules over a connected augmented algebra. The package for spaces computes minimal resolutions in the category of unstable modules over the *Steenrod* algebra.

The package is designed to allow networked computers to work in parallel as well. The granularity is quite coarse, but the speedup is still close to optimum: the time for N machines to complete the task being roughly $1/N$ of the time required by a single machine. Each internal degree, t , is assigned to a specific machine. When it completes bidegree $(s-1, t)$, it waits until bidegree $(s, t-1)$ has been completed before proceeding to bidegree (s, t) . It is immaterial which machine is calculating internal degree $t-1$: it could be the same machine or another one. Thus, even with only one machine available, it probably makes sense to start that machine on the calculations for several internal degrees simultaneously. The process calculating internal degree t will simply wait for the process calculating internal degree $t-1$ to finish with filtration s before it proceeds to filtration s . This will take full advantage of the machine's resources, since some parts of the calculation are IO bound and some are compute bound. With several machines of similar speed, it is best to apportion the internal degrees round robin. Thus with 4 machines available, to calculate internal degrees 20 to 39, I would assign degrees 20, 24, 28, 32, and 36 to the first machine, degrees 21, 25, 29, 33 and 37 to the second, etc.

Experience with computer algebra shows that bugs and other errors are hard to avoid. To minimize the possibility that any such problems will go undetected, each bidegree of the calculation is tested for various forms of consistency. First, the data files are examined to see that they exist and have the expected format, to catch gross errors due to lack of space or various operating system failures. Second, the equality $\dim(\text{Im}(d_{s,t})) = \dim(\text{Ker}(d_{s-1,t}))$ is checked. This is a crude test for exactness. The additional test that $d^2 = 0$ is too slow to use routinely. Third, the equality $\dim(\text{Im}(d_{s,t})) + \dim(\text{Ker}(d_{s,t})) = \dim(C_{s,t})$ is checked, where the dimension of $C_{s,t}$ is calculated as the sum of $\dim(A_{t-\deg(g)})$ over all A generators $g \in C_s$. Finally, linear independence of the calculated

bases of $\text{Im}_{s,t}$ and $\text{Ker}_{s,t}$ is tested. These four checks can be made very quickly and, if passed, suggest some confidence may reasonably be placed in the calculations. So far, the second test was failed once. When that happened, I ran the more complete test to check $d^2 = 0$ and from its results was able to trace the error to a bug in an early version of the routine which multiplies Milnor basis elements. The first tests are failed irregularly, and indicate the need for the user to intervene, typically by reducing the number of processes or by finding more disk storage. They serve to prevent the program from writing incorrect results in these circumstances, permitting graceful continuation after correcting the problem.

3. History

The first program of this sort was done by Liulevicius [Liu] in 1964. Using an IBM 7094 with the equivalent of 294 K bytes of memory, he was able to reach internal degree 32. After that, I am not aware of any attempts to automate the calculation of minimal resolutions for non-commutative algebras until 1983, when I wrote one in LISP. It was effective in low dimensions, but spent nearly all its time in garbage collection by the time it reached internal degree 40.

In 1984 and 1985 I rewrote that program in a form designed for maximal efficiency. To that end, the programs were written in Fortran and Assembler, with the Assembler being used for bit manipulation and for routines that were called most frequently. Quite a few changes were made at this stage. One was a switch from the admissible basis to the Milnor basis for the Steenrod algebra, eliminating the need for recursive application of the Adem relations in the multiplication routine. Another was the replacement of automatic memory allocation and garbage collection, as is done in LISP, by explicit allocation and deallocation of memory. Of major significance in memory usage was a switch from symbolic representation of elements by characters and numbers to their representation by bitstrings. For example, in the LISP version, the element $Sq^7 + Sq^{(4,1)}$ of the Steenrod algebra would have been represented by the list of lists of numbers $((7) (4 1))$. In the bitstring version, $Sq^7 + Sq^{(4,1)}$ could be represented by the 4 bit string 1100, since the Steenrod algebra is 4 dimensional in degree 7. This switch also resulted in vastly less effort in adding elements, since the earlier version required an enormous amount of sorting and merging of lists, while the bit string representation simply requires that blocks of memory be exclusive or'ed together. Profiling of this version showed that more than 50 percent of the program's time was spent in the process of adding vectors, so that this is a significant issue. This version was able to reach internal degree 69 of $\text{Ext}(Z_2)$, at which point I stopped because the program falsely claimed that there were two elements in bidegree $(4,69)$. This is easily checked to be false using the May spectral sequence. At about that time I moved from the University of Illinois at Chicago to Wayne State University. The move showed a major fault in the

notion of writing programs so carefully tailored to the hardware and operating system. The effort of adapting the programs to run at Wayne under a different operating system with a different file structure wasted several months of effort. This was in the days before easy access to the Internet made remote logins and file transfer a triviality. In addition, limitations on funding and availability of computer time made it impossible to reach the same dimension again in order to determine the source of the error that had occurred at UIC. However, I was able to make some interesting calculations related to $MO(8)$ and to provide actual minimal resolutions for stunted projective spaces, with the version running at Wayne.

Having discovered the amount of effort required to move the package from one system to another, even though it was based on the same hardware, it was clear that portability was a significant factor. Since it is unlikely that great breakthroughs in homotopy theory will result from mechanical calculations of the E_2 term of the Adams spectral sequence, another important feature of portability is the ability to distribute the package to other algebraic topologists for use as a sort of Adams spectral sequence ‘pocket calculator’. Hence the current version was written entirely in a high level language, C, and treats all its data files as streams of characters, so that no special assumptions need to be made about the types of file access available. In fact, it was initially developed on an MS-DOS machine, an 8 MHz AT clone with 640 K of memory. With this hardware, the programs reached internal degree 40 in computing $\text{Ext}(Z_2)$. Further development, namely the routines for chain maps and null homotopies, and the routines to report the results in Postscript format, was done in a Unix environment on a Sun workstation. The programs are sufficiently generic in their design that they should run with no difficulty on any Unix system. On a non-Unix system, a number of tasks that are performed by scripts in the Unix environment would have to be done by hand, or the scripts would have to be rewritten to work under the operating system in question. Since the scripts are quite elementary, this should not be difficult.

Despite the attention to portability in place of efficiency, this version of the program has succeeded in significantly surpassing previous calculations. Perhaps there is a moral here. The current calculations of $\text{Ext}(Z_2)$ were originally stopped at internal degree 90 so that all the multiplicative structure up to that degree could be calculated. The point is that, in the process of finding the minimal resolution, a Z_2 basis for the image and kernel of each differential is computed. If the bases for the images are saved, computing chain maps and null-homotopies is very fast. If they are discarded, the bases have to be recomputed in order to compute chain maps or null-homotopies. Thus, it makes sense to compute all the multiplicative structure that is wanted before discarding the image files. However, at degree 90, I had come to occupy as much of our department’s file space as I could fairly grab, so stopped there for the time being.

During preparation of the final version of this paper, I decided to continue

the calculation without saving the image files, and had enough computer time to calculate out to internal degree 116. In internal degree 116, the largest bidegree, $(5,116)$, has dimension 22465 over Z_2 and took about 16 hours to process on a Sparc 2 with 32 megabytes of memory.

