Simplicial Computations in Topology

Simplicial Computations in Topology

Carlos Alquézar Baeta



A thesis submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
University of Zaragoza
2026

To those who have suffered during the writing of this thesis.
To those who will suffer during the reading of this thesis.

Cerrando el círculo.

Outro. Kase.O.

Lo primerio de todo, gracias a Michel y Jorge, mis directores de tesis, que lo han tenido dificil y no han parado de ayudarme y empujarme para que siguiera adelante. Cuando leáis esto, que sepáis que tenéis que estar orgullosos porque habéis sido unos grandes directores. Me habéis ayudado tanto en lo académico como en lo personal. Ha costado más de lo que pensaba pero habéis estado siempre al pie del cañón en cualquier circunstancia y habéis sabido ser buenos guías y consejeros. Vuestra impronta se quedará conmigo. Gracias también a Enrique Artal, José Ignacio Cogolludo, Luis Ugarte, Álvaro Lozano y Edwin León. Ha sido un verdadero placer compartir este tiempo con vosotros. Gracias a Maite Lozano y Marta Macho por inspirarme en las clases de Topología Algebraica. Recuerdo con humor el primer día de clase cuando me preguntó Marta qué formación tenía de base. Cuando le dije que era ingeniero... su cara un poema. Pero sus clases, al igual que las de Maite, eran increíblemente estimulantes, y cuando hubo que tomar la decisión de empezar el doctorado, fueron buenas consejeras, aunque fuera en conversaciones de pasillo. No nos damos cuenta a veces que cuando hablamos con otras personas, la forma en la que nos expresamos puede estar influyendo en quien nos escucha más de lo que pensamos. En ese momento encontré a dos personas a las que admiro que me dijeron... inténtalo, y paso a paso. Gracias también a toda la crew de doctorandos que han llegado los últimos años y que han revolucionado el área de Geometría y Topología. Especial mención a Marta Aldasoro, compañera de cierre de tesis que me ha inspirado para apretar dientes y darlo todo en esta recta final. Cruzando fronteras, gracias a Giovanni Bazzoni por el tiempo que compartimos en Como y que guardo con muy buen recuerdo. Fue una gran experiencia tanto en lo académico como en lo personal. Gracias también a Aniceto Murillo por toda la ayuda en las últimas semanas antes de entregar la tesis y darme las claves que necesitaba. Gracias a José Brox por todo el apoyo y cariño que siempre me transmite y que es mutuo. Y para terminar este apartado, mención especial a Luis Rández. Él fue clave en un momento decisivo y le estaré eternamente agradecido. Ante una situación difícil estuvo a la altura y fue refugio. Gracias Luis.

Gracias a todos los compañeros y compañeras del Departamento de Matemáticas y Computación de la Universidad de La Rioja. Fue una gran experiencia y aprendí mucho esos años a vuestro lado. Gracias de corazón por vuestra acogida y el tiempo que compartimos juntos. En especial, gracias a Manuel Higueras, uno de mis grandes apoyos en esa época y con el que, tras muchas conversaciones vespertinas (a veces acompañadas de unas refrescantes jarras de cerveza) en las que nos reíamos y debatíamos sobre cómo sobrellevar la insoportable levedad del ser, forjamos una gran amistad.

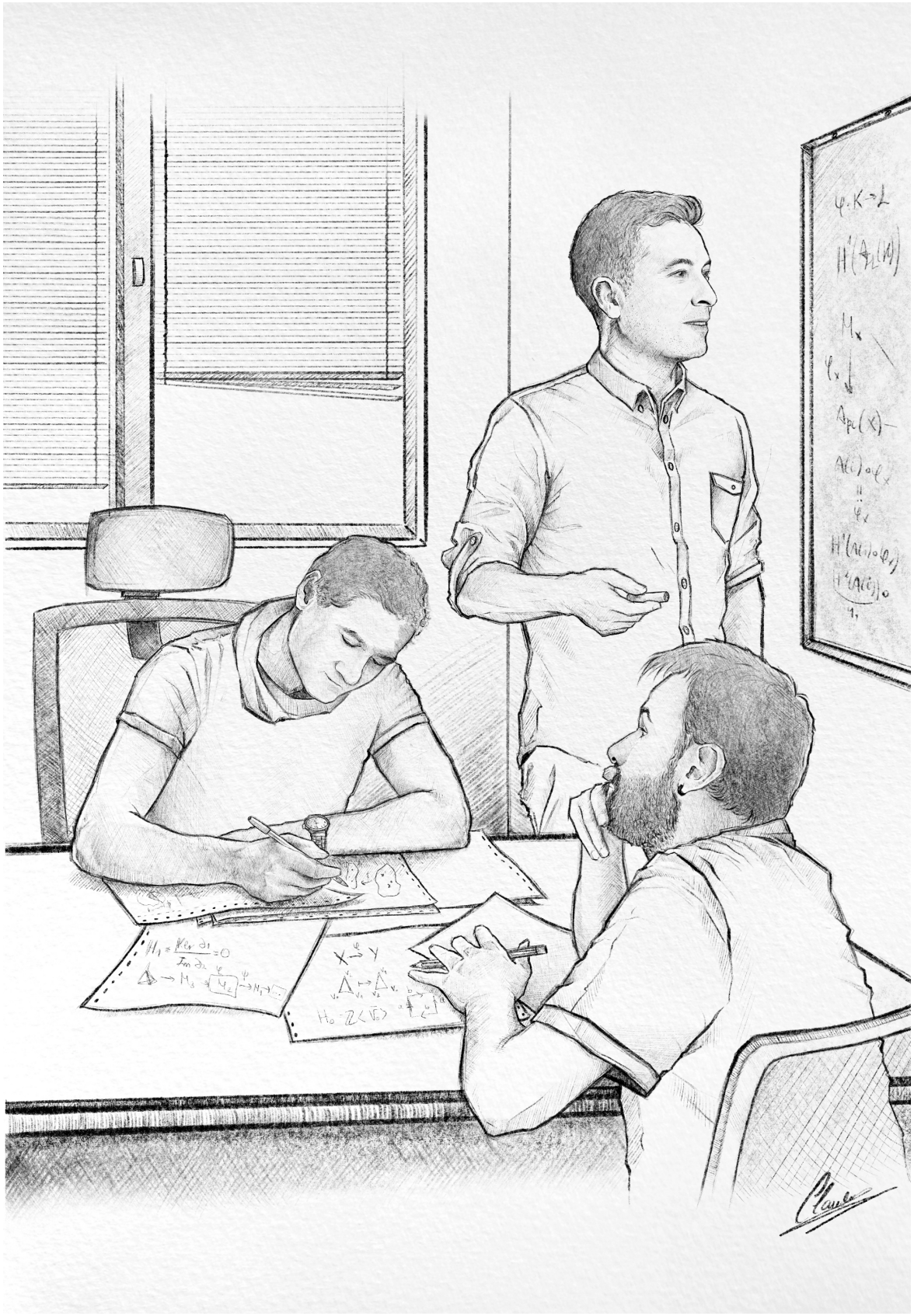
Recordando mis orígenes, gracias a Francisco Serón por abrirme camino y darme la primera oportunidad. Sin duda fue un importante punto de inflexión que marcó mi devenir, tanto personal como profesionalmente. Gracias a Manuel G. Bedia por inspirarme y alentarme para que mi trayectoria fuera y siga siendo lo más caótica posible, inverosímil, frenética, extravagante, infame y estimulante. Faro referente, oráculo. Sé que en algún momento volverá a leer esto y conseguiré sacarle de nuevo una sonrisa. Gracias también a Miguel Aguilera por inspirarme y ayudarme en este camino. Siempre kilómetros por delante, inalcanzable. Lo considero mi hermano mayor académico. Un gran apoyo y, si he conseguido terminar, ha sido en parte gracias a la motivación que me da el poder agradecerle en algún momento toda la ayuda que me dió, sobre todo en los comienzos, cuando mi cabeza era una centrifugadora sin control. Ahora lo sigue siendo pero hemos conseguido ciertos periodos de lucidez. Estoy convencido de que en algún momento nuestros caminos académicos se volverán a cruzar. Tomás Gómez, Daniel Martínez, David, Jorge...buenas aventuras compartimos juntos.

