THE SIMPLICIAL COMPLEXES PACKAGE FOR MACAULAY2

BEN HERSEY, GREGORY G. SMITH, AND ALEXANDRE ZOTINE

ABSTRACT. This article demonstrates some of the updated features of the `SimplicialComplexes` package in *Macaulay2*.

This paper gives an overview of the `SimplicialComplexes` package in *Macaulay2* [M2]. The previous version of this package was developed by Sorin Popescu, Mike Stillman, and Gregory G. Smith in 2010. We have taken the opportunity to clean up the package and perform some maintenance to ensure that it stays computationally relevant and viable. We have also implemented a new data type in `SimplicialMap` in order to work with simplicial maps, and several constructors for well-known complexes.

In this package, simplicial complexes are represented algebraically through Stanley–Reisner ideals: Let Δ be an abstract simplicial complex with vertex set $V = \{v_0, v_1, \dots, v_n\}$, k be any commutative ring, and set $S = k[x_0, \dots, x_n]$. Then the **Stanley–Reisner ideal**, or **facet ideal** of Δ is defined to be square-free monomial ideal

$$I_\Delta \coloneqq \left(\prod_{j=1}^{k} x_{i_j} \,\middle|\, \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\} \not\subset \Delta \right) \subset S,$$

and the **Stanley–Reisner ring** corresponding to Δ is $k[\Delta] = S/I_\Delta$. In other words, I_Δ consists of generators corresponding to the *non-faces* of Δ. This produces a one-to-one correspondence between simplicial complexes and square-free monomial ideals, which makes it useful for computation in *Macaulay2*.

Example 0.0. There are two ways of constructing a simplicial complex, one by listing the products of the variables forming faces, or by constructing the Stanley–Reisner ideal (which has generators given by the *non-faces*). For instance, we can construct the 1-skeleton of a 2-simplex using both methods:

```
i2 : S = ZZ[x_0,x_1,x_2];
i3 : Δ = simplicialComplex {x_0*x_1, x_0*x_2, x_1*x_2}
o3 = simplicialComplex {x x , x x , x x }
                          1 2   0 2   0 1
o3 : SimplicialComplex
i4 : IΔ = monomialIdeal {x_0*x_1*x_2};
i5 : Δ' = simplicialComplex IΔ
o5 = simplicialComplex {x x , x x , x x }
                          1 2   0 2   0 1
```

Date: 2021–07–15.
2020 *Mathematics Subject Classification*. 05E45, 13F55, 55U10.

FIGURE 1. The simplicial complex ⧓ (left) and its Alexander dual ⧓* (right).

```
o5 : SimplicialComplex
```

In Section 1, we demonstrate some of the topological features of the package. Namely, the use of the small manifolds database provided by Lutz [Lut17], and some homological computations. In Section 2 we review some Stanley–Reisner theory and describe how to work with simplicial complexes using their corresponding Stanley–Reisner ideal. We also cover f- and h-vectors, shellability, and being Cohen-Macaulay. We give examples on how to check if a simplicial complex is Cohen-Macaulay using the package.

Finally, in Section 3, we overview the some chain complex constructions arising from simplicial complexes. We give an example of ideal homogenization and resolutions supported on a simplicial complex, and we provide examples of Taylor, Lyubeznik, and Buchberger resolutions; as well as Scarf complexes.

1. **Constructors.** In this package, every simplicial complex Δ is defined relative to a polynomial ring, and the vertices of Δ will correspond variables of this ring. The most basic constructor for simplicial complexes is the `simplicialComplex` method, which accepts two types of inputs. The first type of input is a list of square-free monomials in a polynomial ring. The support of each of these monomials determines a face, and the output is a the smallest simplicial complex containing all of the faces. For example, let's construct "bowtie" complex ⧓, depicted in Figure 3.

```
i3 : R = QQ[x_0..x_4];
i4 : ⧓ = simplicialComplex{x_0*x_1*x_2, x_2*x_3*x_4}
o4 = simplicialComplex | x_2x_3x_4 x_0x_1x_2 |
o4 : SimplicialComplex
```