Other approaches to machine calculation of Ext have been taken by other authors over the years. Most notably, Tangora's lambda algebra calculations [MCT], his more recent machine calculations of the May spectral sequence (as of yet unpublished), and the lambda algebra calculations done by Curtis, Goerss, Mahowald, and Milgram in various combinations ([CGMM], [CM86], [CM88], among others). Each approach has its advantages and disadvantages. The May spectral sequence has reached the highest degrees, but is inherently unable to give a complete picture of the product structure. The lambda algebra approach has the great virtue that the unstable calculations are embedded within the stable calculation. With our approach to calculating unstable Ext, each sphere requires a distinct calculation. However, the lambda algebra programs currently available seem to have reached their limit at about $t = 80$. In addition, retrieving the products and Massey products from the lambda algebra calculations is sometimes considerably more difficult than with minimal resolutions. Finally, there are applications in which a minimal resolution plays an important role [DM] [Davis]. Our program is the ideal tool for such applications.

4. Results

The first task in comparing the chart of $\text{Ext}(Z_2)$ presented here with earlier calculations, is to determine which differences are real and which are simply a result of different choices of basis. An example of a real difference occurs in the 60 stem, where we show $h_0^5 B_4 \neq 0$, where earlier calculations had only an associated graded in which $h_0^3 B_4 = 0$

The first instance of an apparent, but unreal, difference which is visible in the charts is in filtration 9 of the 23-stem. We clearly have $h_0^2 i$, as expected, but the h_1 multiple, $h_1 P^1 d_0$ appears to be the sum of the two terms in this bidegree. In fact, with the basis for $\text{Ext}^{9,32}$ which emerges from the program, this is correct: the machine has found the elements $h_0^2 i$ and $h_0^2 i + h_1 P^1 d_0$ as *its* basis for $\text{Ext}^{9,32}$. When all the elements in the bidegree are decomposable, the product structure allows us to sort out the situation. For example, in $\text{Ext}^{9,32}$ there is only one h_0 multiple, and there is only one h_1 multiple, so it is clear how the elements produced by the program relate to the elements derived from the May spectral sequence or the lambda algebra or the cobar construction.

A serious possibility of confusion arises when an indecomposable occurs in a bidegree which has more than one nonzero element. For example, in filtration 4 of the 18-stem, we have $h_1^3 h_4$ and an indecomposable, f_0 . The traditional definition of f_0 is by its representative in the May spectral sequence, $h_2^2 b_{03}$, or the Massey product $\langle h_0^2, h_3^2, h_2 \rangle$. We can compute the Massey product,

but its indeterminacy is nonzero, so this is no help. The best definition would be in terms of the Hopf algebra squaring operations: $f_0 = Sq_1 c_0$. However, the mechanical computation of the squaring operations is still out of reach in general (see the section on coproducts and Steenrod operations below). In the absence of a resolution of such a question, we can simply proceed with the calculations until we find a question whose answer depends upon whether the machine's generator is f_0 or $f_0 + h_1^3 h_4$, at which point we will hopefully find that the question itself resolves the issue.

The next such question arises in filtration 4 of the 38-stem. The generators chosen by the program both support a nontrivial h_0 . This one is easily resolved by the techniques of [HRS]. If we define e_1 as $Sq_0 e_0$ (as we should) then e_1 is carried on the 21-cell of the stunted projective space P_{17}^∞ . The 22-cell, which also maps into filtration 4, implies that $2e_1 = 0$ in filtration 4 modulo filtration 6. This forces $h_0 e_1 = 0$. Hence, e_1 is the sum of the two generators found by the program. Alternatively, the null homotopy of $h_1^2 h_4^2$, applied to each of the generators, contains the term $Sq^2 h_2$, and therefore the Massey product $\langle h_1, h_2, h_1^2, h_4^2 \rangle$ defining e_1 , is the sum of those two generators. The element $h_0^2 h_3 h_5$ is easily identified with the second of the two generators found by the program, since it is the only element divisible by h_0 .

Next, in the 40-stem, we have the question of identifying the second element of the 'f-family', $f_1 = Sq_0 f_0 \in \text{Ext}^{4,44}$. The program produces generators which are $h_2^3 h_5$ and either f_1 or $f_1 + h_2^3 h_5$. The disposition of this case is presumably similar to that of f_0 . We will write \tilde{f}_0 and \tilde{f}_1 to avoid prejudging this question.

5. The 60 stem of $\text{Ext}_A(Z_2, Z_2)$

Here is the systematic description of the 60-stem in $\text{Ext}(Z_2)$, listed by filtration. All nonzero products are listed, except those which follow from a product which is listed by multiplication by some power of h_0 . In most cases, only one of a set of related Massey products is listed. For example, from $B_3 = \langle h_4, h_0 h_3, x \rangle$ it is easy to derive $\langle h_4, h_0 x, h_3 \rangle$. In fact, from $h_0^2 g_2 = h_3 x$, both of these can be derived from $\langle h_4, h_0^2 g_2, h_0 \rangle$. However, since there is no complete account of the product structure of Ext in the literature, we have chosen not to omit brackets related by relations like $h_0^2 g_2 = h_3 x$ or $h_3 r = h_0^2 x + h_1 t$. We have omitted all brackets of the form $\langle h_i, x, h_i \rangle$, since they contain $h_{i+1} x$ by the relation $y(x \cup_1 x) \in \langle x, y, x \rangle$. Except for the periodicity operators computed from null homotopies of $h_0^4 h_3$ and $h_0^8 h_4$, the information here is derived entirely from chain maps induced by indecomposables. Thus, all the brackets have the form $\langle h_i, x, y \rangle$ with y indecomposable. Finally, this table carries with it the implicit claim that any such Massey product which cannot be derived from those listed is zero.