Gracias a toda la familia del Triatlón Cierzo, que me enseñaron los valores del deporte, el esfuerzo y la constancia. No sé si todo pero algo seguro se me ha pegado. Los leones no sabemos rendirnos. Grandes amigos con los que seguro nos vemos más pronto que tarde nadando en pantanos, mares o subiendo montañas. La vida fluctúa pero hay que saber navegar en esas aguas. Gracias también a mi núcleo duro, a Silvia Gimeno y Pablo Barrecheguren, fieros compañeros de batalla, siempre en primera línea, sin protección y a cuerpo descubierto. Juntos somos más duros que una roca. Héctor estará leyendo esto también. Va por tí hermano. Gracias a mis amigos del colegio, que aunque no hayamos estado tan presentes los últimos tiempos, siempre me acuerdo de vosotros.

Gracias a mis padres, María Baeta Berdié y Joaquín Alquézar Alloza, por todo en general y, en particular, por haber entendido este último año la dificultad del proceso y darme el espacio que necesitaba. Ha costado llegar hasta aquí pero me habéis enseñado a pelear y a soportar la crudeza de la vida. También a relativizar los problemas y evadirse de ellos cuando hace falta. Gracias a mi segunda madre, Chon, por la inmensa ayuda y apoyo durante este tiempo. Estoy seguro de que Nacho estará muy orgulloso de vosotras y que nos acompañará siempre allá donde vayamos. Gracias también a mi hermana Sara, y a mi sobrina Iziar. Espero que el vernos esforzándonos los unos a los otros nos motive para seguir creciendo y mejorando en la vida. Gracias a mis tíos, Mariana y José, siempre presentes. Sé que se hubiesen alegrado mucho por leer esto. Fueron un pilar fundamental en mi vida y todavía me siguen inspirando y motivando para seguir creciendo. Gracias también a mi suegra Tere, por su apoyo en general y particularmente este último año, que sin pedir nada a cambio, velaba por que no me faltara de nada y pudiera seguir centrado en terminar este trabajo. Y por último, pero no por ello menos importante, gracias a Sandra y Germán, el mejor regalo que me ha podido tocar. Gracias por entenderme este último periodo y ponérmelo fácil para que pudiera cerrar esta etapa y construir una nueva. Gracias por vuestra paciencia. Os quiero mucho.

Amigos que se fueron me visitan a diario.
Estéis donde estéis, me acuerdo de
vosotros antes de salir en el backstage.
Para todos los que estáis, ya sabéis, que
lo que dais será lo que recibiréis. Que se
prepare el mundo porque voy a
intervenir. Tengo un mensaje para los
que se quieran venir: no son problemas,
son oportunidades, de mejorar y superar
adversidades. Tú tienes gran poder, solo
quiérete.
Puedes lograr cualquier cosa
esforzándote.

Outro. Kase.O.



CONTENTS

INTRODUCTION iii

Chapter I. Preliminaries 1
I.1 Commutative Differential Graded Algebras 1
I.2 Simplicial sets 7
I.3 The functor $\mathcal{K} \rightsquigarrow \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ 16

Chapter II. Revisiting the PL de Rham theory 25
II.1 Whitney elementary forms 25
II.2 Products of Whitney forms 29
II.3 Dupont's contraction 41

Chapter III. The minimal model of a finite simplicial set 45
III.1 Effective representation of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ elements 46
III.2 $\mathcal{C}(\mathcal{K})$ as a proxy to $\mathcal{H}(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$ 49
III.3 Primitives of exact differential forms on simplicial sets 52
III.4 Effective computation of Dupont's formula 60
III.5 The SullivanMM package: source code and documentation 61

Chapter IV. Examples 63
IV.1 Complement of smooth projective curves 63
IV.2 Formality of the $\mathcal{K}3$ Surface 65

CONCLUSIONS 69

Appendix 71

Chapter A. The Sullivan Models package 73

Chapter B. The homogeneous part of $\mathcal{M}^3$ for the $\mathcal{K}3$ surface 119

Chapter C. Example of an element $p \in \mathcal{A}^3_{\mathcal{PL}}(\mathcal{K}3)$ 127

BIBLIOGRAPHY 143

INTRODUCTION

The aim of this thesis is to provide an effective method (with a concrete implementation in a Computer Algebra System) for the computation of the Sullivan minimal model of a finite type simplicial set $\mathcal{K} = \{\mathcal{K}_n\}_{n\geq 0}$ ([**57, 22**]). This model is defined as the minimal model of the algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ of polynomial differential forms on $\mathcal{K}$. The relevance of this construction stems from the fact that it provides topological information, such as the rational homotopy type of certain spaces (called Sullivan spaces [**57, 9, 21, 24, 22**]).

For the case of a finitely presented connected Commutative Differential Graded Algebra (CDGA), Manero and Marco provided in [**47**] an effective method (with a concrete implementation) to compute its minimal model up to a given degree. However, since the homogeneous part of degree 0 of the algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ is infinite-dimensional, this method cannot be applied to the case we are interested in.

In this work, we present an adaptation of the previous method to make it suitable for $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. For this purpose, we need to introduce some notation and basic definitions. Afterward, the main contribution of this work will be described.

For readers not familiar with simplicial sets, since our method is valid for any kind of simplicial structure, we recommend thinking in terms of simplicial complexes instead of simplicial sets for the moment. We refer them to [**27**] and section I.2 for a proper definition of these structures. To facilitate understanding, we split the rest of the introduction into several parts.

Algebraic Topological models for cochain complexes. A brief description of an *Algebraic Topological model* (or AT-model) can be summarized as follows (for a detailed description and implementations see [**13, 35, 53, 52, 51**]): given a cochain complex (C, d_C), an AT-model of C is a tuple (M, π, ι, ϕ) where $(M, 0)$ is a cochain complex with trivial differential such that

$$M \cong \mathcal{H}(C),$$

π and ι are chain maps:

$$\begin{aligned} \pi: &\quad C \to M, \\ \iota: &\quad M \to C, \end{aligned}$$

and ϕ is a chain homotopy:

$$\phi : C^* \to C^{*-1},$$

such that

$$\begin{aligned} \pi \circ \iota &= \mathrm{id}_M, \\ \iota \circ \pi - \mathrm{id}_C &= d_C \circ \phi + \phi \circ d_C, \end{aligned}$$

Commutative Differential Graded Algebras. A *graded commutative* algebra (or GCA) is a graded algebra, $\mathcal{A}$, where the product of two homogeneous elements, x, y, of degrees i, j respectively, satisfies:

$$xy = (-1)^{ij} yx.$$

In addition, if we have a linear map of degree 1, $d : \mathcal{A} \to \mathcal{A}$, such that $d^2 = 0$ and it satisfies the Leibniz rule:

$$d(xy) = d(x)y + (-1)^i x d(y),$$

then we say that $(\mathcal{A}, d)$ is a *commutative differential graded algebra* (CDGA).