The second type of input is a monomial ideal I generated by square-free monomials, and the output is the simplicial complex whose Stanley-Reisner ideal is I. We can also determine the Stanley-Riesner ideal of a simplicial complex, using the `ideal` method. Let's compute ⋈ again, this time using the Stanley-Riesner ideal as input.

```
i5 : I⧓ = ideal ⧓
o5 = ideal (x x , x x , x x , x x )
             0 3   1 3   0 4   1 4
o5 : Ideal of R
i6 : ⧓′ = simplicialComplex I⧓;
i7 : ⧓ === ⧓′
o7 = true
```

If $I = (m_1, m_2, \dots, m_q) \subset S$ is a monomial ideal, with minimal generators $m_i = \prod x_j^{a_{ij}}$, then the **Alexander dual** of I is defined to be $I^* := \bigcap_{i=1}^{q} (x_0^{a_{i_1}},\ x_1^{a_{i_2}}, \dots,\ x_{n-1}^{a_{i_{n-1}}})$. If $I = I_\Delta$ for some simplicial complex Δ, then I^* is also a square-free monomial ideal and is the Stanley–Reisner ideal of a simplicial complex Δ^*, which we call the **Alexander dual** complex to Δ. There is also a combinatorial description of Δ^*, given by $\Delta^* = \{F \subset V \mid V \setminus F \notin \Delta\}$. The Alexander dual to ⧓ is also given in Figure 3 and we can compute it in Macaulay2 using the `dual` method.

```
i133 : dual ⋈
o133 = simplicialComplex | x_1x_2x_4 x_0x_2x_4 x_1x_2x_3 x_0x_2x_3 |
o133 : SimplicialComplex
i134 : dual(monomialIdeal ⋈) == monomialIdeal dual ⋈
o134 = true
```

For a simplicial complex Δ and a face $F \in \Delta$, we define the **link** of F to be the subcomplex of Δ defined by $\mathrm{link}_\Delta(F) := \{G \in \Delta \mid F \cup G \in \Delta \text{ and } F \cap G = \varnothing\}$. For example The link of v_2 in ⧓* is the 4-cycle graph induced on the remaining vertices.

```
i18 : link(dual ⧓, x_2)
o18 = simplicialComplex | x_1x_4 x_0x_4 x_1x_3 x_0x_3 |
o18 : SimplicialComplex
i19 : link(dual ⧓, x_2) === inducedSubcomplex(dual ⧓, {x_0,x_1,x_3,x_4})
o19 = true
```

There are also constructors for the star of a face in a simplicial complex (`star`), the wedge product of simplicial complexes (`wedge`), the join of simplicial complexes (`*`), the elementary collapse of a free face (`elementaryCollapse`), and the barycentric subdivision of a simplicial complex (`barycentricSubdivision`). There are also constructors for some well known examples. As an example we construct the Ziegler ball.

```
i34 : R = QQ[a..j];
i35 : zieglerBallComplex R
o35 = simplicialComplex | bfgj bcgj befj abej abcj aehi adhi aefi abfi abdi
      cfgh degh cdgh bcfh adeh bcdh bcfg adeg acdg abef abcd |
o35 : SimplicialComplex
```

Lutz has provided a database enumerating all of the 2 and 3-manifolds having 10 or less vertices [Lut17]. We have implemented these databases into the package—however we have excluded the database of 3-manifolds with 10 vertices, due to the large number of examples causing long loading times.

These databases can be used to find nice testbeds of examples: for instance, we can search for simplicial maps

```
i2 : R = ZZ[a..i];
i3 : S = ZZ[x_0..x_6];
i4 : Γ = smallManifold(2,7,1,S);
i5 : maplist = flatten for i to 2 list (
     for j in subsets(toList(R_0..R_8),7) list (
         phi := map(smallManifold(3,9,i,R),Γ,j);
         if isWellDefined phi then phi else continue
```

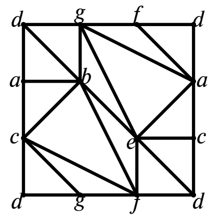

FIGURE 2. The minimal triangulation of the torus. Image credit: [Lut08]

```
        )
    );
i6 : maplist_0
o6 = | a b e f g h i |
```

By construction, all of these maps should be inclusions.

```
i7 : isInjective\maplist
o7 = {true, true, true, true, true, true}
o7 : List
```

