COMPUTING THE HOMOLOGY OF THE C-MOTIVIC LAMBDA ALGEBRA

KEITA ALLEN

Abstract. The classical lambda algebra is a bigraded differential algebra, useful for computing both the stable and unstable homotopy groups of spheres. A procedure called the Curtis algorithm yields a recursive way to compute the homology of this algebra. Recent work has introduced $k$-motivic versions of the lambda algebra for appropriate choice of base field $k$, and proved that they are of similar utility in the context of motivic homotopy theory. We investigate a generalization of the Curtis algorithm for the classical lambda algebra to computing the homology of the C-motivic lambda algebra.

Contents

1. Introduction 1
   Acknowledgements 2
2. Spectral sequences 3
3. The classical lambda algebra 6
4. The classical Curtis algorithm 8
5. The C-motivic lambda algebra 13
6. A C-motivic Curtis algorithm 15
7. Future directions 18
References 18

1. Introduction

If one were to poll a collection of algebraic topologists and ask them to name a problem they believed to be central to their field, many would name the computation of the homotopy groups of spheres. These groups, denoted $\pi_n(S^m)$, are fairly simple to define, but have been proven to both hold a plethora of information and be incredibly hard to compute.

As a simplification of sorts, recent efforts have focused on trying to determine what are known as the stable homotopy groups of spheres, denoted $\pi_n(S)$. In this setting, we have a powerful computational tool: the (classical) mod 2 Adams spectral sequence. This machine starts with a page of groups, $E_2 = \text{Ext}^*_A(F_2, F_2)$, where $A$ is the mod 2 Steenrod algebra, and describes an iterative procedure by which one can recover information about the 2-torsion part of $\pi_n(S)$.

However, even with this simplification, our work is far from easy! The starting point for the spectral sequence, the groups $\text{Ext}^*_A(F_2, F_2)$, are already extremely...
complicated, and determining these groups is a problem in itself. One tool for determining these groups is the (classical) mod 2 lambda algebra, denoted $\Lambda_{cl}$. This is a bigraded differential algebra, whose homology is exactly $\text{Ext}^{*,*}(\mathbb{F}_2, \mathbb{F}_2)$. This algebra has an explicit presentation which is simple enough to allow us to reasonably compute its homology, and in fact some insights by E. B. Curtis have given us a fairly efficient process for this computation. This process is dubbed the (classical) Curtis algorithm, and is outlined in [Rav86] and described in more detail in [Tan85].

Now, the discussion above is all in the context of what I will call classical homotopy theory, the homotopy theory of spaces. Innovations by Vladimir Voevodsky have introduced a parallel theory, known as motivic homotopy theory, which is the homotopy theory of smooth schemes over some choice of base field $k$. Here, we can make definitions paralleling many of those described above; there are $k$-motivic stable homotopy groups, a $k$-motivic mod 2 Adams spectral sequence, and most recently, thanks to work of Balderrama, Culver and Quigley in [BCQ21], a $k$-motivic mod 2 lambda algebra.

The development of this theory has allowed for the application of techniques from homotopy theory to algebraic geometry – and vice versa. This has been to tremendous effect! For an application towards algebraic geometry, the development of motivic homotopy theory was towards Voevodsky’s resolution of the Milnor conjectures; for an application towards classical homotopy theory, the computation of the stable homotopy groups of spheres has been extended to a vastly larger range by Isaksen, Wang and Xu in [IWX20], crucially making use of the $\mathbb{C}$-motivic Adams spectral sequence.

With this context, it is natural to ask whether there is some interplay between the classical and motivic versions of the lambda algebra; given the computational breakthrough mentioned above, of particular interest to topologists is the case of base field $\mathbb{C}$. It has been suggested that in this case there is indeed an intimate relationship; Behrens has conjectured that with some care, one can effectively read off the homology of the $\mathbb{C}$-motivic lambda algebra from the output of the classical Curtis algorithm. The purpose of this paper is to explore this relationship.

In §2, we recall some basics about spectral sequences, and give a description of the Adams spectral sequence. In §3, we recall the classical lambda algebra. In §4 we give a description of the classical Curtis algorithm, along with some computations in a low range. In §5, we introduce the $\mathbb{C}$-motivic lambda algebra, and discuss its structure. In §6, we discuss the modification to the Curtis algorithm to computing in the $\mathbb{C}$-motivic case.

**Prerequisites.** We will ask for familiarity with some fundamentals of algebraic topology, such as homotopy groups. We will also assume some basic familiarity with homological algebra; in particular, we will ask that the reader know of Ext. Although not strictly necessary, it may also be nice to have some familiarity with spectral sequences. We will give a quick overview of this material in §2, but we encourage the interested reader to look at and around our cited sources for further information.

**Acknowledgements**

I would like to express much thanks to Mark Behrens, for suggesting this project and supervising it through thick and thin. I would not have gotten very far on this project were it not for his patient explanations of the relevant material. Much
thanks also to Peter May, for helping to organize my stay at Chicago, suggesting further directions for research, and showing me a lot of new math. Thank you to Victoria Flores and the Leadership Alliance for funding and housing me this summer; this was my first opportunity to fully focus on math that I was interested in, and it was wonderful. Finally, thank you to all the great people I got a chance to meet in the Leadership Alliance cohort at UChicago and in the math REU, for making this a very fun summer.

2. Spectral sequences

In this section, we give a brief introduction to spectral sequences, to the extent that they will be used in the remainder of the paper. From what the author has heard and experienced, the only way to become comfortable with spectral sequences is by working through many computations with them. We will not accomplish this here; the purpose of this section is solely to collect some general facts about spectral sequences that may be referenced later on in the paper. We encourage readers to skip this section initially, and return as necessary.

For a reader looking to learn about spectral sequences, we refer to [May], [Mil21], and [Rav86] for some places where they may start (listed roughly in order of ascending difficulty), and to [McC01] for more details. We borrow much of the exposition on spectral sequences in general abelian categories from the Stacks project, [Sta18, Tag 011M].

2.1. What is a spectral sequence? At first glance, a spectral sequence is a scary, complicated jumble of homological algebra. This may not change much upon second and third glances, but they have proved to be of remarkable utility throughout algebraic fields, scariness notwithstanding. We give the most general definition here.