When $\mathcal{M}$ is a freely generated CDGA such that its generators can be ordered in a non-decreasing degree sequence in such a way that the differential of each generator can be expressed in terms of the previous ones, we say that the algebra is *minimal*. If $\mathcal{A}$ is a CDGA that is not minimal, then there exists a quasi-isomorphism (a morphism that induces an isomorphism in cohomology), $\varphi : \mathcal{M} \to \mathcal{A}$, from a minimal CDGA $\mathcal{M}$ to $\mathcal{A}$. Moreover, $\mathcal{M}$ is unique up to isomorphism, and $(\mathcal{M}, \varphi)$ is called the *Sullivan minimal model* of $\mathcal{A}$ ([**26, 21, 37**]).

The algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$[21]. Before defining this algebra, two structures must be introduced. First, consider the free graded commutative algebra $\bigwedge(t_0, \ldots, t_n, y_0, \ldots, y_n)$, where the generators t_i have degree 0, and the generators y_j have degree 1. The differential d is given by

$$\begin{aligned} d(t_i) &= y_i, \\ d(y_j) &= 0. \end{aligned}$$

The cochain algebra $(\mathcal{A}_{\mathcal{PL}})_n$ is defined as the quotient:

$$(\mathcal{A}_{\mathcal{PL}})_n := \frac{\bigwedge(t_0, \ldots, t_n, y_0, \ldots, y_n)}{\langle \sum_{i=0}^n t_i - 1, \sum_{j=0}^n y_j \rangle}.$$

Then, the simplicial cochain algebra $\mathcal{A}_{\mathcal{PL}} := \{(\mathcal{A}_{\mathcal{PL}})_n\}_{n \geq 0}$ is defined as the simplicial set such that the n-simplices are all the elements of the cochain algebra $(\mathcal{A}_{\mathcal{PL}})_n$, and the face and degeneracy morphisms are the unique cochain algebra morphisms:

$$\begin{aligned} \partial_i : &\quad (A_{PL})_n \to (A_{PL})_{n-1}, \quad 0 \leq i \leq n, \\ s_j : &\quad (A_{PL})_n \to (A_{PL})_{n+1}, \quad 0 \leq j \leq n, \end{aligned}$$

satisfying:

$$\partial_i(t_k) = \begin{cases} t_k, & k < i, \\ 0, & k = i, \\ t_{k-1}, & k > i, \end{cases} \qquad s_j(t_k) = \begin{cases} t_k, & k < j, \\ t_k + t_{k+1}, & k = j, \\ t_{k+1}, & k > j. \end{cases}$$

After the definition of these structures, the algebra

$$\mathcal{A}_{\mathcal{PL}}(\mathcal{K}) := \{\Phi : \mathcal{K} \to \mathcal{A}_{\mathcal{PL}}\}$$

is defined as the cochain algebra whose elements $\Phi \in \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ are collections of simplicial maps $\Phi = \{\Phi_n : \mathcal{K}_n \to (\mathcal{A}_{\mathcal{PL}})_n\}_{n \geq 0}$, commuting with face and degeneracy morphisms. This algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K}) = \bigoplus_{p \geq 0} \mathcal{A}^p_{\mathcal{PL}}(\mathcal{K})$ is, in fact, a CDGA, where addition, scalar multiplication, product, and the differential are induced by the corresponding operations in the algebras $(\mathcal{A}_{\mathcal{PL}})_n$. Denote by $d_{\mathcal{A}}$ its differential.

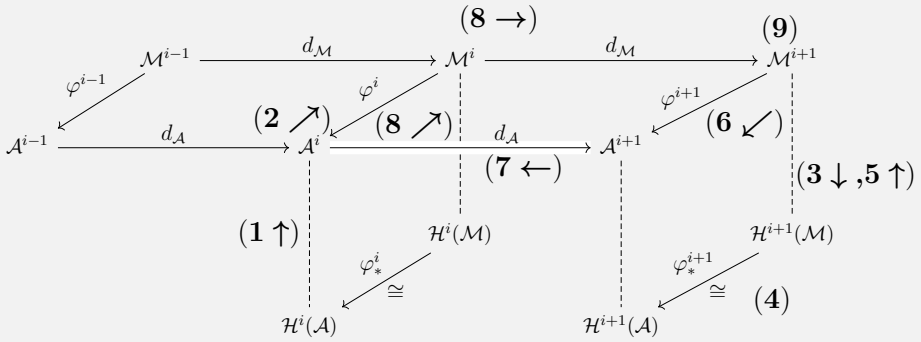
Let $\mathcal{C}(\mathcal{K}) = \bigoplus_{p\geq 0} \mathcal{C}^p(\mathcal{K})$ be the simplicial cochain complex associated with $\mathcal{K}$, and denote by $d_{\mathcal{C}}$ its differential. By de Rham [**15**], there exists a quasi-isomorphism of cochain complexes (but not cochain algebras) $\oint : \mathcal{A}_{\mathcal{PL}}(\mathcal{K}) \to \mathcal{C}(\mathcal{K})$ and, due to Whitney [**62, 43**], there exists, for this quasi-isomorphism $\oint$, a section $\mathcal{W} : \mathcal{C}(\mathcal{K}) \to \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, called the *Whitney map*, which is also a quasi-isomorphism of cochain complexes. In particular, the following diagram commutes:

$$\begin{array}{ccc} \mathcal{A}^{p-1}_{\mathcal{PL}}(\mathcal{K}) & \xrightarrow{d^{p-1}_{\mathcal{A}}} & \mathcal{A}^{p}_{\mathcal{PL}}(\mathcal{K}) \\ \oint \downarrow \uparrow \mathcal{W} & \circlearrowleft & \oint \downarrow \uparrow \mathcal{W} \\ \mathcal{C}^{p-1}(\mathcal{K}) & \xrightarrow[d^{p-1}_{\mathcal{C}}]{} & \mathcal{C}^{p}(\mathcal{K}) \end{array}$$

MMarco's implementation. As mentioned at the beginning, our starting point is the effective method previously implemented by Manero and Marco [**47**] for the computation of the minimal model of a finitely generated connected CDGA.

Remark. For the rest of the document, and in agreement with Victor Manero, as a gesture of gratitude and recognition to Miguel Ángel Marco, we will refer to this approach as **MMarco**'s method.

Key steps of MMarco's implementation



(1) Lifts generators $\mathcal{H}^i(\mathcal{A}) \to \mathcal{A}^i$.
(2) Updates $(\mathcal{M}^i, \varphi^i)$ in order to make φ^i_* surjective.
(3) Gets a basis of $\mathcal{H}^{i+1}(\mathcal{M})$.
(4) Computes the kernel of the induced morphism φ^{i+1}_*.
(5) Lifts the generators of $\ker(\varphi^{i+1}_*)$.
(6) For each generator $z \in \ker(\varphi^{i+1}_*)$, gets its image $\varphi(z) \in \mathcal{A}^{i+1}$.
(7) For each $\varphi(z) \in \mathcal{A}^{i+1}$, finds $\theta \in \mathcal{A}^i$ such that $d_{\mathcal{A}}(\theta) = \varphi(z)$.
(8) Updates $(\mathcal{M}^i, \varphi^i)$ in order to make φ^i_* injective.
(9) $(\mathcal{M}, \varphi)$ is now an i-minimal model of $\mathcal{A}$.

Arrow notation: having the needed elements → compute/modify/link to.