The database also contains many triangulations of various interesting surfaces, such as the torus, Klein bottle, and real projective plane. Below are the smallest indices (and hence minimal triangulations of) these surfaces in the database, and see Figure 2 for a visual representation of the torus triangulation.

```
i8 : Torus = smallManifold(2, 7, 6, R);
i9 : KleinBottle = smallManifold(2, 8, 12, R);
i10 : RP2 = smallManifold(2, 6, 1, R);
```

We can check that these are the right surfaces by computing their homology. Theorems 6.2–6.4 from Munkres confirm that they match [Mun18].

```
i11 : for i to 2 list prune HH_i Torus
            2    1
o11 = {0, ZZ , ZZ }
o11 : List
i12 : for i to 2 list prune HH_i KleinBottle
o12 = {0, cokernel | 2 |, 0}
                   | 0 |
o12 : List
i13 : for i to 2 list prune HH_i RP2
o13 = {0, cokernel | 2 |, 0}
o13 : List
```

We can explicitly identify the generators of the homology by mapping circles onto the torus.

```
i6 : Circle = skeleton(1, simplexComplex(2,T));
i7 : f1 = map(Torus,Circle,matrix{{R_3,R_6,R_5}});
o7 : prints are being ugly
i8 : prune homology(1, f1)
o8 = | 1 |
```

FIGURE 3. The simplicial complex $\bowtie$ (left) and its Alexander dual $\bowtie^*$ (right).

```
      | 0 |
                 2         1
o8 : Matrix ZZ  <--- ZZ
i9 : f2 = map(Torus,Circle,matrix{{R_3,R_0,R_2}});
o9 : prints are being ugly
i10 : prune homology(1, f2)
o10 = | 0 |
       | 1 |
                  2         1
o10 : Matrix ZZ  <--- ZZ
```

2. **Stanley–Reisner Theory.** Stanley–Reisner theory connects homological properties of $k[\Delta]$ to combinatorial and topological properties of Δ. We will discuss some of the connections here, but a survey of results can be found in [BH93, Sta96, MS05].

Let Δ be an abstract simplicial complex with vertex set $V = \{v_0, v_1, \ldots, v_n\}$, let k be a commutative ring, and let $S = k[x_0, x_1, \ldots, x_n]$. The **Stanley–Reisner ideal** of Δ is defined to be square-free monomial ideal $I_\Delta := \left(\prod_{j=1}^{k} x_{i_j} \mid \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\} \not\subset \Delta\right) \subset S$ and the **Stanley–Reisner ring** corresponding to Δ is $k[\Delta] = S/I_\Delta$. This correspondence between simplicial complexes and square-free monomial ideals is one-to-one. Stanley–Reisner theory connects homological properties of $k[\Delta]$ to combinatorial and topological properties of Δ. A survey of results can be found in [BH93, Sta96, MS05].

If $I = (m_1, m_2, \ldots, m_q) \subset S$ is a monomial ideal, with minimal generators $m_i = \prod x_j^{a_{i_j}}$, then the **Alexander dual** of I is defined to be $I^* := \bigcap_{i=1}^{q} (x_0^{a_{i_1}},\ x_1^{a_{i_2}}, \ldots,\ x_{n-1}^{a_{i_{n-1}}})$. If $I = I_\Delta$ for some simplicial complex Δ, then I^* is also a square-free monomial ideal and is the Stanley–Reisner ideal of a simplicial complex Δ^*, which we call the **Alexander dual** complex to Δ. There is also a combinatorial description of Δ^*, given by $\Delta^* = \{F \subset V \mid V \setminus F \notin \Delta\}$. One of the attractive features of Alexander duality is the relationship between the cohomology of Δ and the homology of Δ^*. More specifically, if Δ is a simplicial complex on n vertices, then $\widetilde{H}_{i-1}(\Delta^*) = \widetilde{H}^{n-2-i}(\Delta)$ for all $i \in \mathbb{Z}$, see [MS05, Theorem 5.6].