Definition 2.1. Let $\mathfrak{A}$ be an abelian category. A spectral sequence in $\mathfrak{A}$ is a system $\{E_r, T_r, d_r\}_{r \geq 1}$, where

- $E_r$ is an object in $\mathfrak{A}$,
- $T_r : \mathfrak{A} \to \mathfrak{A}$ is an isomorphism of categories (also known as a shift functor), and
- $d_r : E_r \to T_r E_r$ is a morphism such that $T_r d_r \circ d_r = 0$,

such that $E_{r+1} = \ker d_r / \text{im } T_r^{-1} d_r$. The object $E_r$ are called pages, and the morphisms $d_r$ are called differentials.

As the name for the terms $E_r$ may suggest, a common way to visualize spectral sequences is as a book, where one can move from page $E_r$ to page $E_{r+1}$ by taking homology with respect to the $d_r$ differential. By taking homology, we obtain an object which is “smaller” than the last page, and when we study spectral sequences we are most often concerned with what the “limit” of this successive shrinking is. Towards making this idea rigorous, we make the following definitions. We let $T_r = \text{id}_{E_r}$ for now to prevent notational clutter; everything can also be done while accounting for $T_r$.

1For those unfamiliar with abelian categories, you can think of it as something like the category of $R$-modules, where you can take things like kernels, cokernels, etc.
Definition 2.2. We define subobjects $Z_r, B_r \subset E_1$, fitting into the following sequence of inclusions

$$0 \rightarrow B_1 \rightarrow B_2 \rightarrow \cdots \rightarrow B_r \rightarrow \cdots \rightarrow Z_r \rightarrow \cdots \rightarrow Z_2 \rightarrow Z_1 \rightarrow E_1$$

such that $E_r \cong Z_r/B_r$. We do this inductively:

- Define $Z_1 = E_1$, $B_1 = 0$.
- With $Z_{r-1}, B_{r-1}$ defined such that $Z_{r-1}/B_{r-1} \cong E_{r-1}$ define:
  - $Z_r$ to be the unique subobject of $E_r$ such that $Z_r \subset Z_{r-1}$, and $Z_r/B_{r-1}$ is the kernel of $d_{r-1} : Z_{r-1}/B_{r-1} \rightarrow Z_{r-1}/B_{r-1}$, and
  - $B_r$ to be the unique subobject of $E_r$ such that $B_{r-1} \subset B_r$, and $B_r/B_{r-1}$ is the image of $d_{r-1} : Z_{r-1}/B_{r-1} \rightarrow Z_{r-1}/B_{r-1}$.

If they exist, let $Z_\infty = \bigcap_r Z_r$ and $B_\infty = \bigcup_r B_r$. Then these fit into the above sequence as follows:

$$0 \rightarrow B_1 \rightarrow B_2 \rightarrow \cdots \rightarrow B_\infty \rightarrow Z_\infty \rightarrow \cdots \rightarrow Z_2 \rightarrow Z_1 \rightarrow E_1$$

We define the $E_\infty$ page of our spectral sequence, also known as the limit, to be the object $Z_\infty/B_\infty$.

Many times it will be the case that $d_i = 0$, or in the case when $\mathfrak{A}$ is graded, that certain graded pieces $(d_i)_p = 0$ for $i$ greater than some $r$. In this case, (the relevant pieces of) $Z_\infty$ and $B_\infty$ are (the relevant pieces of) $Z_r$ and $B_r$ respectively. This means that, at least partially, $E_r = E_\infty$; if our goal is to compute this $E_\infty$ page, we can accomplish this by having knowledge of the $E_r$ page.

2.2. Where do spectral sequences come from, and why should we care? So far, we have given some general definitions regarding spectral sequences, but we have yet to actually describe any. More pressingly, we have not given a reason to actually care about them; we know they have something like a limiting page, but can this compute something we care about? Luckily, the answer to this is yes. In the remainder of this section, we will describe a class of spectral sequences which will help us compute the homology of a chain complex equipped with a filtration, which is the case which will arise later in the paper.

We now fix our abelian category to be the category of $(\mathbb{Z}\text{-graded})$ chain complexes of $R$-modules, $\text{Ch}(R)$ for some commutative ring $R$, and draw from exposition in [McC01]. For a generalization to filtered differential objects in abelian categories, we refer the reader to [Sta18, Tag 012A]; one should be glad to know that essentially everything we discuss carries over directly.

Definition 2.3. Take a chain complex $C$ in $\text{Ch}(R)$. An (ascending) filtration on $C$ is a collection of subcomplexes $F_iC \subset C$, $i \in \mathbb{Z}$, such that $F_iC \subset F_{i+1}C$.

$$\cdots \longrightarrow F_{i-1}C \longrightarrow F_iC \longrightarrow F_{i+1}C \longrightarrow \cdots$$

We say the filtration is exhaustive if $\bigcup_i F_iC = C$, and Hausdorff if $\bigcap_i F_iC = 0$.

We often denote a chain complex equipped with a filtration as a pair $(C, F)$, and refer to it as a filtered chain complex. Our first observation is that the homology of a filtered chain complex inherits a filtration via the formula

$$F_pH(C) = \text{im}(H(F_pC) \rightarrow H(C)),$$

where the map is induced by the inclusion.
Now, associated to a filtered chain complex is a certain graded object, which we discuss now.

**Definition 2.5.** Given a filtered complex \((C, F)\), the *associated graded object* of \((C, F)\), denoted \(\text{gr}(C, F)\), is a graded chain complex formed by taking the cokernel of each inclusion:

\[
\text{gr}_i(C, F) = F_iC/F_{i-1}C.
\]

When there is no ambiguity about the filtration on \(C\), it is common to denote \(\text{gr}(C, F)\) simply by \(\text{gr} C\).

Conceptually, \(\text{gr}_i C\) is the portion of \(C\) which lies purely in “filtration degree” \(i\). Further, \(\text{gr}_i C\) is a chain complex itself, and so we can consider its homology \(H(\text{gr}_i C)\); this can be thought of as the portion of the homology of \(C\) coming purely from filtration degree \(i\). A natural question to ask then is whether it is possible to recover the homology of the whole complex \(H(C)\) from these portions. With the descriptions given, one might naively expect to have \(H(C) = \bigoplus_i H(\text{gr}_i C)\); sadly, this is not the case in general. This is roughly because the differential on \(C\) does not have to preserve filtration degree of elements; one can imagine a case where a class which lives in filtration degree \(i\) is the boundary of a class which lives in higher filtration degree. In this case, classes which survive in \(H(\text{gr}_i C)\) could be zero in \(H(C)\); \(H(C)\) is ‘smaller’ than \(\bigoplus_i H(\text{gr}_i C)\).