The method is constructive, i.e., it builds the model from lower to higher degrees, adding generators sequentially so that the model satisfies the required conditions. Let $(\mathcal{A}, d_{\mathcal{A}})$ be the

CDGA for which we want to obtain its minimal model, and let $(\mathcal{M}, d_{\mathcal{M}})$ initially be the trivial CDGA (with no generators). The method can be summarized as follows:

(1) The method takes a basis $[h_0^i], \ldots, [h_{l_0}^i]$ of $\mathcal{H}^i(\mathcal{A})$ of the first non-trivial cohomology group of $\mathcal{A}$ and lifts its generators; **(2)** the model under construction $(\mathcal{M}, \varphi)$ is updated at degree i with new generators $x_0^i, \ldots, x_{l_0}^i$ in such a way that the induced morphism $\varphi_*^i : \mathcal{H}^i(\mathcal{M}) \to \mathcal{H}^i(\mathcal{A})$ becomes surjective; **(3)** the method takes a basis of the cohomology of the model $\mathcal{M}$ at degree $i+1$ and **(4)** computes the kernel of the induced morphism φ_* to check if it is injective at degree $i+1$; in the case where $dim(\ker(\varphi_*^{i+1})) \neq 0$, **(5)** a basis of the kernel $[z_0^{i+1}], \ldots, [z_q^{i+1}]$ is taken and their generators are lifted to $\mathcal{M}^{i+1}$; for each lifted element z_j^{i+1}, **(6)** its image under the morphism $\varphi(z_j^{i+1}) \in \mathcal{A}^{i+1}$ is obtained and, as it is known by construction that these elements $\varphi(z_j^{i+1})$ are exact, **(7)** for each one the method finds an element $\theta_j^i \in \mathcal{A}^i$ such that $d(\theta_j^i) = \varphi(z_j^{i+1})$; finally, **(8)** the model $(\mathcal{M}, \varphi)$ is updated by adding, for each θ_j^i, a generator y_j^i such that $\varphi(y_j^i) = \theta_j^i$ and $d(y_j^i) = z_j^{i+1}$; at that point **(9)**, $(\mathcal{M}, \varphi)$ is an i-minimal model of $\mathcal{A}$ and the method repeats the procedure recursively for degree $i+1$ until either a maximum degree or a recursion limit (both defined by the user) is reached.

However, as mentioned before, due to the structure of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, this method does not work for this CDGA. Specifically, the method fails at concrete steps of the algorithm, but it is possible to solve these problems. In particular, if the given algebra $\mathcal{A}$ has generators of degree zero, then the homogeneous parts of the algebra may be infinite-dimensional. Since this method involves linear algebra computations on the homogeneous parts of $\mathcal{A}$, it does not work for this kind of algebra. The main motivation of this work is to adapt the method to overcome this limitation.

Our contribution. Once the basic concepts have been introduced, we are able to describe the main contribution of this thesis, i.e., the effective computation of the minimal model of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$:

The first step was to detect those points in **MMarco**'s method that fail when attempting to compute the minimal model of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. After carefully analyzing the method, we specifically identified the following problems:

- **(P1)** Effective representation of the $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ elements.
- **(P2)** Inability to compute a basis of $\mathcal{H}^*(\mathcal{A}_{\mathcal{PL}}(\mathcal{K})) \Rightarrow$ **[Steps 1, 3 and 4]**
- **(P3)** Given an exact element $\varphi(z) \in \mathcal{A}_{\mathcal{PL}}^{i+1}(\mathcal{K})$, inability to find a primitive $\theta \in \mathcal{A}_{\mathcal{PL}}^{i}(\mathcal{K})$ such that $d_{\mathcal{A}}(\theta) = \varphi(z) \Rightarrow$ **[Step 7]**.

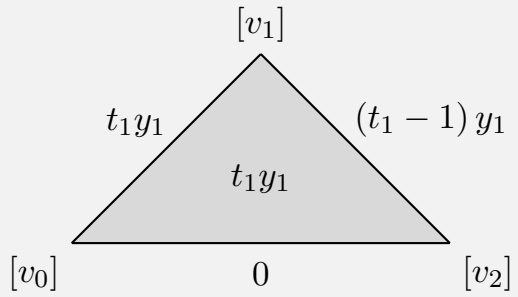
The source of these problems lies in the fact that linear algebra computations involve maps between vector spaces, and $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, as a graded vector space, is not finitely generated. In the next sections, our approach to solving these problems is described.

(P1) Effective representation of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ elements

As mentioned earlier, an element of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ is a collection of simplicial maps $\Phi = \{\Phi_n : \mathcal{K}_n \to (\mathcal{A}_{\mathcal{PL}})_n\}_{n\geq 0}$, where $\mathcal{K}_n$ is the set of simplices of $\mathcal{K}$ of dimension n, and Φ_n must satisfy certain simplicial conditions.

We decided to represent these objects as dictionaries indexed by the non-degenerate simplices of $\mathcal{K}$, whose values are the polynomials assigned to those elements. Although it is not possible to represent all elements of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, with this approach it is easy to represent a finite set of elements. As an illustrative example of this representation, let $\mathcal{K} = \Delta^2$ be the standard 2-simplex; the diagram below shows a homogeneous element $\Phi \in \mathcal{A}_{\mathcal{PL}}^1(\mathcal{K})$:

Representation of a 1-form of $\mathcal{A}_{\mathcal{PL}}(\Delta^2)$



As a dictionary:

$$\begin{aligned} Simplex\,(v_0, v_1) &\longrightarrow t_1 y_1 \\ Simplex\,(v_1, v_2) &\longrightarrow (t_1 - 1)\, y_1 \\ Simplex\,(v_0, v_1, v_2) &\longrightarrow t_1 y_1 \\ Otherwise &\longrightarrow 0 \end{aligned}$$

Although this representation allows us to both represent elements of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ and implement algebra operations (via local computations simplex by simplex), since the method works with finite simplicial sets, we found a way to represent a generating set for this algebra that makes it easier for the user to manipulate algebraic expressions. This representation is based on the discrete Hodge decomposition of the cochain complex $\mathcal{C}(\mathcal{K}) = \{\mathcal{C}^p(\mathcal{K})\}_{p\geq 0}$ for every degree p up to the dimension of $\mathcal{K}$ (see [**32, 38, 61**]). Briefly, this decomposition allows us to decompose the vector spaces $\mathcal{C}^p(\mathcal{K})$ into three orthogonal subspaces: $\mathcal{C}^p(\mathcal{K}) = \mathcal{H}^p \oplus \mathcal{X}^p \oplus \mathcal{Y}^p$ such that

- $\mathcal{H}^p$ is the subspace of $\mathcal{C}^p(\mathcal{K})$ generated by the harmonic cochains and is isomorphic to $\mathcal{H}^p(\mathcal{C}(\mathcal{K}))$ (by abuse of notation, we use the same name for both vector spaces).
- $\mathcal{X}^p$ is the subspace generated by the exact p-cochains.
- $\mathcal{Y}^p$ is the subspace generated by the coexact p-cochains ($\notin \ker(d_{\mathcal{C}})$).
- If $d_{\mathcal{C}}$ is the coboundary operator defined in $\mathcal{C}^*(\mathcal{K})$, then one of the properties of this decomposition is that $d_{\mathcal{C}}(\mathcal{Y}^{p-1}) = \mathcal{X}^p$.