$$\begin{aligned}
7 \quad B_3 &= \langle h_4, h_0^2 g_2, h_0 \rangle = \langle h_4, h_0 h_3, x \rangle = \langle h_0, h_1, D_2 \rangle \\
8 \quad h_0 B_3 &= h_2 Q_2 = h_5 k = \langle h_4, h_3^2, r \rangle = \langle h_4, h_0^2 h_4^2, d_0 \rangle \\
9 \quad B_4 & \\
\quad h_0^2 B_3 &= h_2 h_5 j = P^1 D_1 = \langle h_3, h_0 c_2, P^1 h_2 \rangle = \langle h_1, h_0^3, D_2 \rangle \\
&= \langle h_1, h_1 h_5 c_0, \tilde{f}_0 \rangle \\
10 \quad h_0 B_4 & \\
11 \quad h_0^2 B_4 &= \langle h_2, p, i \rangle \\
\quad h_1 B_{21} &= d_0 B_1 = \langle h_5, h_0^2 P^1 e_0, h_2 \rangle = \langle h_4, \tilde{f}_0, j \rangle = \langle h_4, j, \tilde{f}_0 \rangle \\
&= \langle h_4, z, h_2 \rangle = \langle h_2, h_0 B_2, c_0 \rangle \\
&\in \langle h_0, h_2 B_2, c_0 \rangle \\
&\text{relation } h_3 x' = h_0^2 B_4 + h_1 B_{21} = \langle h_4, h_0 d_0, r \rangle \\
12 \quad h_0^3 B_4 &= h_0 h_3 x' = i x = P^2 g_2 = \langle h_5, d_0^2, h_0^4 \rangle \in \langle h_2, h_2^2 g, r \rangle \\
\quad g^3 &= \langle h_4, P^1 h_1, m \rangle = \langle h_2, h_1 g, m \rangle = \langle h_1, h_2 g, m \rangle \\
&\in \langle h_3, h_0^2 h_3, w \rangle \\
&\text{relation } r^2 = g^3 + h_0^3 B_4 \\
13 \quad h_0^4 B_4 &= h_0 r^2 = \langle h_2, h_1 P^1 h_1, B_1 \rangle \\
14 \quad h_0^5 B_4 &= h_3 r i = h_3 P^2 x = \langle h_2, h_0^4, Q_1 \rangle = \langle h_2, h_0 r, j \rangle \\
15 \quad d_0^2 l &= d_0 e_0 k = d_0 g j = e_0^2 j = g i e_0 = m P^1 e_0 = \langle h_2, h_1 d_0, z \rangle \\
&= \langle h_2, h_0 P^1 h_2, w \rangle = \langle h_2, h_0^2 e_0, u \rangle = \langle h_2, h_0^2 j, r \rangle \\
&= \langle h_1, h_0 e_0, z \rangle = \langle h_1, h_0 d_0^2, r \rangle = \langle h_0, h_0 \tilde{f}_0, z \rangle \\
&= \langle h_0, h_0^2 d_0, w \rangle = \langle h_0, h_0^2 g, u \rangle = \langle h_0, h_0^2 k, r \rangle \\
18 \quad d_0 i^2 &= j P^1 j = r P^2 d_0 = \langle h_3, h_0^4 i, k \rangle = \langle h_2, P^2 c_0, l \rangle \\
&= \langle h_2, h_0^2 d_0 P^1 d_0, g \rangle = \langle h_2, h_0^2 d_0 P^1 e_0, e_0 \rangle \\
&= \langle h_1, h_1 P^1 c_0, z \rangle = \langle h_1, P^2 h_2, u \rangle = \langle h_0, P^2 h_1, v \rangle \\
&= \langle h_0, P^2 c_0, m \rangle = \langle h_0, h_0^5 r, k \rangle = \langle h_0, h_0^2 d_0 P^1 e_0, g \rangle \\
24 \quad d_0 P^4 d_0 &= \langle h_2, P^4 h_1, i \rangle = \langle h_1, h_1 P^1 c_0, P^3 e_0 \rangle = \langle h_1, P^4 h_2, i \rangle \\
&= \langle h_0, P^4 h_1, j \rangle = \langle h_0, h_0^5 P^2 i, g \rangle \\
25 \quad h_0 d_0 P^4 d_0 &= h_2 P^5 e_0 = e_0 P^5 h_2 \\
26 \quad h_0^2 d_0 P^4 d_0 &= h_2^2 P^5 d_0 = h_2 d_0 P^5 h_2 = \langle h_1, h_0^2 P^4 h_2, i \rangle
\end{aligned}$$

6. Coproducts and Steenrod operations

Chain maps aren't the only way to compute products when A is a Hopf algebra. We could also compute the diagonal, $C \rightarrow C \otimes C$. This would have the added benefit that it would be relatively easy to then compute cup- i products to obtain the action of the Steenrod algebra on Ext . In the absence of a Yoneda type description of the cup- i products, computing the diagonal map appears to be the only mechanical way of computing the Steenrod operations. Thus, despite a gut feeling that the effort was doomed to failure because of the enormous size of $C \otimes C$, I wrote the routines needed to compute the diagonal and set them to work computing the diagonal for $t \leq 60$. The results are summarized in Table 1. The terms column tells the number of terms in the coproduct when it is written as the sum of terms $x \otimes y$ with the x 's linearly independent monomials. Computing the coproduct of $h_0 h_4^3 = h_0 h_3^2 h_5$ was quite discouraging. The problem is that $h_0 h_4^3$

TABLE 1. Coproducts for $t \leq 60$, $s \leq 4$

generator(s)	time	size (bytes)	terms
filtration 1 (6 generators)	19 secs	2.5 sec	—
filtration 2 (15 generators)	1047 secs	287 K	—
filtration 3 (21 generators)	54534 secs	2370 K	—
filtration 4 (1 st 20 generators)	139975 secs = 38.9 hrs	3399 K	—
$h_2^3 h_5 = h_1^2 h_3 h_5$	34962 sec (9.7 hours)	668 K	12584
$h_0 c_2$	59658 secs (16.6 hours)	744 K	13893
g_2	74587 secs (20.7 hours)	961 K	17842
$h_0 h_4^3 = h_0 h_3^2 h_5$	1285683 secs (14.9 DAYS)	4261 K	19865
$h_3 c_2$	399092 secs (4.6 days)	1444 K	25087
$h_5 c_1$	715938 secs (8.3 days)	1713 K	28926

is highly decomposable, and the diagonal contains every possible decomposition. Thus, the coproduct, naively computed, is virtually useless, both on time and storage grounds. In comparison, the chain maps are stunningly efficient: we only need to know the chain maps induced by indecomposable elements, and we only need their values on A -generators of the resolution. Unfortunately, this leaves us with no easy means of finding the Steenrod operations in Ext in order to decide, for example, which of the two possibilities in $\text{Ext}^{4,22}$ is f_0 .

Occasionally, we can compute the cup-1 product using the fact that

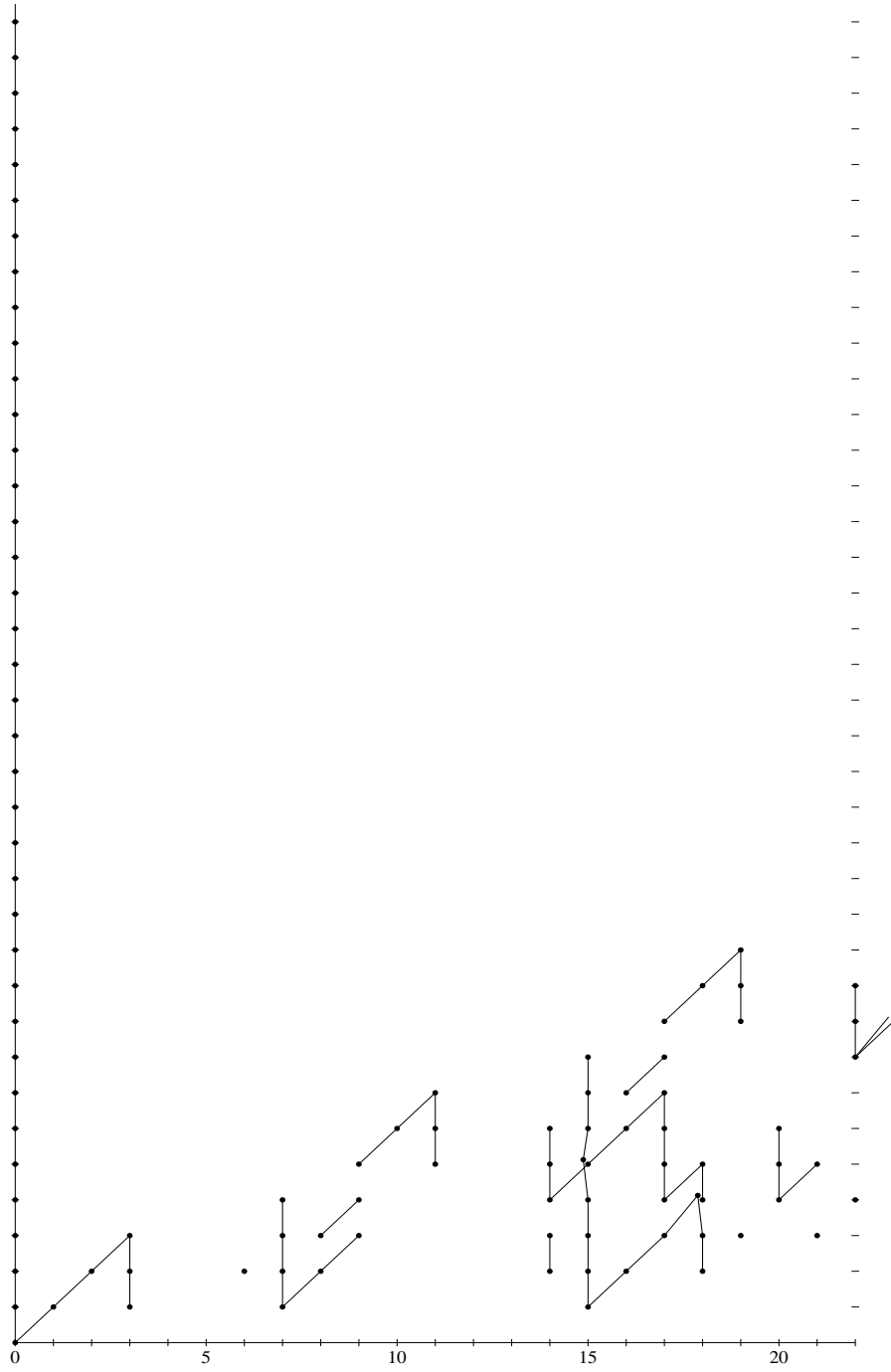
$$y(x \cup_1 x) \in \langle x, y, x \rangle .$$