Example 2.1. Consider the simplicial complex $\bowtie$, depicted in Figure 3. The Stanley–Reisner ideal of $\bowtie$ is $I_\bowtie = (x_0x_1x_2,\ x_0x_3,\ x_1x_3,\ x_0x_4,\ x_1x_4,\ x_2x_3x_4)$. We can exhibit the correspondence between $\bowtie$ and $I_\bowtie$ using the methods `simplicialComplex` and `ideal`.

```
i28 : S = QQ[x_0..x_4];
i29 : I⋈ = monomialIdeal(x_0*x_1*x_2,x_0*x_3,x_1*x_3,x_0*x_4,x_1*x_4,x_2*x_3*x_4);
o29 : MonomialIdeal of S
```

```
i30 : ⋈ = simplicialComplex I⋈
o30 = simplicialComplex | x_3x_4 x_2x_4 x_2x_3 x_1x_2 x_0x_2 x_0x_1 |
o30 : SimplicialComplex
i31 : I⋈ == ideal ⋈
o31 = true
```

We can use the `dual` method to compute the Alexander dual of $\bowtie$.

```
i133 : dual ⋈
o133 = simplicialComplex | x_1x_2x_4 x_0x_2x_4 x_1x_2x_3 x_0x_2x_3 |
o133 : SimplicialComplex
i134 : dual(monomialIdeal ⋈) == monomialIdeal dual ⋈
o134 = true
```

Moreover, we can verify the combinatorial description of $\bowtie^*$ by showing that the minimal generators of $I_{\bowtie}$ correspond the complements of the facets of $\bowtie^*$.

```
i140 : dualFacets = first entries facets dual ⋈
o140 = {x x x , x x x , x x x , x x x }
         1 2 4   0 2 4   1 2 3   0 2 3
o140 : List
i141 : sort first entries gens I⋈ == sort for F in dualFacets list(
           product for v in vertices ⋈ list(
               if member(v, support F) then continue else v)
               )
           )
o141 = true
```

Finally, we exhibit the isomorphisms between the cohomology of $\bowtie$ and the homology of $\bowtie^*$.

```
i94 : all(-1..5, i -> all(-1..5, i -> prune HH^(3-i) ⋈ == prune HH_(i-1) dual ⋈)
o94 = true
```

For a simplicial complex Δ and a face $F \in \Delta$, we define the **link** of F to be the subcomplex of Δ defined by $\operatorname{link}_\Delta(F) := \{G \in \Delta \mid F \cup G \in \Delta \text{ and } F \cap G = \varnothing\}$. We can now exhibit a more substantive result of Stanley–Reisner theory, which is the "dual version" of Hochster's formula, see [MS05, Corollary 1.40]. This formula allows us to compute the multigraded Betti numbers of I_Δ, which are defined to be $\beta_{i,m}(I_\Delta) = \operatorname{Tor}_i^S(I_\Delta, k)_m$. For a subset $F \subset V$, we will use the notation $\beta_{i,F}$ to refer to the Betti number in homological degree i and multidegree $(a_1, a_2, \dots, a_n) \in \mathbb{Z}^n$, where $a_i = 1$ if $v_i \in F$ and $a_i = 0$ otherwise.

Theorem 2.2 (Hochster's Formula, dual version)**.** *Let Δ be a simplicial complex with vertex set $V = \{v_0, v_1, \dots, v_n\}$ and $S = k[x_0, x_1, \dots, x_n]$, where k is a field. The nonzero multigraded Betti numbers of I_Δ and S/I_Δ lie in squarefree degrees. Moreover, if $F \subset V$, then*

$$\beta_{i,F}(I_\Delta) = \beta_{i+1,F}(S/I_\Delta) = \dim_k \widetilde{H}_{i-1}\big(\operatorname{link}_{\Delta^*}(V \setminus F); k\big).$$

Example 2.3. In Example 2.1, we computed the Alexander dual of the complex $\bowtie$. We can use the `link` method to compute the links of various faces. The link of the central vertex v_2 is a square.

```
i27 : link(dual ⋈, x_2)
o27 = simplicialComplex | x_1x_4 x_0x_4 x_1x_3 x_0x_3 |
o27 : SimplicialComplex
```

We can also construct a function that computes the multigraded betti numbers $\beta_{i,F}(S/I_{\bowtie})$.

```
hochster = (i, m) -> (
    G := product select(vertices ⋈, v -> not member(v, support m));
    rank (homology link(dual ⋈, G_S))_i
    )
```

Since we have a bound on the nonzero betti numbers of I_Δ, we can collect them into a matrix

```
i94 : V = vertices ⋈;
i95 : squarefreeMonomials = unique sort apply(remove(subsets V, 0), m -> lcm m);
i96 : matrix for i to length res I⋈ - 1 list (
          for F in squarefreeMonomials list hochster(i-1,F)
          )
o96 = | 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 |
      | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 1 1 0 |
      | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 2 |
                   3        31
o96 : Matrix ZZ  <--- ZZ
```

where the rows are indexed by the homological degree (starting at 0), and the columns are indexed by the squarefree multidegrees.