Given the discrete Hodge decomposition, the generating set of the algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ is computed as the image under the Whitney map $\mathcal{W}$ of the basis obtained by the Hodge decomposition of $\mathcal{C}(\mathcal{K})$:

$$\mathcal{A}_{\mathcal{PL}}(\mathcal{K}) = \bigoplus_{0\leq p\leq |\mathcal{K}|} \langle \mathcal{W}(\mathcal{H}^p) \oplus \mathcal{W}(\mathcal{X}^p) \oplus \mathcal{W}(\mathcal{Y}^p) \rangle.$$

(P2) $\mathcal{C}(\mathcal{K})$ as a proxy for $\mathcal{H}(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$

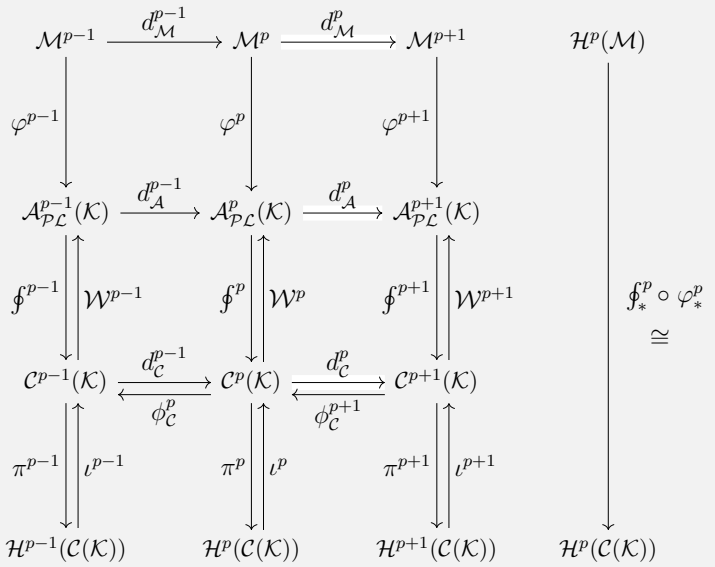
The solution we found to address problem **(P2)** involves the use of the simplicial cochain complex $\mathcal{C}^*(\mathcal{K})$ and the simplicial cohomology $\mathcal{H}^*(\mathcal{C}(\mathcal{K}))$ as a proxy between the cohomology of the minimal model $(\mathcal{M}, \varphi)$ and the cohomology of the algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. This approach is based on the fact that $\pi : \mathcal{C}(\mathcal{K}) \to \mathcal{H}(\mathcal{C}(\mathcal{K}))$ is a quasi-isomorphism (in particular, $\pi_* = id_{\mathcal{H}}$) and also

the induced morphisms:

$$\mathcal{H}(\mathcal{M}) \xrightarrow[\cong]{\varphi_*} \mathcal{H}(\mathcal{A}_{\mathcal{PL}}(\mathcal{K})) \xrightarrow[\cong]{\oint_*} \mathcal{H}(\mathcal{C}(\mathcal{K})).$$

The key idea of this approach is the combination of the discrete Hodge decomposition and the AT-model maps $\pi,\ \iota,\ \phi$, with the quasi-isomorphisms $\oint,\ \mathcal{W}$, in order to move back and forth between $\mathcal{H}(\mathcal{C}(\mathcal{K}))$, $\mathcal{C}(\mathcal{K})$, and $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ without the need to compute $\mathcal{H}(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$.

The use of an AT-Model as a solution to problem (P2)



(P3) Primitives of exact differential forms on simplicial sets

The main problem we encountered in the attempt to implement an effective method to compute the minimal model of the algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ is presented here.
At **step 7** of **MMarco**'s method, when the algorithm finds that, at some degree $i{+}1$, the induced morphism $\varphi^{i+1}_* : \mathcal{H}^{i+1}(\mathcal{M}) \to \mathcal{H}^{i+1}(\mathcal{A})$ is not injective, for each representative $z \in \mathcal{M}^i$ of the generators of $\ker(\varphi^{i+1}_*)$, the method needs to find an element $\theta \in \mathcal{A}^i$ such that $d_A(\theta) = \varphi(z)$. In their implementation, they formulate the problem as a system of linear equations where the coefficient matrix is the matrix representation of the linear map $d^i_{\mathcal{A}} : \mathcal{A}^i \to \mathcal{A}^{i+1}$. In order to compute that matrix, the method needs to compute a basis for both vector spaces and the coordinates of each element $\varphi(z) \in \mathcal{A}^{i+1}$. The problem with this approach for the algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ is precisely the impossibility of building such a matrix.

To overcome this obstacle, we found three methods that solve the problem. We present different solutions for two reasons: first, the initial method we describe was our starting point, but the attempt to improve computational performance and find a smarter solution led us to the third method, which is currently used by default in our implementation; second, we found in the literature works related to other disciplines (apparently distant from Sullivan's theory of minimal models and rational homotopy) where the first method we describe is useful for solving or addressing other problems. In particular, we came across the works of Arnold et al. [**6, 4, 5**], which provide structure to a relatively new field of knowledge called Finite Element Exterior

Calculus (FEEC). However, we have not found any reference in that field mentioning the third method we will describe. Briefly, Dupont found a simplicial contraction that solves the problem in a computationally efficient manner compared to the other approaches. For this reason, we consider it interesting to present the different solutions we found within the same document, as a way to establish bridges between distinct disciplines, in the hope that it may be useful to anyone interested in this problem.

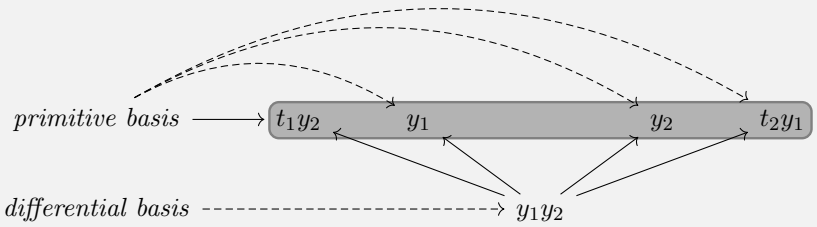
Method 1: the differential $d_{\mathcal{A}} : \mathcal{P}_{r+1}\Lambda^k(\mathcal{K}) \to \mathcal{P}_r\Lambda^{k+1}(\mathcal{K})$

In the field of Finite Element Exterior Calculus (FEEC, see [**6, 4, 5**]), the algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ is used with different notation and motivation. In that context, the algebra of polynomial differential forms $\mathcal{P}\Lambda_n$ is defined, where $\Lambda_n = \langle y_0, \dots, y_n \rangle$ refers to the exterior algebra over the variables $y_0, \dots, y_n$, and $\mathcal{P} = \mathbb{Q}[t_0, \dots, t_n]$ is the polynomial ring over $\mathbb{Q}$ generated by $t_0, \dots, t_n$. In FEEC, the notation $\mathcal{P}_r\Lambda^k$ is used to refer to k-differential forms with coefficients in the polynomial ring $\mathcal{P}$ up to degree r. For an element $\omega \in \mathcal{P}_{r+1}\Lambda^k$, it is known that its image under the differential lies in the space of forms $d(\omega) \in \mathcal{P}_r\Lambda^{k+1}$; that is, the differential decreases the degree of the coefficients by 1 and increases the degree of the form by 1.

To address the problem of finding a primitive under the differential of an exact element of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, the first method we present consists of truncating the differential matrix by the degree of the polynomial coefficients, taking as the primitive basis the monomials present in the image of the Koszul differential, and adding to the linear system the simplicial restrictions necessary to satisfy the patch conditions.

Key idea: Truncate the differential matrix

For each polynomial differential form assigned to each simplex of $\mathcal{K}$, first compute a basis for the differential domain, truncating by coefficients degree.



Then, the differential matrix is built adding to the linear system the simplicial restrictions in order to guarantee that the solution satisfies the patch conditions.