All we have to do is find an element y which annihilates x and which acts monomorphically on the bidegree containing $x \cup_1 x$.

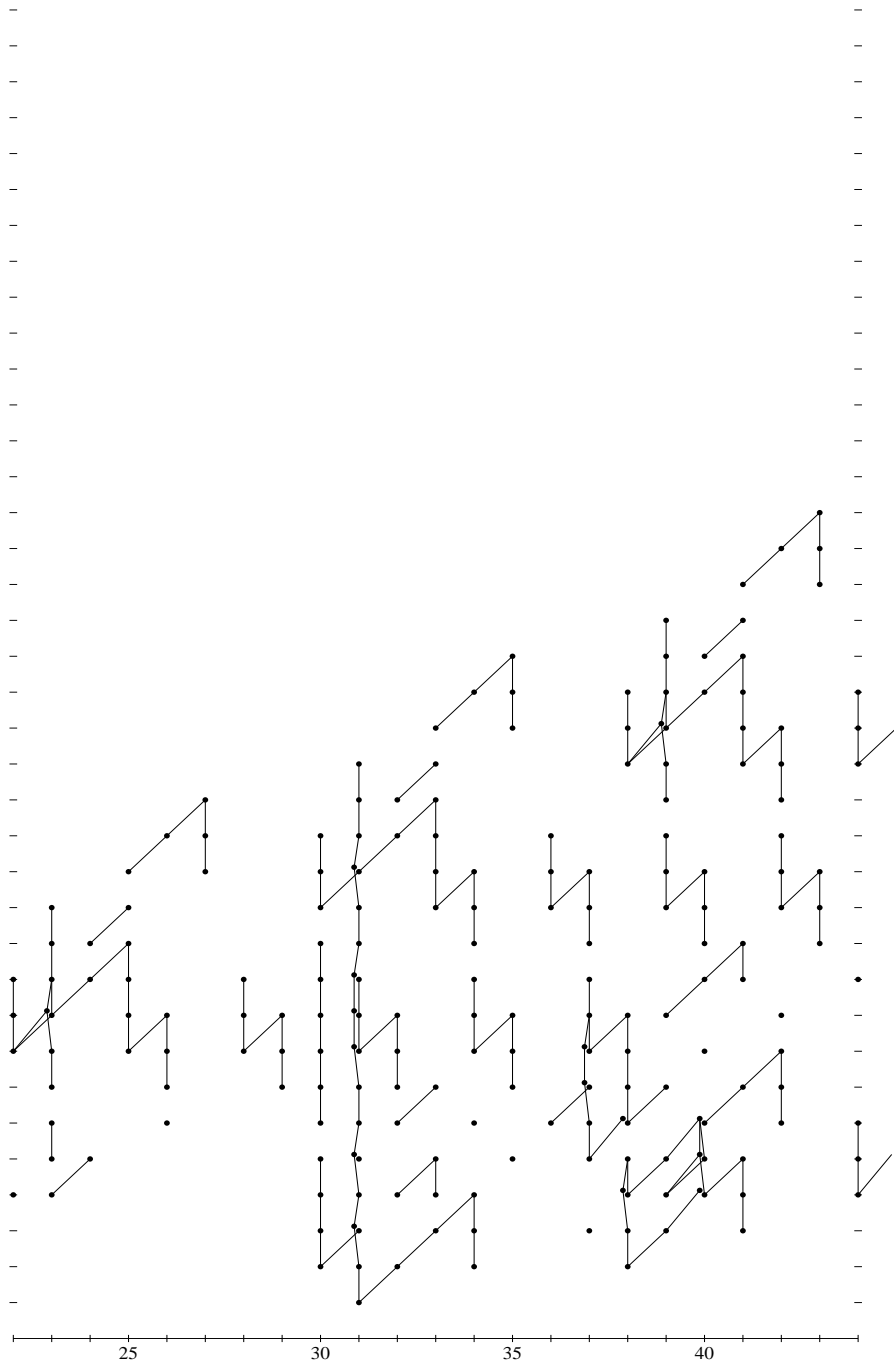
7. Charts

The first set of charts consists of Ext for the stable sphere S^0 for $t - s \leq 88$, $t \leq 116$, and $s < 38$. The next set of charts consists of stable Ext of the Moore space $S^0 \cup_2 e^1$ for $t - s \leq 44$. The last set consists of unstable Ext of S^3 for $t - s \leq 91$ and $t \leq 100$. Comparison of the last 2 sets quite clearly shows the isomorphism discussed in [CM88]. In the first set, due to the density of ink, only h_0 and h_1 multiples are indicated. In the last two sets, h_2 multiples are also shown.