Given our discussion near the beginning of the section, a question we can ask then is whether there is a spectral sequence which starts with this object \(E_1^{n,s} = H_s(\text{gr}_n C)\), converging to something near \(H(C)\). The answer to this question turns out to be in the affirmative; this is the spectral sequence associated to a filtered complex.

**Construction 2.7 ([McC01] Thms 2.5, 3.2).** Given a chain complex \(C\) with an ascending filtration \(F\), there is a spectral sequence of bigraded \(R\)-modules \(\{E_r^{n,*}, d_r\}, r = 1, 2, \ldots\), where

\[
E_1^{n,*} \simeq H^*(\text{gr}_n C),
\]

and where \(d_r\) has bidegree \((-r, 1)\). Further, if this filtration is exhaustive and Hausdorff, then this spectral sequence converges to \(H(C)\), in the sense that we have

\[
E_\infty^{n,*} \simeq \text{gr}_n(H^*(C), F).
\]

This information is often condensed as \(E_1^{n,*} \simeq H^*(\text{gr}_n C) \Rightarrow H^*(C)\); this is called the *signature* of the spectral sequence.

The details of the construction of this spectral sequence are a bit messy; it is often treated as a black box. However, they are not untractable; one can find these details in the proof of Thm. 2.6 in [McC01], or in [Sta18, Tag 012A]. The main information necessary to understand the spectral sequence more explicitly are the groups \(Z_r\) and \(B_r\); we summarize here.

\[
Z^n_r = \frac{F_n C \cap d^{-1}(F^{n-r} C) + F_n C}{F_{n-1} C}, \quad B^n_r = \frac{F_n C \cap d(F^{n+r-1} C) + F_{n-1} C}{F_{n-1} C}
\]

So morally, one can think of \(Z^n_r\) as the elements in filtration degree less than or equal to \(n\) which have boundaries in filtration degree less than or equal to \(n - r\), and \(B^n_r\) as the elements in filtration degree less than or equal to \(n\) which are boundaries of
elements in filtration degree less than or equal to $n + r - 1$. The differentials $d_r$ are obtained by restricting the differential $d$ on $C$ to $Z_r$. This understanding will help us out later.

3. The classical lambda algebra

In this section, we introduce the prototypical version of our main object of study, the classical mod 2 lambda algebra, and describe the Curtis algorithm for computing its homology. References for this section include [Tan85] and §3.3 of [Rav86].

Definition 3.1. The (mod 2) lambda algebra, denoted $\Lambda_{cl}$, is the bigraded associative differential algebra over $\mathbb{F}_2$, generated by elements $\lambda_i, n \geq 0$. These are subject to relations

$$\lambda_i \lambda_{2i+1+n} = \sum_{j \geq 0} \binom{n-j}{j} \lambda_{i+n-j} \lambda_{2i+1+j}, \quad i, n \geq 0,$$

and have a differential defined by

$$d(\lambda_i) = \sum_{j \geq 1} \binom{i-j}{j} \lambda_{i-j} \lambda_{j-1},$$

extended to monomials via the Leibniz rule, $d(xy) = d(x)y + xd(y)$. The bidegree $(s, t)$ is given by

$$|\lambda_i| = (1, i + 1).$$

Convention 3.1. In what follows, we will denote the classical lambda algebra simply by $\Lambda$, when there is no ambiguity.

The biggest utility of the lambda algebra lies in the fact that it is an $E_1$-term for the Adams spectral sequence for the sphere.

Theorem 3.5 ([BCK+66]). $H^*(\Lambda) \simeq \text{Ext}_{\mathbb{A}}^*(\mathbb{F}_2, \mathbb{F}_2)$, the $E_2$ page for the Adams spectral sequence converging to the 2-primary part of the stable homotopy groups of spheres. Further, this isomorphism is multiplicative; the multiplicative structures on $H(\Lambda)$ inherited from the multiplication on $\Lambda$ and that on $\text{Ext}_{\mathbb{A}}(\mathbb{F}_2, \mathbb{F}_2)$ are compatible.

Let us move to discussing the structure of this algebra.

Definition 3.6. A monomial $\lambda_{i_1} \lambda_{i_2} \ldots \lambda_{i_m} \in \Lambda$ is called admissible if $2i_j + 1 > i_{j+1}$ for all $j$.

The relations (3.2) give us the following fact regarding the structure of the lambda algebra.

Proposition 3.7. The admissible monomials form a basis for $\Lambda$ (as an $\mathbb{F}_2$-vector space).

For this reason, when working with elements in the lambda algebra, we will assume that they are the sums of admissible monomials. This admissible basis is what gives rise to some of the interesting structure in the lambda algebra, which is ultimately exploited in the Curtis algorithm. The story starts by considering the following subcomplexes.

Definition 3.8. $\Lambda(n) \subset \Lambda$ is the subcomplex spanned by the admissible monomials $\lambda_{i_1} \lambda_{i_2} \ldots \lambda_{i_m}$ with leading term $i_1 < n$. 
Remark 3.9. Note that before making Definition 3.8, it must be checked that the differential on $\Lambda$ does not increase leading term. Although not immediately obvious, this is a fairly straightforward and not particularly enlightening exercise, so we will omit proof here.

We can see that $\Lambda(n) \subset \Lambda(n + 1)$; this gives rise to a filtration on the lambda algebra.

Construction 3.10. There is an (exhaustive, Hausdorff) ascending filtration $F = \{F_i\}$ on $\Lambda$, with

$$F_i \Lambda = \begin{cases} 0 & i \leq 0 \\ \Lambda(i) & i > 0 \end{cases}$$

We will call this the leading term filtration.

To see why this filtration is worth considering, let us begin by analyzing its filtration quotients. The following result tells us that they turn out to be interesting.