Method 2: The combinatorial model

Although the first method solves our problem, the computational cost of building the differential matrix makes it, in practice, intractable when working with simplicial sets with a large number of simplices. To improve performance and reduce computational complexity, we identified that one of the bottlenecks of the previous method is the computation of the simplicial constraints. This part of the method involves adding an equation to the linear system for every

pair of simplices $\sigma_1, \sigma_2 \in \mathcal{K}$ such that $\sigma_1 \cap \sigma_2 \neq \emptyset$ and for every monomial basis element. To avoid this step, we modified the previous method following the ideas of [**25**].
This second method is based on the same idea as the previous one, in the sense that the differential matrix is truncated according to the degree of the coefficients. However, in this case, the domain of the differential is constructed as the vector space of coexact elements of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ of degree $*-1$ with coefficients up to a certain degree in the variables t_i. By performing algebraic operations with elements of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, we shift from the local approach of the previous method (based on adding constraints simplex by simplex) to a global approach, where we abstract away from the underlying simplicial structure. In this way, we ensure that if the system finds a solution, it will satisfy the patch conditions.

This method significantly improves the computational performance of building the differential matrix, but the combinatorial explosion for large simplicial sets remains inherently present in the method. In practice, this restricts us to working with small-scale simplicial structures. This is the reason why the third and final method was developed.

Method 3: Dupont's contraction

The third and final method we propose for effective computation is, in fact, the only one we have found capable of effectively handling large simplicial sets. The method implements Dupont's contraction defined in [**19, 13, 44**], solving the problem of finding primitives without the need to solve any kind of linear system. The method simply applies Dupont's contraction locally, simplex by simplex, satisfying the patch conditions by definition.

Briefly, Dupont completed the triple $(\oint, \mathcal{W}, s)$ finding a simplicial contraction from $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ to $\mathcal{C}(\mathcal{K})$ defined by the operator $s : \mathcal{A}^*_{\mathcal{PL}}(\mathcal{K}) \to \mathcal{A}^{*-1}_{\mathcal{PL}}(\mathcal{K})$ such that

$$\mathrm{id}_{\mathcal{A}} - \mathcal{W} \circ \oint = d \circ s + s \circ d.$$

In this work, we provide a concrete and effective implementation of Dupont's contraction. To the best of our knowledge, this is the first time such a contraction has been implemented in a computer algebra system for application to arbitrary simplicial sets.

Structure of the document

The rest of the document is structured as follows:

In Chapter I, we present basic concepts and definitions well-known in the literature but necessary for the remainder of the document. Specifically, in Section I.1, we define a minimal model (in the sense of Sullivan) and describe the **MMarco** method for the effective computation of the minimal model of a connected and finitely generated CDGA; in Section I.2, we introduce simplicial objects and explain what algebraic topological models are and how to construct them; and in Section I.3, we present the $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ construction used by Sullivan to define the minimal model of a topological space.

In Chapter II, we go into detail by defining Whitney differential forms. In Section II.1, we give a formal definition and present the most important properties of these elements; in Section II.2, we provide a proof of Dupont's formula to transfer a certain algebraic structure from the algebra of polynomial forms to the simplicial cochain algebra associated with a simplicial set,

concluding with an open question regarding whether a generalization of this formula is known; and we conclude the chapter with Section II.3, describing Dupont's contraction and how it can be implemented effectively.

In Chapter III, our adaptation of the **MMarco** method is described in detail.

Finally, in Chapter IV, we present an illustrative collection of examples, including the computation of the minimal model and the first rational homotopy groups of the complement of smooth projective curves. In particular, we prove that these curves are not Sullivan spaces by computing the minimal model of their universal cover. Furthermore, we also analyze the $\mathcal{K}3$ surface and verify its formality [**16**].

I

Preliminaries

Section **§ I.1**

Commutative Differential Graded Algebras

Definition (I.1.1). A graded $\mathcal{R}$-*algebra* $\mathcal{A}$ is a graded $\mathcal{R}$-module, together with an associative linear map of degree zero

$$\begin{aligned}(\mathcal{A}\otimes\mathcal{A})^k &\longrightarrow \mathcal{A}^k\\ x\otimes y &\mapsto xy\end{aligned}$$

that has an identity $1\in\mathcal{A}^0$. This map is called *product* and satisfies the following property:

$$xy\in\mathcal{A}^{i+j},\ \forall x\in\mathcal{A}^i,\ \forall y\in\mathcal{A}^j.$$

Definition (I.1.2). A *morphism* $\phi:\mathcal{A}\to\mathcal{B}$ of graded $\mathcal{R}$-algebras is a linear map of degree zero such that $\phi(xy)=\phi(x)\phi(y)$ and $\phi(1)=1$. Notice that a morphism of graded $\mathcal{R}$-algebas preserves the degree.

Example (I.1.3). Let $\mathcal{M}$ and $\mathcal{N}$ be the graded $\mathcal{R}$-algebras

- $\mathcal{M}=\mathcal{R}\langle x_1,x_2\rangle$ with $|x_1|=1,|x_2|=2$,
- $\mathcal{N}=\mathcal{R}\langle y_1,y_2,y_3\rangle$ with $|y_1|=1,|y_2|=|y_3|=2$.

We can define the morphism φ of graded $\mathcal{R}$-algebras as follows:

$$\begin{aligned}\varphi:\mathcal{M} &\longrightarrow \mathcal{N}\\ x_1 &\mapsto y_1\\ x_2 &\mapsto y_2-y_3.\end{aligned}$$

Definition (I.1.4). A *graded commutative* algebra (or GCA) is a graded algebra, $\mathcal{A}$, where the product satisfies the property:

$$xy=(-1)^{ij}yx,\ \forall x\in\mathcal{A}^i,\ \forall y\in\mathcal{A}^j.$$

It is important to remark that this algebra is not commutative, but graded commutative. Also, the property does not work for non-homogeneous elements, i.e when one of the factors is not homogeneous.

EXAMPLE (I.1.5). Let $\mathcal{A} = \mathbb{Q}\langle a, b, c\rangle$ with $|a|$ and $|c|$ odds and $|b|$ even. Consider now the non-homogeneous element $a + b$ and the products $(a + b) \cdot c$ and $c \cdot (a + b)$. In that case, notice that the commutative law does not work, since

$$\begin{aligned} (a+b)\cdot c &= a\cdot c + b\cdot c, \\ c\cdot(a+b) &= -a\cdot c + b\cdot c, \end{aligned}$$

and it means that, in general

$$(a + b) \cdot c \neq \pm c \cdot (a + b).$$

DEFINITION (I.1.6). A commutative differential graded algebra (or $CDGA$), $\mathcal{A}$, over a ring $\mathcal{R}$ is a graded $\mathcal{R}$-algebra, $A = \bigoplus_{i=0}^{\infty} \mathcal{A}^i$, together with an $\mathcal{R}$-linear map $d_{\mathcal{A}} : \mathcal{A} \to \mathcal{A}$ that satisfies the following conditions:

- $d_{\mathcal{A}}^2 = 0$,
- $d_{\mathcal{A}}(x) \subseteq \mathcal{A}^{i+1}$, $\forall x \in \mathcal{A}^i$,
- $d_{\mathcal{A}}(xy) = d_{\mathcal{A}}(x)y + (-1)^i x d_{\mathcal{A}}(y)$, $\forall x \in \mathcal{A}^i$, $\forall y \in \mathcal{A}$.

From now on, we will asume that $\mathcal{R} = \mathbb{Q}$.