We say that a simplicial complex Δ is **pure** if all of the facets of Δ have the same dimension. We say that Δ is **shellable** if we can order the facets $F_1, \dots, F_m$ of Δ so that $\langle F_i \rangle \cap \langle F_1, F_2, \dots, F_{i-1} \rangle$ is a pure codimension one simplicial complex. We say that a simplicial complex Δ is **Cohen-Macaulay** if $k[\Delta]$ is a Cohen-Macaulay ring. It is known that every shellable simplicial complex is Cohen-Macaulay [BH93, Theorem 5.1.13] and every Cohen-Macaulay simplicial complex is pure [BH93, Corollary 5.1.5].

For Stanley–Reisner rings, we can use the combinatorics of Δ to determine if $k[\Delta]$ is a Cohen-Macaulay ring. Specifically, $k[\Delta]$ is Cohen-Macaulay if and only if $\widetilde{H}_i(\mathrm{link}_\Delta(F)\,;\,k) = 0$ for all $F \in \Delta$ and $i < \dim \mathrm{link}_\Delta(F)$, see [BH93, Corollary 5.3.9]. We can also relate the Cohen-Macaulay property to the h-vector of Δ. We can write the Hilbert series of $k[\Delta]$ as a rational function

$$\sum \dim\, k[\Delta]_i\, t^i = \frac{h_0 + h_1 t + \cdots h_k t^k}{(1-t)^{d+1}}$$

where $d = \dim \Delta$, and $0 \le k \le d+1$ is such that $h_k \neq 0$. If Δ is a d-dimensional Cohen-Macaulay complex with n vertices, then [BH93, Lemma 5.1.10] proves that

$$0 \le h_i \le \binom{n-d+i}{i}$$

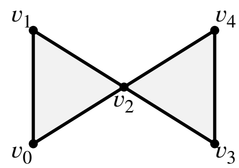

FIGURE 4. The bowtie complex

for $i = 0, 1, \ldots, d+1$. We can also compute the h-vector using the combinatorics of Δ. Specifically, if Δ is a d-dimensional simplicial complex, then [BH93, Lemma 5.1.8] tells us that

$$h_j = \sum_{i=0}^{j} (-1)^{j-i} \binom{d+1-i}{j-i} f_{i-1}$$

for $j = 0, 1, \ldots, d+1$.

Example 2.4. The simplicial complex $\bowtie$ introduced in example 2.1 is shellable and Cohen-Macaulay. We can verify the Cohen-Macaulay property using [BH93, Corollary 5.3.9].

```
i43 : faceList = flatten for i from -1 to dim ⋈ list (faces ⋈)#i
o43 = {1, x , x , x , x , x , x x , x x , x x , x x , x x , x x }
           0   1   2   3   4   0 1   0 2   1 2   2 3   2 4   3 4
o43 : List
i44 : all(faceList, F -> all(0..dim(link(⋈,F))-1, i -> HH_i link(⋈,F) == 0))
o44 = true
```

However, this package does not provide a method for checking that Δ is shellable. Such a method is available in the `SimplicialDecomposability` package in `Macaulay2`, see [Coo10].

Let us also consider the "bowtie" simplicial complex ⧓, which is obtained by filling in the loops of the simplicial complex $\bowtie$, see Figure 4. Since the intersection of the two facets of ⧓ is a single point, which has codimension two, the bowtie complex is not shellable. The simplicial complex ⧓ is not Cohen-Macaulay either, which we can verify by computing the h-vector of ⧓.

```
i3 : I⧓ = monomialIdeal(x_0*x_3, x_1*x_3, x_0*x_4, x_1*x_4 );
o3 : MonomialIdeal of S
i4 : ⧓ = simplicialComplex I⧓;
i5 : k⧓ = S/I⧓;
i6 : d = dim ⧓;
i7 : f⧓ = fVector ⧓
o7 = {1, 5, 6, 2}
o7 : List
i8 : h⧓ = for j to d+1 list(
         sum for i to j list (-1)^(j-i)*binomial(d+1-i,j-i)*(f⧓#(i))
         )
o8 = {1, 2, -1, 0}
o8 : List
```

Since the h-vector of ⧓ has a negative entry, this simplicial complex cannot be Cohen-Macaulay, by [BH93, Lemma 5.1.10].