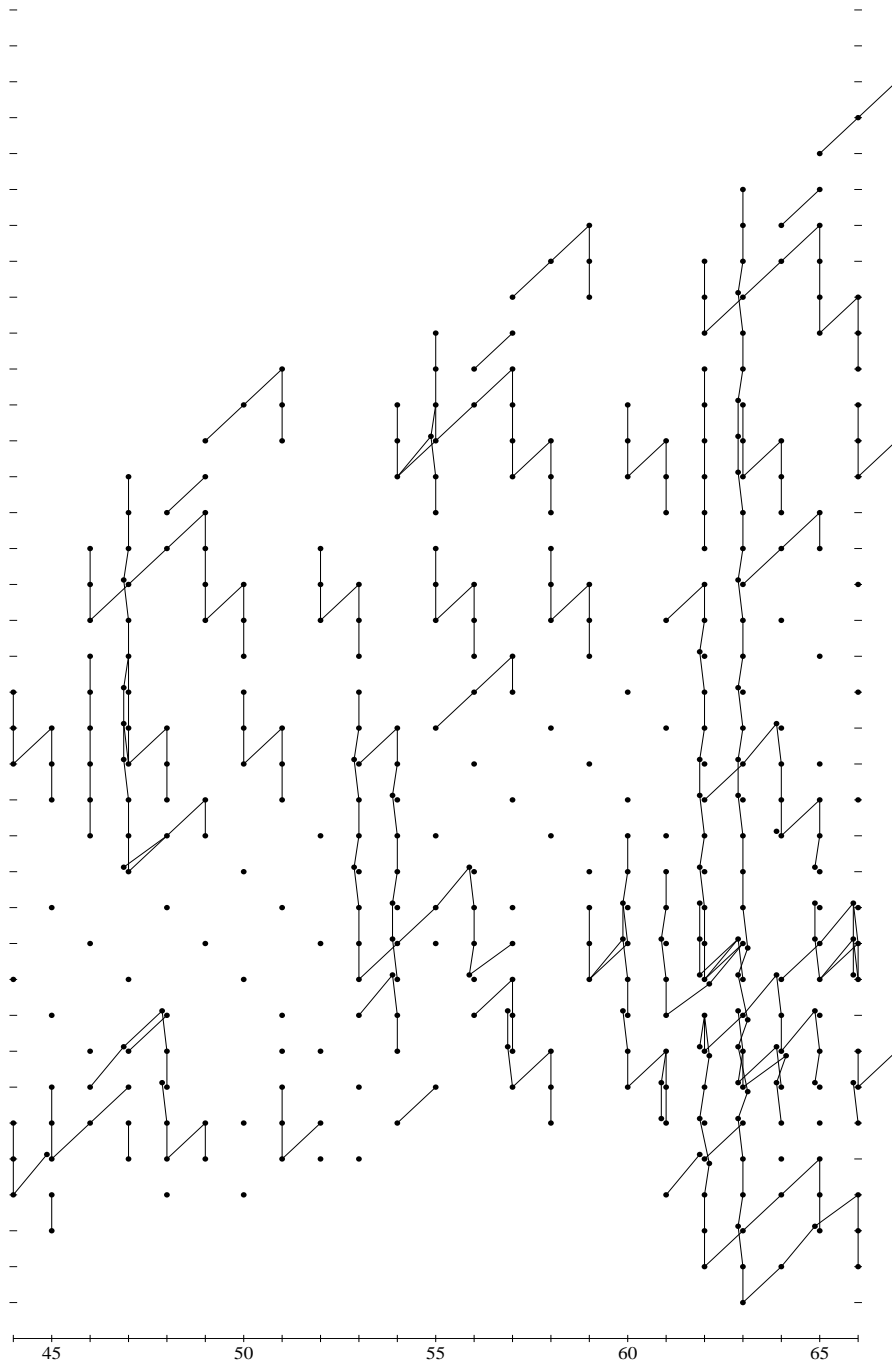
The reader is encouraged to xerox these charts so they may be pasted together to obtain a continuous chart for each Ext. The tick marks along the left and right edges should be aligned atop one another.



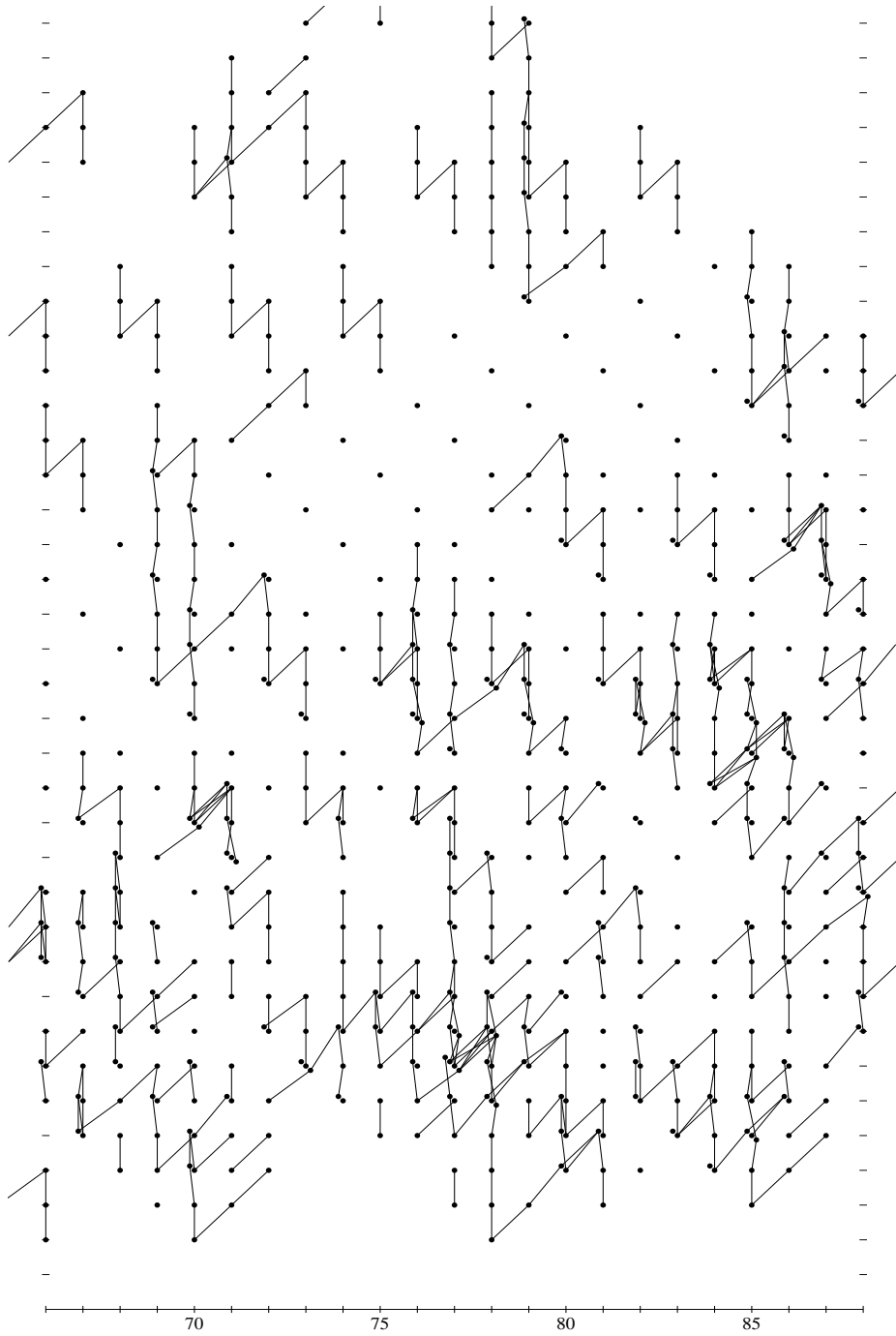
$\text{Ext}_A^{st}(Z_2, Z_2), \quad 0 \leq t - s \leq 22$