Proposition 3.11. For all $n \geq 1$, we have a short exact sequence of complexes

$$0 \rightarrow \Lambda(n) \rightarrow \Lambda(n + 1) \rightarrow \Sigma^{1,n+1} \Lambda(2n + 1) \rightarrow 0.$$  

Proof. The main content of this statement is an identification between $\Lambda(n + 1)/\Lambda(n)$ and $\Lambda(2n+1)$. This is seen by "dropping the leading term" on the nonzero elements:

$$\Lambda(n + 1) \leftrightarrow \Lambda(2n + 1)$$

$$\lambda_n \lambda_{i_1} \lambda_{i_2} \cdots \lambda_{i_m} \leftrightarrow \lambda_{i_1} \lambda_{i_2} \cdots \lambda_{i_m}.$$  

Given that $\lambda_n \lambda_{i_1} \lambda_{i_2} \cdots \lambda_{i_m}$ is admissible, we know that $i_1 < 2n + 1$, which guarantees that $\lambda_{i_1} \lambda_{i_2} \cdots \lambda_{i_m}$ is in $\Lambda(2n + 1)$. We can check that this identification respects the differential; shifting degrees according to the degree of $\lambda_n$ gives us the desired result.

Figure 1 provides an illustration of the structure given by the leading term filtration. The observation above tells us that the filtration quotients can themselves be considered as (shifts of) objects further up in the filtration, giving us a recursive flavor which eventually gives rise to the Curtis algorithm.

As some may have guessed, the Curtis algorithm comes from a certain spectral sequence converging to the homology of $\Lambda$. In §3.3 of [Rav86], Ravenel constructs this by noting that long exact sequences in homology associated to the short exact

![Figure 1. An illustration of the leading term filtration on $\Lambda$ and its associated graded object.](image-url)
sequences in Proposition 3.11 assemble into an exact couple. To help us a bit with getting a more explicit grasp of the pages and differentials in this spectral sequence, and to additionally make the convergence easier to see, we provide an alternative construction of this spectral sequence directly from the leading term filtration on $\Lambda$.

**Construction 3.14** (Algebraic EHP spectral sequence). The leading term filtration on $\Lambda$ gives rise to a spectral sequence with signature

$$E_1^{n,s,t} = \begin{cases} H^{s-1,t-n}(\Lambda(2n-1)) & s > 0, n \leq m \\ \mathbb{Z}/2 & s, t, n = 0 \\ 0 & \text{else} \end{cases} \implies H^{s,t}(\Lambda),$$

with differentials $d_r$ having tridegree $(-r, 1, 0)$.

**Proof.** This spectral sequence comes from applying Construction 2.7 to $\Lambda$, initially considered as a $\mathbb{Z}$-graded chain complex by forgetting the second degree $t$. Given Prop. 3.11, we obtain a spectral sequence with signature

$$E_1^{n,s} = \begin{cases} H^{s-1,*}(\Lambda(2n-1)) & s > 0 \\ \mathbb{Z}/2 & s, n = 0 \\ 0 & \text{else} \end{cases} \implies H^{s,*}(\Lambda),$$

Since the differential $d$ on $\Lambda$ is homogeneous with respect to $t$, the details of Construction 2.7 permit us to additionally consider this grading, giving us the trigraded spectral sequence above. \hfill \Box

By “truncating” the leading term filtration at $\Lambda(m)$, we can mimic the construction above to retrieve a spectral sequence converging instead to the homology of this subcomplex.

**Construction 3.17** (Unstable algebraic EHP spectral sequences). For $m > 0$, the leading term filtration on $\Lambda(m)$ gives rise to a spectral sequence with signature

$$E_1^{n,s,t} = \begin{cases} H^{s-1,t-n}(\Lambda(2n-1)) & s > 0, n \leq m \\ \mathbb{Z}/2 & s, t, n = 0 \\ 0 & \text{else} \end{cases} \implies H^{s,t}(\Lambda),$$

with differentials $d_r$ having tridegree $(-r, 1, 0)$.

With this in mind, we are now ready to discuss the algorithm.

4. **The classical Curtis algorithm**

The Curtis algorithm is an inductive process for computing the homology of the lambda algebra via the EHP spectral sequences. The key principle is that instead of keeping track of entire polynomials in the lambda algebra, it suffices to keep track of their “leading terms.” Before we illustrate this, however, we need to discuss what we mean by “leading.”

**Construction 4.1.** The monomials in $\Lambda$ are given a total ordering lexicographically. This ordering is defined recursively:

- 1 is lexicographically less than all monomials.
Given monomials $x = \lambda_{i_1}x'$ and $y = \lambda_{j_1}y'$, where $x'$ and $y'$ are also monomials, $x$ is lexicographically less than $y$ if $i_1 < j_1$, or if $i_1 = j_1$ and $x'$ is lexicographically less than $y'$.

Let us now move to discussing how we will be storing the data we compute.

4.1. Curtis tables. A Curtis table is a table displaying the data of the EHP spectral sequence. From here on out, we will streamline our notation a bit.

Notation 4.2. We denote a monomial $\lambda_{i_1}\lambda_{i_2}\ldots\lambda_{i_n}$ as a list of numbers $i_1 i_1 \ldots i_n$. For example, $\lambda_{11}\lambda_{5}\lambda_{2}$ is denoted as 11 5 2.

With that, Figure 2 is a Curtis table, completed out to stem degree $t - s = 14$. The table is sorted horizontally by stem degree, and vertically by filtration degree $n$. As one may guess, the dots in the table represent elements of the $E_1$ page of the EHP spectral sequence, filtered horizontally by stem and vertically by leading term. The arrows in the table are the differentials in the spectral sequence. The dots in the table which do not support differentials represent elements which survive to elements in $H(\Lambda) = \text{Ext}_A(F_2, F_2)$. These are highlighted in red for viewing convenience. By truncating the table and ignoring information coming from below a fixed $n = m$, we obtain a table for the truncated EHP spectral sequence for $\Lambda(m)$, which we can then use to read off the homology of this subcomplex.

Example 4.3. We can see that $H(\Lambda)$ has no elements in stem 4, as all the dots in that column support a differential. However, $H(\Lambda(5))$ has one surviving element in stem 4, represented in the table as 3 1; the entry which kills it in $\Lambda$, 5, is not in $\Lambda(5)$.