Some other notable examples of simplicial complexes that are Cohen-Macaulay, but not shellable, are the Rudin ball and the Ziegler ball. Both of these complexes are included in the package and can be constructed using the methods `rudinBallComplex` and `zieglerBallComplex` respectively.

3. **Resolutions of Monomial Ideals.** Let Δ be a simplicial complex with q vertices and let $\widetilde{C}(\Delta;k)$ be the augmented chain complex of Δ with incidence function ε. We will let $S = k[x_0,\dots,x_n]$ be a polynomial ring over the commutative ring k, which is typically a field. We assume that the ring S has the fine $\mathbb{N}^{n+1}$ grading.

For a monomial ideal $I \subset S$, minimally generated by $m_1, m_2, \dots, m_q$, a **labelling** of Δ by I is a bijection which assigns a minimal generator of I to each vertex of Δ. Without loss of generality, we will assume this assignment is $v_i \longmapsto m_i$ for $i = 1,2,\dots,q$. For each face $F \in \Delta$ we can construct the monomial $m_F = \operatorname{lcm}(m_i \mid i \in F)$. The I**-homogenization** of $\widetilde{C}(\Delta;k)$ is the chain complex $(\mathbf{G}, d)$ such that

$$G_0 = S \;\text{ and }\; G_i = \bigoplus_{\dim(F)=i-1} S(m_F).$$

We will use $\{f_F \mid F \in \Delta\}$ to denote the canonical basis of G and we define the differential of $\mathbf{G}$ using

$$d(f_F) \;=\sum_{\dim(F)=i-1} \frac{m_F}{m_{F'}} \cdot \varepsilon(F,F') f_{F'}.$$

We say that I has a **resolution supported on** Δ if the I-homogenization of $\widetilde{C}(\Delta;k)$ is a free resolution for some labelling of the vertices of Δ. For further details on I-homogenization, we refer the reader to any one of [BPS98, MS05, Pee11, PV11].

Example 3.5. Let $S = \mathbb{Q}[x_0,x_1,x_2,x_3]$, let $I = (x_0x_1, x_0x_2, x_0x_3, x_1x_2x_3)$, and let Γ be the simplicial complex with facets $\{v_1,v_2,v_3\}$ and $\{v_2,v_4\}$. If we use the labelling $v_1 \longmapsto x_0x_1$, $v_2 \longmapsto x_0x_2$, $v_3 \longmapsto x_0x_3$, and $v_4 \longmapsto x_1x_2x_3$, we get the labelled simplicial complex shown in Figure 5. We can

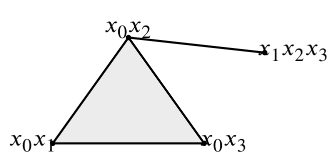

FIGURE 5. The homogenization of Γ

use `Macaulay2` to compute both $\widetilde{C}(\Delta;\mathbb{Q})$ and the I-homogenization of Δ relative to this labelling, which is the minimal free resolution of S/I.

```
i9 : R = ZZ/101[y_0..y_13];
i10 : S = QQ[x_0..x_3];
i11 : Δ = simplicialComplex{R_0*R_1*R_2, R_2*R_3};
i12 : I = ideal(x_0*x_1,x_0*x_2,x_0*x_3,x_1*x_2*x_3);
o12 : Ideal of S
i13 : C = chainComplex Δ
        ZZ 1        ZZ 4        ZZ 4        ZZ 1
o13 = (---)  <-- (---)  <-- (---)  <-- (---)
```

```
         101          101          101          101
       -1           0            1            2
o13 : ChainComplex
i24 : G = chainComplex(Δ, Labels => {x_0*x_1,x_0*x_2,x_0*x_3,x_1*x_2*x_3});
i25 : G.dd
           1                                          4
o25 = 0 : S  <------------------------------------- S  : 1
                | x_0x_1 x_0x_2 x_0x_3 x_1x_2x_3 |
           4                                      4
      1 : S  <---------------------------------- S  : 2
               {2} | -x_2 -x_3 0    0       |
               {2} | x_1  0    -x_3 0       |
               {2} | 0    x_1  x_2  -x_1x_2 |
               {3} | 0    0    0    x_0     |
           4                    1
      2 : S  <---------------- S  : 3
               {3} | x_3  |
               {3} | -x_2 |
               {3} | x_1  |
               {4} | 0    |
o25 : ChainComplexMap
i15 : (res (S^1/I)) == G
o15 = true
```