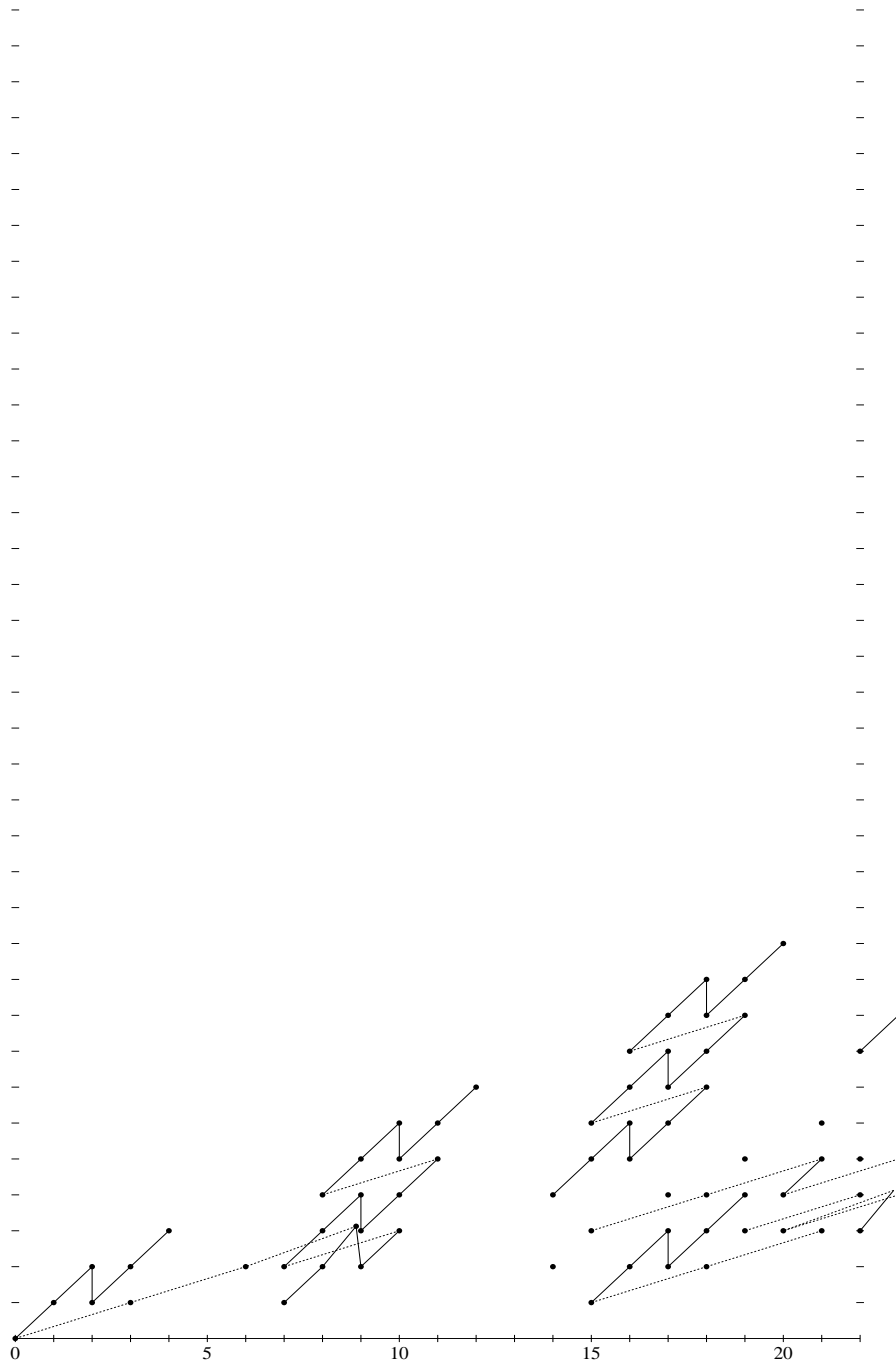
$\text{Ext}_A^{st}(Z_2, Z_2), 22 \leq t - s \leq 44$



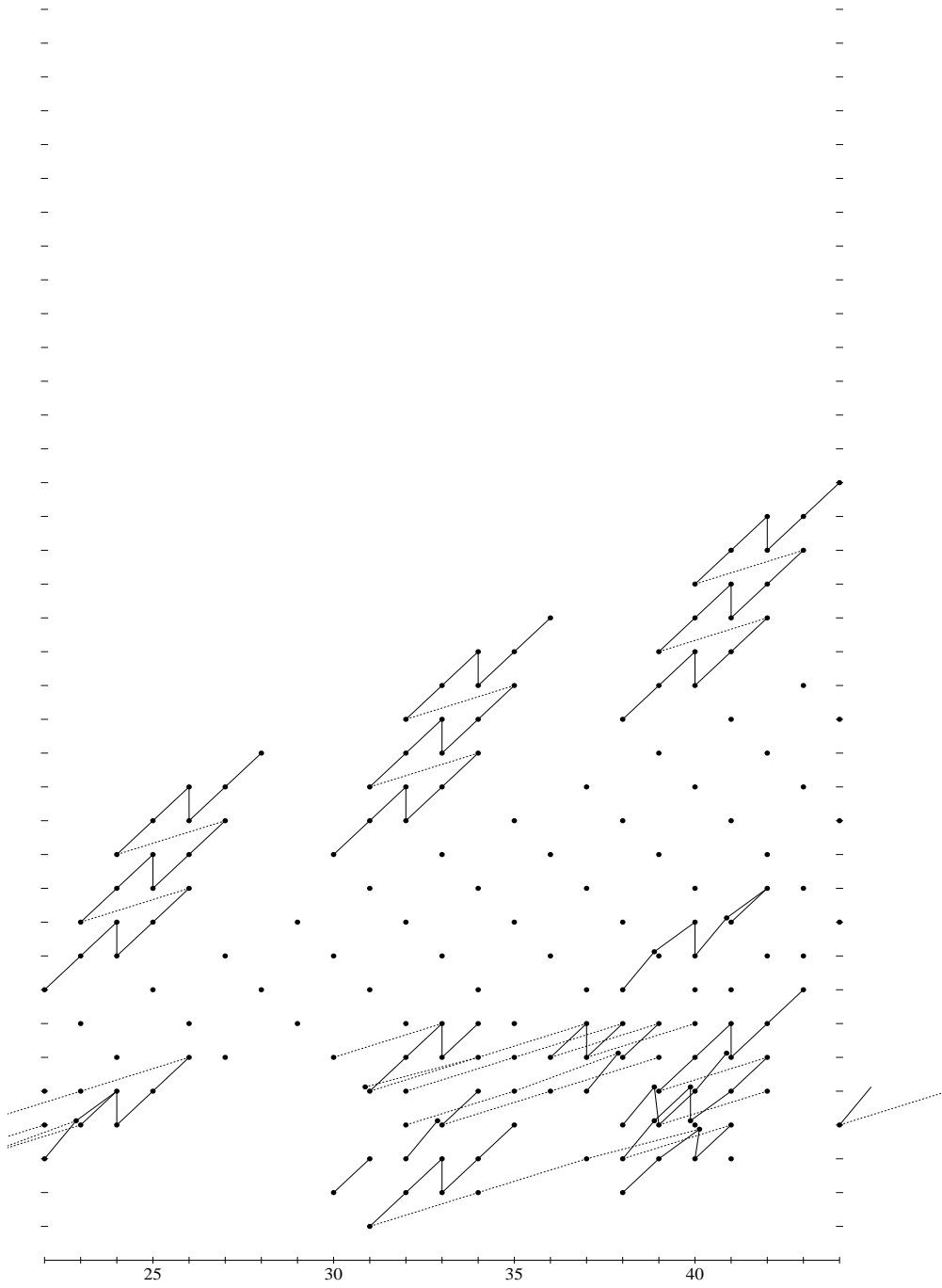
$$\text{Ext}_A^{st}(Z_2, Z_2), \quad 44 \leq t - s \leq 66$$