For our later discussion of the algorithm, it will be helpful to have a more concrete understanding of the entries and differentials in the table. Each column in the figure stores the lexicographically leading term of cycles and preimages of boundaries, filtered vertically by their leading term. The spectral sequence allows us to reconstruct entire cycles from this information, as we will see. The differentials in the Curtis table record boundaries. Precisely, an arrow going from a monomial $x$ to a monomial $y$ means “a polynomial with leading term $y$ is the boundary of a polynomial with leading term $x$.”

From a Curtis table completed out to stem $k$, we can read off the Adams $E_2$ page $\text{Ext}_A(F_2, F_2)$ for stems less than $k$ fairly simply. One takes the elements which do not support differentials and places them in their appropriate place on a chart using their degree. The Adams $E_2$ page obtained from the Curtis table in Figure 2 is displayed in Figure 3.

With this background, we are ready to discuss the algorithm.

4.2. The classical algorithm. The Curtis algorithm can be thought of as a way of filling out the table in Figure 2 via (strong) induction on stem. That is, the data in the Curtis table for stem less than $k$ is used to fill out the column for stem $k$. A detailed description of the algorithm (which does not make reference to spectral sequences), in addition to a proof of correctness, is available in [Tan85]. We would like to thank Mark Behrens for his patient explanations of this process.

We will explain the general procedure while walking through the process of expanding the table out from stem 5 out to stem 6, as most of the phenomena we need to look out for are already visible in this range. The process can be largely split up into two parts:
Figure 2. A Curtis table for the classical lambda algebra.

(1) Populate column $k$ with the stem $k$ portion of the $E_1$ page of the EHP spectral sequence, using previously computed data.

(2) Compute differentials in the spectral sequence, using the (purely algebraic) formula for the boundary map $d$ on $\Lambda$. The elements which we know do not support differentials represent classes in $H(\Lambda)$.

Before we get started, a quick note on notation.

Convention 4.1. Unless stated otherwise, $H^\ast(\Lambda)/H^\ast(\Lambda(n))$, with a single index, will refer the portion of $H(\Lambda)/H(\Lambda(n))$ living in a fixed stem.

4.2.1. Step 1: filling in the table. Let us start by discussing how we fill columns in. The entries which populate the box in row $n$ of column $t - s = k$ are the elements in stem $k$ of $H(\text{gr}_n \Lambda)$. The base case of our induction is when $k = 0$; in this case, we can see that the only nonzero group we have is $H^0(\text{gr}_1 \Lambda) = H^0(\Lambda(1)) \cong \Lambda(1)$, which has the elements $0^k$ for all $k \geq 0$. 
Now assume we have completely determined all information in the Curtis table for \( t - s < k \). We use this to determine the data in stem \( k \). Since \( H^k(gr_n \Lambda) \cong H^{k-n+1}(\Lambda(2n-1)) \), we equivalently want to find the elements living in stem \( k-n+1 \) of \( H(\Lambda(2n-1)) \). For \( n > 2 \), this is something we have already completely computed!

We take a look at column \( k-n+1 \) of our table, ignore information coming from rows greater than \( 2n-1 \), and read off the homology by taking the elements which do not support a differential. We then mark these in the appropriate cell, making sure to multiply by \( \lambda_{n-1} \) to appropriately reidentify them as elements in \( H(gr_n \Lambda) \).

**Example 4.4.** Let’s try to fill out the box in row \( n = 4 \) of Figure 4, boxed in blue on the right hand side. For this, we are looking for elements of \( H^3(\Lambda(7)) \). We look at the column \( t - s = 3 \) of our Curtis table, and disregard information coming from below row \( n = 7 \) (in this case, there is no such information). We take all elements in rows \( n \leq 7 \) which do not support a differential, which in this case are 1 1 1, 2
Figure 4. Computing the stem 6 portion of $H(\Lambda)$. The cells boxed in a given color on the left are those used to determine the new box of the same color on the right. Ignore the arrows coming into stem 6; we are not supposed to know those yet.

1, and 3, and multiply them on the left by $(n - 1) = 3$. These are the elements we write down in the box.

We pause here for two remarks.

Remark 4.5. It is important to make sure that one reads off $H^{k-n+1}(\Lambda(2n - 1))$, not $H^{k-n+1}(\Lambda)$. For example, when filling in the box in row $n = 3$, one runs into the situation given in Example 4.3 of determining $H^4(\Lambda(5))$. At a glance, it may appear that there are no surviving elements here; however, while 3 1 supports a differential in $\Lambda$, it does not support a differential in $\Lambda(5)$, as $5 \notin \Lambda(5)$.

Remark 4.6. Note that our knowledge of $H(\Lambda)$ or $H(\Lambda(m))$ in any given stem relies on our computation of the $E_1$ page and differentials out of it in one stem degree higher; we need this information to determine whether cycles are boundaries or
not. For this reason, at this stage we cannot fill out the box for row \( n = 2 \); this involves the computation of \( H^{k-1}(\Lambda(3)) \), which requires us to compute differentials in stem \( k \) first.

4.2.2. Step 2: Computing differentials. Once we have filled in column \( k \) for rows \( n > 2 \), we move to determining differentials. This is via a recursive process referred to in [Tan85] as “completing the cocycle”. For this process, it is important that we start work top to bottom; start with row \( n = 3 \) and work the way down.

(1) Given a class \( x = \lambda_{i_1}\lambda_{i_2} \ldots \lambda_{i_m} \) in the table, compute \( d(x) \). If \( d(x) = 0 \), we do nothing; \( x \) is a candidate to be a permanent cycle.

Otherwise, let \( y = \lambda_{j_1}\lambda_{j_2} \ldots \lambda_{j_{m+1}} \) be the lexicographically leading term of \( d(x) \). If \( y \) is in the table and does not already support a differential, there is a differential taking the class \( x \) to the class \( y \). Draw this into the table as an arrow from (the lexicographically leading term of) \( x \) to \( y \).

(2) Otherwise, we want to find the longest tail segment of \( y, \lambda_{j_1} \ldots \lambda_{j_{m+1}} \), which has a differential going into it. When we have this, let \( z \) denote the element in the table which hits it under the differential. We then start over at step 1, replacing \( x \) with \( x + \lambda_{i_1} \ldots \lambda_{i_{\ell-1}} z \); compute its differential, check if its leading term exists in the table...