The I-homogenization of Δ is dependent on how you label the vertices, and that is reflected by the ordering of the monomials in the `Labels` argument. Indeed, if we swap the labels of v_1 and v_4, then the I-homogenization is no longer a resolution of S/I.

```
i22 : G' = chainComplex(Δ, Labels => {x_1*x_2*x_3,x_0*x_2,x_0*x_3,x_0*x_1});
i23 : prune homology G'
o23 = 0 : cokernel | x_0x_3 x_0x_2 x_0x_1 x_1x_2x_3 |
      1 : cokernel {3} | x_3 |
      2 : 0
      3 : 0
o23 : GradedModule
```

Given a monomial ideal I, there are several algorithms that will produce a simplicial complex Δ and a labelling of Δ by the minimal generators of I such that the I-homogenization of $\widetilde{C}(\Delta;k)$ is a free resolution of S/I, though often non-minimally. Examples of such constructions are the Taylor resolution, Lyubeznik resolution, and the Buchberger resolution, all of which are implemented in `SimplicialComplexes`. We have also implemented a constructor for the Scarf complex, which is a complex that is not always a free resolution of S/I, but when it is a free resolution it is minimal. We will not describe these constructions here, but a concise description of the Taylor resolution, Lyubeznik resolution, and Scarf complex is given in [Mer12], and a description of the Buchberger resolution is given in [OW16].

Example 3.6. Consider the monomial ideal $I = (x_1x_3, x_2^2, x_0x_2, x_1^2, x_0^2) \subset \mathbb{C}[x_0,\dots,x_3]$. The Taylor resolution of I can be realized as an I-homogenization of the 4-simplex.

```
i2 : R = QQ[a,b,c,d,e];
i3 : S = QQ[x_0..x_3];
i4 : I = monomialIdeal(x_1*x_3, x_2^2,  x_0*x_2, x_1^2, x_0^2);
o4 : MonomialIdeal of S
i5 : T = taylorResolution J
      1      5      10      10      5      1
o5 = S  <-- S  <-- S   <-- S   <-- S  <-- S
     0      1      2       3       4      5
o5 : ChainComplex
i6 : T == chainComplex(simplexComplex(4,R),Labels => first entries mingens I)
o6 = true
```

The Buchberger simplicial complex is a subcomplex of the 4-simplex, and the Buchberger resolution is an I-homogenization of the Buchberger simplicial complex. For this example, the Buchberger resolution is the minimal free resolution of S/I, but this is not always the case.

```
i7 : buchbergerSimplicialComplex(J,R)
o7 = simplicialComplex | acde abcd |
o7 : SimplicialComplex
i8 : B = buchbergerResolution J
      1      5      9      7      2
o8 = S  <-- S  <-- S  <-- S  <-- S
     0      1      2      3      4
o8 : ChainComplex
i10 : betti B === betti(res J)
o10 = true
```

Lyubeznik simplicial complexes and resolutions are constructed relative to a total order on the minimal generators of I. Every ordering will produce a resolution, but these resolutions need not be isomorphic. When no ordering is given, the methods `lyubeznikSimplicialComplex` and `lyubeznikResolution` will order the generators relative to the monomial order on S which, in Macaulay2, is graded revlex by default. The option `MonomialOrder` reorders the minimal generators of I relative to the monomial ordering on S. For example, `MonomialOrder => {2,1,0,3,4}` refers to the total ordering $x_0x_2 < x_2^2 < x_1x_3 < x_1^2 < x_0^2$ on the minimal generators of I. We see that by changing the ordering we can both produce the worst case (Taylor resolution) and best case (minimal free resolution).

```
i11 : lyubeznikSimplicialComplex(J,R)
o11 = simplicialComplex | abcde |
o11 : SimplicialComplex
i12 : lyubeznikResolution(J) == taylorResolution(J)
o12 = true
i13 : lyubeznikSimplicialComplex(J, R, MonomialOrder => {2,1,0,3,4})
o13 = simplicialComplex | acde abcd |
o13 : SimplicialComplex
i14 : L = lyubeznikResolution(J,MonomialOrder => {2,1,0,3,4})
       1      5      9      7      2
o14 = S  <-- S  <-- S  <-- S  <-- S
      0      1      2      3      4
```

```
o14 : ChainComplex
```