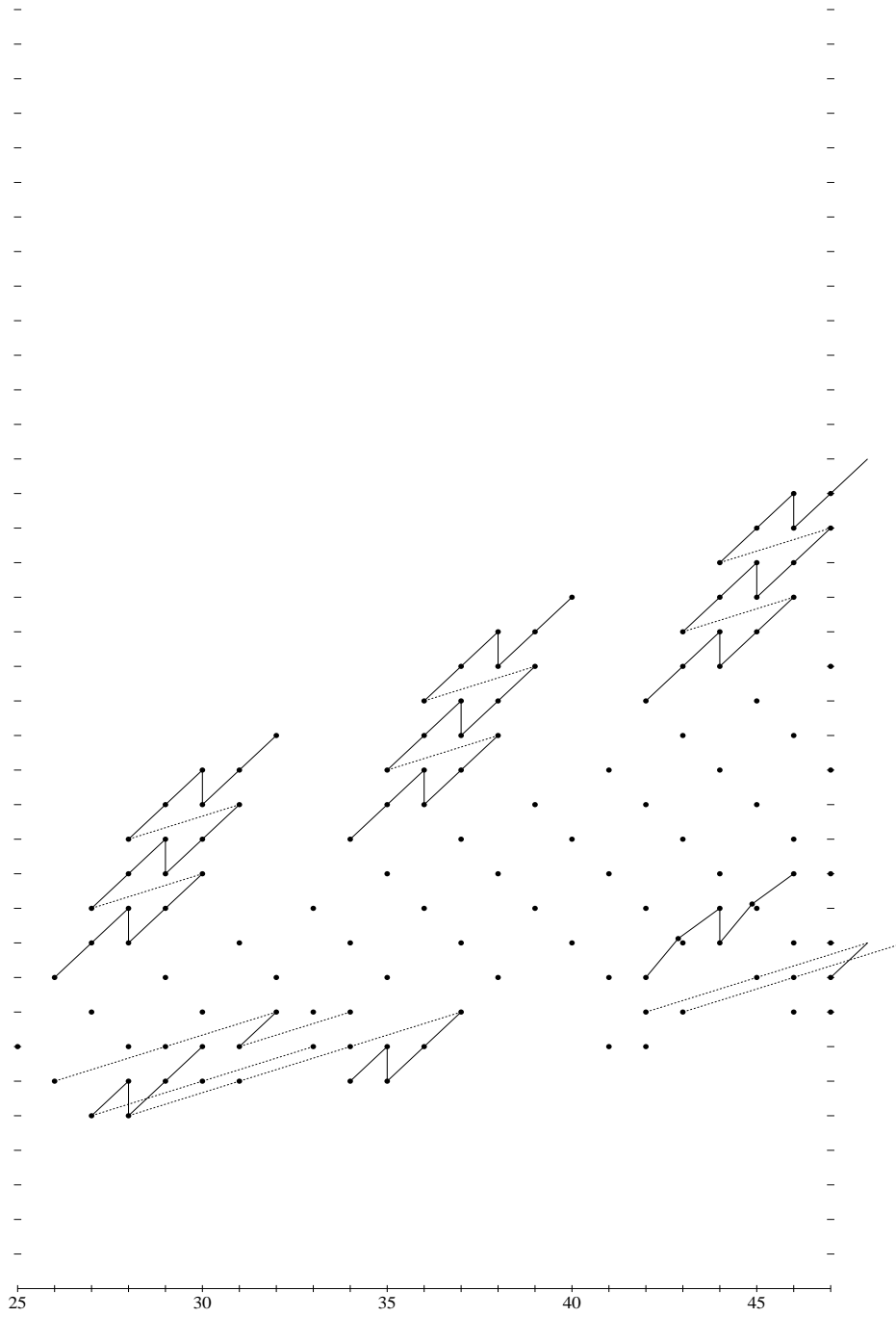
$$\text{Ext}_A^{st}(Z_2, Z_2), \quad 66 \leq t - s \leq 88, \quad t \leq 116$$



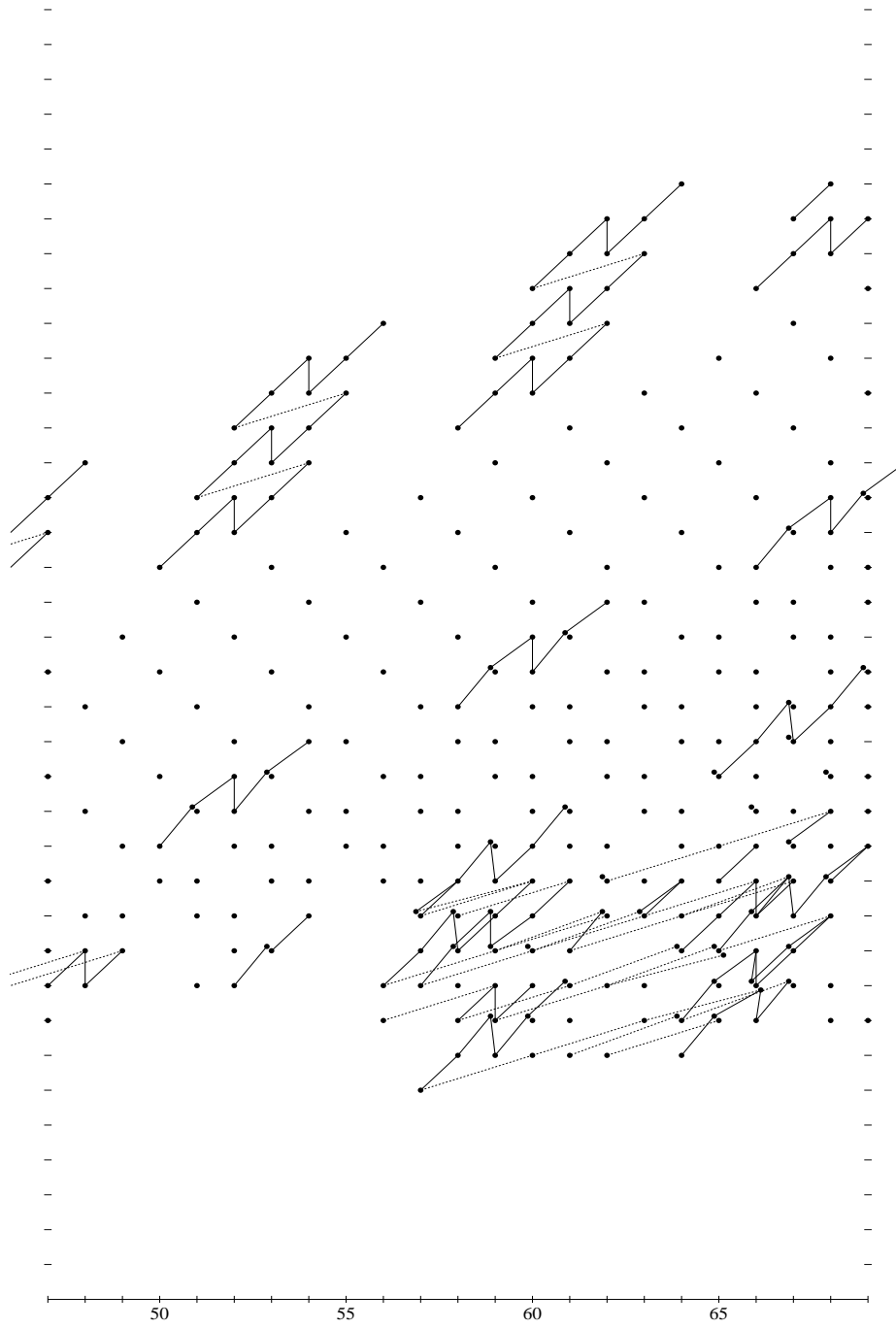
$$\text{Ext}_A^{st}(H^*(S^0 \cup_2 e^1), Z_2), \quad 0 \leq t - s \leq 22$$