Example 4.7. All of the elements that we filled in for stem \( t-s = 6 \) do not require us to go to step 2 above. For example, \( d(4 \ 1 \ 1) = 2 \ 1 \ 1 \), which exists in the table and does not support a differential, so we draw an arrow between those two elements. On the other hand, we have \( d(3 \ 1 \ 1) = 0 \), so we leave it alone for now.

Example 4.8. Going out to further stems, we find examples where we have to go step 2. Taking a look at \( 2, \) we can take \( 9 \ 1 \ 1 \) in stem \( 11 \). We have that \( d(9 \ 1 \ 1) = 7 \ 1 \ 1 \ 1 + 5311.7111 \) is an element in the table, but already supports a differential, from \( 8 \ 2 \ 1 \). So we add \( 8 \ 2 \ 1 \) and compute the boundary of the new polynomial; we obtain \( d(9 \ 1 \ 1 + 8 \ 2 \ 1) = 6121 + 5311 + 4321 \). We see that \( 6121 \) is not in the table, but \( 1 \ 2 \ 1 \) is and has an arrow coming from \( 2 \ 3 \). So we add \( 6 \ 2 \ 3 \) and compute the boundary again. After repeating this process, we obtain \( d(9 \ 1 \ 1 + 8 \ 2 \ 1 + 6 \ 2 \ 3 + 551 + 443 + 335) = 22 \ 3 \ 3 \), which exists in the table and does not support a differential. So there is a differential from \( 9 \ 1 \ 1 \) to \( 2 \ 2 \ 3 \ 3 \); we draw this into the table as an arrow.

After computing differentials, we have the information to determine \( H^{k-1}(\Lambda(3)) \); this goes in the box in row \( n = 2 \). We have now completed filling out column \( k \) of our table!

5. The \( C \)-motivic lambda algebra

We begin our move into the motivic world. Recent work of Balderrama, Culver and Quigley in [BCQ21] has introduced \( k \)-motivic lambda algebras, where \( k \) is any base field with characteristic not equal to \( 2 \). These allow for computation of the \( E_2 \) page of the \( k \)-motivic Adams spectral sequence. In the remainder of this paper, we specialize to the case of base field \( C \), and work towards describing a Curtis algorithm for computing the homology of the \( C \)-motivic lambda algebra. We begin in this section by covering some structural results.

Definition 5.1. The \( \text{mod} \ 2 \) \( C \)-motivic lambda algebra, denoted \( \Lambda_C \), is a trigraded differential algebra over \( \mathbb{F}_2 \), generated multiplicatively by \( \tau \) and elements \( \lambda_i \) for
\(i \geq 0\). These are subject to relations
\[
\lambda_i \tau = \tau \lambda_i \quad \text{for all } i
\]
\[\lambda_i \lambda_{2i+i+n} = \begin{cases} \sum_{j \geq 0} \binom{n-j-1}{j} \lambda_{i+n-j} \lambda_{2i+1+j} & \text{if odd or } n \text{ even} \\ \sum_{j \geq 0} \binom{n-j-1}{j} \lambda_{i+n-j} \lambda_{2i+1+j} \tau & \text{if even and } n \text{ odd} \end{cases}
\]
and have differential defined by
\[
d(\lambda_i) = \sum_{j \geq 1} \binom{i-j}{j} \lambda_{i-j} \lambda_{j-1},
\]
extended to monomials via the Leibniz rule, \(d(xy) = d(x)y + xd(y)\). The tridegree \((s, \ell, w)\) is assigned as follows:
\[|\tau| = (0, 0, -1)\]
\[|\lambda_i| = \left(1, i + 1, \left\lceil \frac{i}{2} \right\rceil \right).
\]

One may note that \(\Lambda_C\) looks essentially the same as the classical lambda algebra \(\Lambda_{cl}\), aside from the addition of an extra degree \(w\) and a variable \(\tau\). The new degree \(w\) is called the motivic weight, and from a purely algebraic point of view, the variable \(\tau\) can be thought of as something that adjusts this motivic weight to keep the relations 5.2 homogeneous. With this observation, the following proposition follows from the analogous Proposition 3.7 in the classical case.

**Proposition 5.5.** The admissible monomials (as in Def. 3.6) are a basis for \(\Lambda_C\) as a free \(M_C^2 = F_2[\tau]\)-module.

\[M_C^2 := \pi_* (HF_2),\]
where \(HF_2\) is the \(C\)-motivic Eilenberg-Maclane spectrum for \(F_2\). This offers an analogy between the presence of \(F_2\) in the classical setting and \(F_2[\tau]\) in the \(C\)-motivic setting.

As in the classical setting, the main utility of \(\Lambda_C\) is that it helps us compute the input to the \(C\)-motivic Adams spectral sequence.

**Theorem 5.6** ([BCQ21], Section 2.4). \(H^{*, *, *}(\Lambda_C) \simeq \text{Ext}^{*, *, *}(M_C^2, M_C^2)\), the \(E_2\) page of the Adams spectral sequence converging to the 2-primary part of the \(C\)-motivic stable homotopy groups of spheres. Further, this isomorphism is multiplicative; the multiplicative structures on \(H(\Lambda_C)\) inherited from the multiplication on \(\Lambda_C\) and that on \(\text{Ext}_{\Lambda_C}(M_C^2, M_C^2)\) are compatible.

So towards computing this homology, let us begin to mimic our construction of the classical EHP spectral sequence. We begin with the leading term filtration.

**Definition 5.7.** \(\Lambda_C(n) \subset \Lambda_C\) is the subcomplex spanned (as a \(F_2[\tau]\)-module) by the admissible monomials \(\lambda_{i_1} \ldots \lambda_{i_m}\) with leading term \(i_1 < n\).

**Construction 5.8.** There is an (exhaustive, Hausdorff) ascending filtration \(F = \{F_i : \Lambda_C\}_i\) on \(\Lambda_C\), with
\[
F_i : \Lambda_C = \begin{cases} 0 & i \leq 0 \\ \Lambda_C(i) & i > 0 \end{cases}.
\]
We call this the leading term filtration on \(\Lambda_C\).
We can repeat the proof of Proposition 3.11 to retrieve the following result regarding the associated graded object of this filtration.