I.1.1. Minimal Sullivan Algebras

DEFINITION (I.1.7). Let $\mathcal{V} = \bigoplus_{p\geq 1} \mathcal{V}^p$ be a graded $\mathbb{Q}$-vector space. We denote by $\Lambda\mathcal{V} = (\bigotimes \mathcal{V})/\mathcal{I}$, where $\bigotimes \mathcal{V}$ is the tensor algebra of $\mathcal{V}$, and $\mathcal{I}$ is the bilateral ideal generated by $\{v \otimes w - (-1)^{ij} w \otimes v \mid v \in \mathcal{V}^i,\ w \in \mathcal{V}^j\}$, and with the grading induced by the one in $\mathcal{V}$.

Given $\mathcal{W}$ a subspace of $\mathcal{V}$, $\Lambda^n\mathcal{W}$ will denote the image of $\mathcal{W}\otimes \overset{n}{\cdots} \otimes\mathcal{W}$ in $\Lambda\mathcal{V}$, and $\Lambda^{\geq i}\mathcal{V}$ will be $\bigoplus_{n\geq i} \Lambda^n\mathcal{V}$.

NOTATION (I.1.8). If $e_1, \ldots, e_n$ are a basis of the vector space $\mathcal{V}$, then $\bigwedge(e_1, \ldots, e_n)$ will denote the algebra $\Lambda\mathcal{V}$.

DEFINITION (I.1.9). Let $(\Lambda\mathcal{V}, d_{\mathcal{V}})$ be a $CDGA$ defined over a graded $\mathbb{Q}$-vector space $\mathcal{V}$ with differential $d_{\mathcal{V}}$. Such algebra is said to be a *Sullivan* algebra if it satisfies the following property:

- There exists a filtration

$$\mathcal{V} = \bigcup_{k=0}^{\infty} \mathcal{V}(k)$$

of graded subspaces

$$\mathcal{V}(0) \subseteq \mathcal{V}(1) \subseteq \mathcal{V}(2) \subseteq \cdots \subseteq \mathcal{V}$$

where

$$d_{\mathcal{V}}(\mathcal{V}(0)) = 0$$

and

$$d_{\mathcal{V}} : \mathcal{V}(k) \to \Lambda(\mathcal{V}(k-1))$$

for $k \geq 1$.

DEFINITION (I.1.10). A Sullivan algebra is said to be *minimal* if

$$Im(d_{\mathcal{V}}) \subseteq \Lambda^{\geq 2}\mathcal{V}, \quad \forall v \in \mathcal{V}.$$

Definition (I.1.11). A Sullivan *s-minimal model* $(\mathcal{M}, \varphi)$ of a $CDGA$ $(A, d_{\mathcal{A}})$ is a minimal Sullivan algebra $(\mathcal{M}, d_{\mathcal{M}})$ together with an s-quasi-isomorphism:

$$\varphi : \mathcal{M} \longrightarrow \mathcal{A},$$

i.e., a morphism φ such that, the induced morphisms

$$\varphi^i_* : \mathcal{H}^i(\mathcal{M}) \xrightarrow{\simeq} \mathcal{H}^i(\mathcal{A}), \tag{Eq. I.1.1}$$

are isomorphisms for $i \leq s$, and a monomorphism for $i = s + 1$.

An alternative definition of an s-minimal model is given in [**47**]:

Definition (I.1.12). A differential algebra $\mathcal{A}$ is said to be *s-minimal* (in the sense of Sullivan) if it is freely generated by a collection of elements $\{a_i \in \mathcal{A}\}$ with $i \in I$, for a well-ordered index set I, such that $|a_i| \leq |a_j| \leq s$ if $i < j$ and the differential of a generator a_j is expressed in terms of the preceding a_i with $i < j$.

Definition (I.1.13). A Sullivan *minimal model* $(\mathcal{M}, \varphi)$ of a $CDGA$ $(\mathcal{A}, d_{\mathcal{A}})$ is a Sullivan s-minimal model $(\mathcal{M}, d_{\mathcal{M}})$ for all s.

I.1.2. Effective Method for the Computation of the Minimal Model of a finitely generated $CDGA$

Some implementations of effective methods for the computation of the minimal model can be found in the literature for the case of finitely generated $CDGA$s. In [**30**] an algorithm for the computation of the entire minimal model of a given Sullivan algebra (not necessary minimal) through chain contractions is presented. On the other hand, in [**47**], Manero and Marco presented an effective method (and gave a proof of its correctness) for the computation of the minimal model (up to degree s) of a (general) finitely generated connected $CDGA$. This algorithm takes as input a finite presentation of a connected $CDGA$ $\mathcal{A}$ (i.e. all generators are of positive degree), and it outputs a presentation of its minimal model $\mathcal{M}$ up to a given degree, together with the morphism $\varphi : \mathcal{M} \to \mathcal{A}$. In this work we are going to follow the approach of the latter reference. Simultaneously to the development of this thesis, but independently, Šimková carried out theoretical advances that support this idea [**54, 55**].

MMarco's effective approach is the implementation of the standard method found in the literature to obtain Sullivan minimal models from a general $CDGAs$. It is constructive, i.e., it builds the model from lower to upper degrees, adding generators sequentially, in such a way the model satisfies the needed conditions. For completeness, we describe below their method, but we refer readers to [**47**] for a fully detailed description.

MMarco's implementation. Let $(\mathcal{A}, d_{\mathcal{A}})$ be the $CDGA$ for which we want to get its minimal model up to certain degree s. The goal is to construct an algebra $\mathcal{M}$ that is an s-minimal model of $\mathcal{A}$, together with a morphism of differential algebras $\varphi : \mathcal{M} \to \mathcal{A}$ being an s-quasi isomorphism. In order to describe it, we need a set of generators as in Definition (I.1.12), and for each of them, its degree, differential and image by φ. The method consists on adding these generators sequentially, in such a way that we get the needed conditions. Let $(\mathcal{M}, d_{\mathcal{M}})$ be initially the trivial $CDGA$ (with no generators). Before the description of the method, the authors suggested the use of this notation for the better understanding of the algorithm:

- x^k_j will denote a generator of $\mathcal{M}$ of degree k, whose differential is zero.
- y^k_j will denote a generator of $\mathcal{M}$ of degree k, with nonzero differential.

Each time we add a new generator to $\mathcal{M}$, we will use the following diagram

$$b_j^k \quad \overset{\varphi}{\leftarrow} \quad y_j^k \quad \overset{d_{\mathcal{M}}}{\longrightarrow} \quad z_j^{k+1}$$

to denote that we have added the generator y_j^k, with $d_{\mathcal{M}}(y_j^k) = z_j^{k+1} \in \mathcal{M}^{k+1}$ and $\varphi(y_j^k) = b_j^k \in \mathcal{A}^k$. Defined the notation, the method can be summarized as follows:

(1) First step

We start by computing the smallest k_0 for which $\mathcal{H}^{k_0}(\mathcal{A})$ is not trivial. Take a basis $[a_0^{k_0}], \dots, [a_{l_{k_0}}^{k_0}]$ of $\mathcal{H}^{k_0}(\mathcal{A})$, and choose representatives $a_0^{k_0} \dots, a_{l_{k_0}}^{k_0}$ in $\mathcal{A}$. The first generators to add to $\mathcal{M}$ are $x_0^{k_0}, \dots x_{l_{k_0}}^{k_0}$. That is, we start with the diagram

$$\begin{array}{ccccc} a_1^{k_0} & \leftarrow & x_0^{k_0} & \rightarrow & 0 \\ & & \vdots & & \\ a_{l_{k_0}}^{k_0} & \leftarrow & x_{l_{k_0}}^{k_0} & \rightarrow & 0 \end{array}$$

At this point, we have that $\varphi : \mathcal{M} \to \mathcal{A}$ induces an isomorphism $\mathcal{H}^j(\mathcal{M}) \to \mathcal{H}^j(\mathcal{A})$ for all $j \leq k_0$.