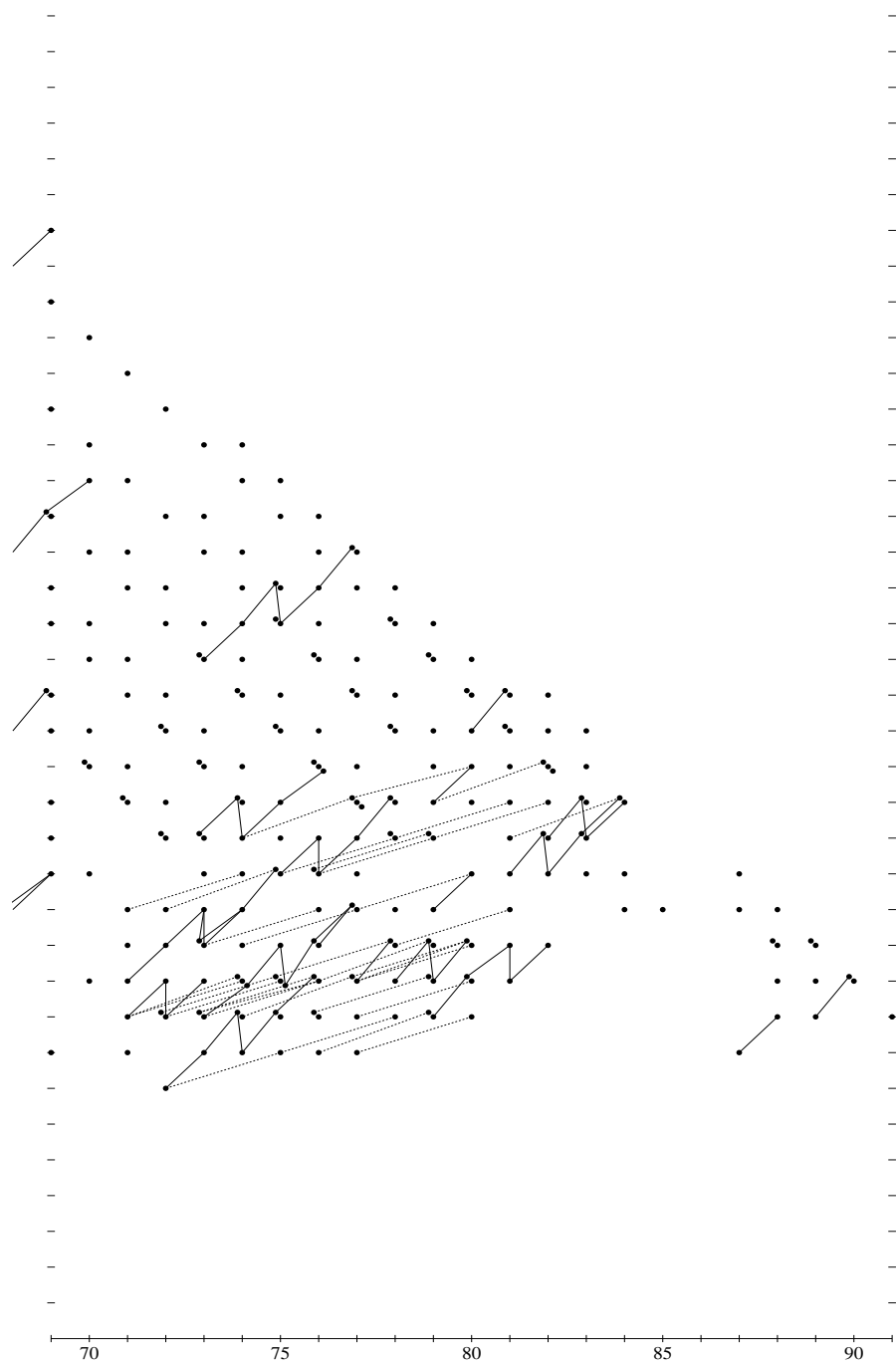
$$\text{Ext}_A^{st}(H^*(S^0 \cup_2 e^1), Z_2), \quad 22 \leq t - s \leq 44$$



$\text{UnstableExt}_A^{s,t}(H^*S^3, Z_2), \quad 25 \leq t - s \leq 47$



$\text{UnstableExt}_A^{s,t}(H^*S^3, Z_2), \quad 47 \leq t - s \leq 69$



$\text{UnstableExt}_A^{st}(H^*S^3, Z_2), \quad 69 \leq t - s \leq 91, \quad t \leq 100$

REFERENCES

- [RRB] Robert R. Bruner, *Calculation of large Ext modules*, Computers in Geometry and Topology (M. C. Tangora, ed.), Marcel Dekker, New York, 1989, pp.79–104.
- [HRS] R. R. Bruner, J. P. May, J. E. McClure, M. Steinberger, *H_∞ Ring Spectra and their applications*, Lecture Notes in Math., vol. 1176, Springer-Verlag, Berlin and New York, 1986.
- [CGMM] E.B. Curtis, P. Goerss, M.E. Mahowald, R.J. Milgram, *Calculations of unstable Adams E_2 terms for spheres*, Algebraic Topology Proc., (Seattle, 1985), (H.R. Miller and D.C. Ravenel, eds.), Lecture Notes in Math., vol. 1286, Springer-Verlag, Berlin and New York, 1987, pp. 208–266.
- [CM86] Edward Curtis and Mark Mahowald, *EHP computations of $E_2(S^n)$* , Computers in Geometry and Topology (M. C. Tangora, ed.), Marcel Dekker, New York, 1989, pp.105–120.
- [CM88] ———, *The unstable Adams spectral sequence for S^3* , Algebraic Topology: Proc. Internat. Conf., (Evanston, 1988), (Mark Mahowald and Stewart Priddy, eds.), Contemp. Math., vol. 96, Amer. Math. Soc., Providence, R. I., 1989, pp. 125–162.
- [DM] Donald M. Davis and Mark Mahowald, *A new spectrum related to 7-connected cobordism*, Algebraic Topology: Proc. Internat. Conf., (Arcata, 1986), (G. Carlsson, R.L. Cohen, H.R. Miller, and D.C. Ravenel, eds.), Lecture Notes in Math., vol. 1370, Springer-Verlag, Berlin and New York, 1989, pp. 126–134.
- [Davis] Donald M. Davis, *Some new immersions and non-immersions of real projective spaces*, Proc. Northwestern Homotopy Theory Conf., (Evanston, 1982), Contemp. Math., vol. 19, Amer. Math. Soc., Providence, R. I., 1989, pp. 51–64.
- [Liu] Arunas Liulevicius, *Coalgebras, resolutions, and the computer*, Math. Algorithms **1** (1966), 4–11.
- [MCT] M.C. Tangora, *Computing the homology of the lambda algebra*, Mem. Amer. Math. Soc., vol. 58, no. 337, Amer. Math. Soc., Providence, R. I., 1985.

MATHEMATICS DEPARTMENT, WAYNE STATE UNIVERSITY, DETROIT, MICHIGAN, 48202
E-mail address: rrb@math.wayne.edu