**Proposition 5.10.** For \( i > 1 \),
\[
gr_i (\Lambda_C, F) \cong \Sigma^{1, i, \lceil \frac{i-1}{2} \rceil} \Lambda_C(2i - 1)
\]

Finally, we obtain our spectral sequences of interest.

**Construction 5.12.** The leading term filtration on \( \Lambda_C \) gives rise to a spectral sequence (of \( F_2[\tau] \)-modules) with signature
\[
E_1^{n,s,t,w} = \begin{cases} 
H^{s-1,t-n,w-[\frac{n-1}{2}]}(\Lambda(2n-1)) & s > 0 \\
\mathbb{Z}/2 & s, t, n, w = 0 \\
0 & \text{else}
\end{cases} \Rightarrow H^{s,t,w}(\Lambda),
\]
with differentials \( d_r \) having tridegree \((-r, 1, 0, 0)\).

**Construction 5.14.** For \( m > 0 \), the leading term filtration on \( \Lambda_C(m) \) gives rise to a spectral sequence (of \( F_2[\tau] \)-modules) with signature
\[
E_1^{n,s,t,w} = \begin{cases} 
H^{s-1,t-n,w-[\frac{n-1}{2}]}(\Lambda(2n-1)) & s > 0, \ n \leq m \\
\mathbb{Z}/2 & s, t, n, w = 0 \\
0 & \text{else}
\end{cases} \Rightarrow H^{s,t,w}(\Lambda),
\]
with differentials \( d_r \) having tridegree \((-r, 1, 0, 0)\).

6. **A \( \mathbb{C} \)-motivic Curtis algorithm**

As in the classical case, the spectral sequences of Constructions 5.12 and 5.14 give rise to a Curtis algorithm which one can use to determine \( H^{s,t,w}(\Lambda) \), using essentially the same process as in the classical case. In order for this to work, however, there are two things which stand out as potential problems.

The first is that, at a glance, it appears that our spectral sequences are not computing \( F_2 \)-vector spaces, but \( F_2[\tau] \)-modules. This seems like it may cause a headache, as we may have to solve nontrivial extension problems. However, it turns out that we don’t have to worry about this. Recall that \( |\tau| = (0,0,-1) \). This tells us that if \( A \) is a \( F_2[\tau] \)-module concentrated in a single motivic weight, then \( \tau \) must act trivially; \( \tau x = 0 \) for any \( x \in A \). In other words, the \( F_2 \)-vector space structure on \( A \) completely determines the \( F_2[\tau] \)-module structure on \( A \), and so we do not have to worry about solving any extra extension problems.

The second one, which is a bit more practically pressing, is regarding the finiteness of this process. In the classical case, we computed \( H(\Lambda) \) inductively, one stem at a time. However, now that we also have the motivic weight to account for, it appears that we might have to either figure out a way to work this into our induction, or somehow compute all motivic weights at once. Our approach will be the latter; with care, this turns out to be a nonissue.

6.1. **\( \mathbb{C} \)-motivic Curtis tables.** We store the data for the \( \mathbb{C} \)-motivic EHP spectral sequences in Curtis tables, of essentially the same form as in the classical case. The elements are sorted horizontally by stem, and vertically by filtration degree; we
do not record motivic weights. We will use a similar shorthand for monomials as before.

**Notation 6.1.** We denote the monomial $\lambda_{i_1} \lambda_{i_2} \ldots \lambda_{i_n} \tau^m$ as $i_1 i_2 \ldots i_n \tau^m$. If $m = 0$, we use the same notation as before and denote this as $\lambda_{i_1} \lambda_{i_2} \ldots \lambda_{i_n}$.

A difference between the classical and $\mathbb{C}$-motivic settings is the potential need to record many $\tau$-multiples of a monomial in the table. For this, we introduce some new notation.

**Notation 6.2.** Let $x$ be a monomial. We write $x(k)$ in the Curtis table to denote the existence of $x\tau^i$ in the table for $i < k$. We write $x(\infty)$ as simply $x$.

For example, $11\,1\,1\,1\,1\,1\,(1)$ means that $11\,1\,1\,1\,1$ is in the table, but $11\,1\,1\,1\,1\,\tau^i$ for $i > 0$ are not.

6.2. **The $\mathbb{C}$-motivic algorithm.** The $\mathbb{C}$-motivic Curtis algorithm follows the same general approach as the classical one;

1. Populate column $k$ with the stem $k$ portion of the $E_1$ page of the $\mathbb{C}$-motivic EHP spectral sequence, using previously computed data.
2. Compute differentials in the spectral sequence, using the (purely algebraic) formula for the boundary map $d$ on $\Lambda_\mathbb{C}$. The elements which we know do not support differentials represent classes in $H(\Lambda_\mathbb{C})$.

The main difference between the classical algorithm and this $\mathbb{C}$-motivic algorithm is that the added motivic weight requires us to be careful to make sure we can compute the homology in each stem in a finite amount of time. We discuss the modifications we need to make to the classical algorithm in order to ensure this.

We adopt the notation in Convention 4.1 to refer to homology in a fixed stem of $\Lambda_\mathbb{C}/\Lambda_\mathbb{C}(n)$.

6.2.1. **Step 1: Filling in the table.** As before, we start by filling the table in with the relevant portion of the $E_2$ page. Conceptually, what we do is the same as the classical case. Our base case is again $k = 0$; our only nonzero contribution is from $H^0(\text{gr}_0 \Lambda_\mathbb{C}) = H^0(\Lambda_\mathbb{C}(1)) \simeq \Lambda_\mathbb{C}(1)$, which has elements $0^k \tau^i$ for $i, k \geq 0$. Following the scheme given in Notation 6.2, we mark this in the table as (empty string), 0, 0 0, . . .

Now assume we have completely determined all information in the $\mathbb{C}$-motivic Curtis table for $t - s < k$. We will again use this to determine the $E_2$ term in stem $k$. Since $H^k(\text{gr}_n \Lambda_\mathbb{C}) \simeq H^{k-n+1}(\Lambda_\mathbb{C}(2n - 1))$, we equivalently want to find the elements living in stem $k - n + 1$ of $H(\Lambda_\mathbb{C}(2n - 1))$. For $n > 2$, this is something we have already completely computed! We take a look at column $k - n + 1$ of our table, ignore information coming from rows greater than $2n - 1$, and read off the homology by taking the elements which do not support a differential. We then mark these in the appropriate cell, making sure to multiply by $\lambda_{n-1}$ to appropriately reidentify them as elements in $H_\mathbb{C}(\text{gr}_n \Lambda)$.