The Scarf simplicial complex of I starts with the labelled 4-simplex and removes any faces F, F' such that $m_F = m_{F'}$. The I-homogenization of the Scarf simplicial complex is the Scarf chain complex. It is often the case that the Scarf chain complex is not a free resolution of S/I, but when it is a resolution, it is minimal, see [BPS98, Lemma 3.1].

```
i16 : scarfSimplicialComplex(J,R)
o16 = simplicialComplex | acde abcd |
o16 : SimplicialComplex
i17 : scarfChainComplex J == buchbergerResolution J
o17 = true
```

Acknowledgements. All three authors were partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[BH93] Winfried Bruns and Jürgen Herzog, *Cohen-Macaulay rings*, Cambridge Studies in Advanced Mathematics, vol. 39, Cambridge University Press, Cambridge, 1993.

[Sta96] R.P. Stanley, *Combinatorics and Commutative Algebra*, 2nd ed., Progress in Mathematics, vol. 41, Birkhäuser Boston, 1996.

[MS05] Ezra Miller and Bernd Sturmfels, *Combinatorial Commutative Algebra*, Graduate Texts in Mathematics, vol. 227, Springer-Verlag New York, 2005.

[Pee11] Irena Peeva, *Graded Syzygies*, Algebra and Applications, vol. 14, Springer-Verlag London, 2011.

[Mun18] James R. Munkres, *Elements Of Algebraic Topology*, CRC Press, 2018.

[M2] Daniel R. Grayson and Michael E. Stillman, *Macaulay2, a software system for research in algebraic geometry*, available at http://www.math.uiuc.edu/Macaulay2/.

[ÀMFRG20] Josep Àlvarez Montaner, Oscar Fernández-Ramos, and Philippe Gimenez, *Pruned cellular free resolutions of monomial ideals*, J. Algebra **541** (2020), 126–145, DOI 10.1016/j.jalgebra.2019.09.013. MR4014733

[BT09] Anders Björner and Martin Tancer, *Note: Combinatorial Alexander duality—a short and elementary proof*, Discrete Comput. Geom. **42** (2009), no. 4, 586–593, DOI 10.1007/s00454-008-9102-x. MR2556456

[Coo10] David Cook II, *Simplicial decomposability*, J. Softw. Algebra Geom. **2** (2010), 20–23, DOI 10.2140/jsag.2010.2.20. MR2881131

[PV11] Irena Peeva and Mauricio Velasco, *Frames and degenerations of monomial resolutions*, Trans. Amer. Math. Soc. **363** (2011), no. 4, 2029–2046, DOI 10.1090/S0002-9947-2010-04980-3. MR2746674

[BPS98] Dave Bayer, Irena Peeva, and Bernd Sturmfels, *Monomial resolutions*, Math. Res. Lett. **5** (1998), no. 1-2, 31–46, DOI 10.4310/MRL.1998.v5.n1.a3. MR1618363

[Mer12] Jeff Mermin, *Three simplicial resolutions*, Progress in commutative algebra 1, de Gruyter, Berlin, 2012, pp. 127–141. MR2932583

[OW16] Anda Olteanu and Volkmar Welker, *The Buchberger resolution*, J. Commut. Algebra **8** (2016), no. 4, 571–587, DOI 10.1216/JCA-2016-8-4-571. MR3566531

[Lut08] Frank H. Lutz, *Enumeration and Random Realization of Triangulated Surfaces*, Discrete Differential Geometry, 2008, pp. 235–253.

[Lut17] Frank H. Lutz, *The Manifold Page* (2017), http://page.math.tu-berlin.de/~lutz/stellar/.

Department of Mathematics and Statistics, Queen's University, Kingston, Ontario, K7L 3N6;
hersey.b@queensu.ca, ggsmith@mast.queensu.ca, 18az45@queensu.ca.