The one difference from the classical case that is worth being careful about is regarding the handling of $\tau$-multiples. For example, given a monomial $x$, one thing that may happen is that all $x\tau^i$ exist in the table for all $i \geq 0$, but $x\tau^i$ supports a differential for $i > 0$. In this case, the only surviving element which contributes to our new column is $x$. 

Example 6.3. In stem 4, one has all \( \tau \)-multiples of 1 1 1 1. However, between stem 4 and stem 5, there is a differential going from the \( \tau \)-multiples of 2 2 1 to the \( \tau \)-multiples of 1 1 1 \( \tau \); thus, the only surviving element is 1 1 1 1.

As before, we do not yet have the information to determine the cell in row \( n = 2 \); we complete this after computing differentials.

6.2.2. Step 2: Computing differentials. Most of the care in transporting techniques from the classical Curtis algorithm to the \( \mathbb{C} \)-motivic setting is needed when computing differentials. The key point is that the differential is \( F_2[\tau] \)-linear;

\[
d(x\tau) = d(x)\tau,
\]

which in words means that if the boundary of \( x \) is \( y \), then the boundary of \( x\tau \) is \( y\tau \). Given that \( \tau \) is essentially a variable controlling motivic weight, this is the idea that lets us compute differentials in multiple motivic weights at once. There are some subtleties, however, which we will discuss now.

Once we have filled in column \( k \) for rows \( n > 2 \), we move to determining differentials out of this column. It is again important that we start work top to bottom; start with row \( n = 3 \) and work the way down.

1. Given a class \( x = \lambda_{i_1}\lambda_{i_2} \ldots \lambda_{i_m}\tau^i \) in our table, compute \( d(x) \). If \( d(x) = 0 \), we do nothing; \( x \), along with any \( \tau \)-multiples of it, are candidates to be permanent cycles.

   Otherwise, let \( y = \lambda_{j_1}\lambda_{j_2} \ldots \lambda_{j_{m+1}}\tau^j \) be the lexicographically leading term of \( d(x) \). If \( y \) is in the table and does not already support a differential, there is a differential taking the class \( x \) to the class \( y \). We then need to see what this implies for any \( \tau \)-multiples of \( x \) in the table. There are three cases which can happen:
   (a) All \( \tau \)-multiples of \( x \) kill all \( \tau \)-multiples of \( y \).
   (b) All \( \tau \)-multiples of \( x \) kill \( \tau \)-multiples of \( y \), but there are some which are left unkill.
   (c) All \( \tau \)-multiples of \( y \) are killed by \( \tau \)-multiples of \( x \), but there are some which do not kill.

   Assume the relevant “\( \tau \)-towers” in the table are \( x(k) \) and \( y(l) \), where \( y' = \lambda_{j_1}\lambda_{j_2} \ldots \lambda_{j_{m+1}}\tau^c \) for \( j - \ell \leq c \leq j \). If \( c \neq j \), write \( y(\ell - j + c) \) into the table. Draw a differential from the tower \( x(k) \) to \( y(\ell - j + c) \). We then break into the cases discussed above.
   (a) If \( k = \ell - j + c \), do nothing else.
   (b) If \( k < \ell - j + c \), we add \( y\tau^k(\ell - j + c - k) \) to the table, to mark the \( \tau \)-multiples of \( y \) which are unkill.
   (c) If \( k > \ell - j + c \), we add \( x\tau^{j-c}(k - \ell + j - c) \) to the table, to mark the \( \tau \)-multiples of \( x \) which do not kill.

2. If \( y \) is not in the table, we want to find the longest tail segment of \( y, \lambda_{j_h} \ldots \lambda_{m+1} \), which has a differential going into it. When we have this, let \( z \) denote the element in the table which hits it under the differential. We then have some cases that could happen for the \( \tau \)-multiples of \( x \) in the table.
   (a) All \( \tau \)-multiples of \( x \) are going to be completed to a cocycle using \( \tau \)-multiples of \( z \).
   (b) Some \( \tau \)-multiples of \( x \) are not completed to cocycles using \( \tau \)-multiples of \( z \).
Let $z$ be part of a “$\tau$-tower” $z_\ell$, where $z_\ell = \lambda_j \cdots \lambda_{m+1} \tau^c$, $j - \ell \leq c \leq j$. The cases above are represented computationally as follows:

(a) If $k \leq \ell - j + c$, do nothing.

(b) If $k > \ell - j + c$, add $x\tau^{j-c}(k - \ell + j - c)$ to the table.

We then start over at step 1, replacing $x$ with $x + \lambda_i \cdots \lambda_{i-1} z$: compute its differential, check if its leading term exists in the table...

We repeat this for all classes in our table.

7. Future directions

We briefly discuss some future directions for research relating to this project.

7.1. Complexity and implementation of algorithms. We have worked on a preliminary implementation the classical Curtis algorithm using Python, automating generation of Curtis tables. Our program, when run on a personal computer, throttled in performance at about stem 26. It is unknown as to what step in the process caused performance to dip drastically, and a closer analysis of the algorithm may reveal ways to improve implementation. Further, any computational insights gained from analyzing the classical algorithm will likely help streamline the implementation of the $C$-motivic algorithm, which we are currently working on.

7.2. Generalization to $R$-motivic setting. The $C$-motivic Adams $E_2$ page serves as the input for a spectral sequence converging to $R$-motivic Adams $E_2$ page, known as the $\tau$-Bockstein spectral sequence. This spectral sequence can be constructed from a filtered complex in the same way as the EHP spectral sequences in this paper, which means that our explicit understanding of the differentials in these spectral sequences can prove to be useful in understanding differentials in this Bockstein spectral sequence. In particular, it appears that there may be an “$R$-motivic Curtis algorithm”, for computing in this spectral sequence. If true, this may provide a new efficient way to start computing $R$-motivic stable stems.

References