(2) Increase degree

Assume that we have already found generators of degree up to $k-1$, such that the map $\varphi : \mathcal{M} \to \mathcal{A}$ induces isomorphisms $\varphi_*^m : \mathcal{H}^m(\mathcal{M}) \to \mathcal{H}^m(\mathcal{A})$ for all $m \leq k-1$. In this step, we will add new generators to get also an isomorphism $\varphi_*^k : \mathcal{H}^k(\mathcal{M}) \to \mathcal{H}^k(\mathcal{A})$, without changing the lower degree cohomologies.

This step has two phases. In the first phase, that might need to be run iteratively, we will add generators of the form y_j^{k-1} until the obtained map in φ_*^k is injective.

Once we have an injective map, we will add generators of type x_j^k (that is, with zero differential), which will respect the lower degree differentials, and the injectivity at degree k, until we get a surjective map.

Step (2.1): Adding generators of the form y_j^{k-1}. Compute the k'th cohomology group $\mathcal{H}^k(\mathcal{M})$ with the already defined generators, and the induced map $\varphi_*^k : \mathcal{H}^k(\mathcal{M}) \to \mathcal{H}^k(\mathcal{A})$. If this map φ_*^k is not injective, take $[z_0^k], \dots, [z_{l_{k-1}^1}^k]$ a basis of $\ker(\varphi_*^k)$, and consider representatives $z_0^k, \dots z_{l_{k-1}^1}^k \in \mathcal{A}$. Then, compute $c_j^k = \varphi(z_j^k) \in \mathcal{A}^k$. Since $[z_j^k]$ is in the kernel of the cohomology map, c_j^k must correspond to a trivial cohomology class, that is, there must be $b_j^{k-1} \in \mathcal{A}^{k-1}$ such that $d_{\mathcal{A}}(b_j^{k-1}) = c_j^k$. So we add the following generators:

$$\begin{array}{ccccc} b_0^{k-1} & \leftarrow & y_0^{k-1} & \rightarrow & z_0^k \\ & & \vdots & & \\ b_{l_{k-1}^1}^{k-1} & \leftarrow & y_{l_{k-1}^1}^{k-1} & \rightarrow & z_{l_{k-1}^1}^k \end{array}$$

Note that, after adding these generators, when the algebra in construction has generators of degree 1, new elements of $\ker(\varphi_*^k)$ could have been added. We can iterate this process if required

adding new generators:

$$\begin{array}{ccccc} b_0^{k-1} & \leftarrow & y_0^{k-1} & \rightarrow & z_0^k \\ & & \vdots & & \\ b_{l_{k-1}^1}^{k-1} & \leftarrow & y_{l_{k-1}^1}^{k-1} & \rightarrow & z_{l_{k-1}^1}^k \\ b_{l_{k-1}^1+1}^{k-1} & \leftarrow & y_{l_{k-1}^1+1}^{k-1} & \rightarrow & z_{l_{k-1}^1+1}^k \\ & & \vdots & & \\ b_{l_{k-1}^1+l_{k-1}^2}^{k-1} & \leftarrow & y_{l_{k-1}^1+l_{k-1}^2}^{k-1} & \rightarrow & z_{l_{k-1}^1+l_{k-1}^2}^k \\ & & \vdots & & \end{array}$$

until the map φ_*^k is injective.

Step (2.2): Adding generators of the form x_i^k. If this map φ_*^k is not surjective, get a basis $[a_1^k], \ldots, [a_j^k], [a_{j+1}^k], \ldots, [a_{l_k}^k]$, of $\mathcal{H}^k(\mathcal{A})$ where $[a_{j+1}^k], \ldots, [a_{l_k}^k]$ is a basis of $Im(\varphi_*^k)$. As before, take representatives $a_1^k, \ldots, a_j^k$, and construct the new generators

$$\begin{array}{ccccc} a_1^k & \leftarrow & x_1^k & \rightarrow & 0 \\ & & \vdots & & \\ a_j^k & \leftarrow & x_j^k & \rightarrow & 0 \end{array}$$

We repeat these two steps until $k = s$. Then we repeat step (2.1) one last time to get injectivity in φ_*^{s+1}. □

Example (I.1.14). Consider $\mathcal{A}$ the algebra generated by six generators of degree 1, $e_1, \ldots, e_6$ and one generator of degree 2, e_7 with no extra relations, and the differential is given by:

$$\begin{array}{l} d_{\mathcal{A}}(e_1) = -e_1 \wedge e_6, \quad d_{\mathcal{A}}(e_2) = -e_2 \wedge e_6, \quad d_{\mathcal{A}}(e_3) = -e_3 \wedge e_6, \\ d_{\mathcal{A}}(e_4) = -e_5 \wedge e_6, \qquad \text{and} \qquad d_{\mathcal{A}}(e_5) = d_{\mathcal{A}}(e_6) = d_{\mathcal{A}}(e_7) = 0. \end{array}$$

We can compute the first cohomology groups:

- $\mathcal{H}^1(\mathcal{A}) = \mathbb{Q}\langle e_5, e_6\rangle$
- $\mathcal{H}^2(\mathcal{A}) = \mathbb{Q}\langle e_4 \wedge e_5, e_4 \wedge e_6, e_7\rangle$
- $\mathcal{H}^3(\mathcal{A}) = \mathbb{Q}\langle e_4 \wedge e_5 \wedge e_6, e_5 \wedge e_7, e_6 \wedge e_7\rangle$

So we can start the method at degree 1:

$$\begin{array}{ccccc} e_6 & \leftarrow & x_0^1 & \rightarrow & 0 \\ e_5 & \leftarrow & x_1^1 & \rightarrow & 0 \end{array}$$

At this point, $\mathcal{H}^2(\mathcal{M})$ would be generated by the class of $x_0^1 \wedge x_1^1$. But the image by the induced map is $-e_5 \wedge e_6$, which is trivial in cohomology, because it is the differential of e_4. So we have to add a new generator in order to make this cohomology class trivial:

$$e_4 \quad \leftarrow \quad y_0^1 \quad \rightarrow \quad x_0^1 \wedge x_1^1$$

Now we get that $d_{\mathcal{M}}(x_0^1 \wedge y_0^1) = d_{\mathcal{M}}(x_1^1 \wedge y_1^1) = 0$, so a basis of $\mathcal{H}^2(\mathcal{M})$ is formed by the classes of $(x_0^1 \wedge y_0^1, x_1^1 \wedge y_0^1)$. Their images by φ are precisely $e_4 \wedge e_6$ and $e_4 \wedge e_5$, which are two elements of the basis of $\mathcal{H}^2(\mathcal{A})$. That is, we have already an injective map at the second cohomology level. Proceeding as described in step (2.2); that is, add a new generator that will be mapped to the remaining element of the basis of $\mathcal{H}^2(\mathcal{A})$:

$$e_7 \quad \leftarrow \quad x_0^2 \quad \rightarrow \quad 0$$

Concerning degree 3, a basis for the cohomology of the already computed $\mathcal{M}$ is given by the classes of $x_0^1 \wedge x_1^1 \wedge y_0^1$, $x_0^1 \wedge x_0^2$ and $x_1^1 \wedge x_0^2$. Since these elements map precisely to the basis of $\mathcal{H}^3(\mathcal{A})$, we already have an isomorphism in degree 3. Up to degree 3 the minimal model of $\mathcal{A}$ is

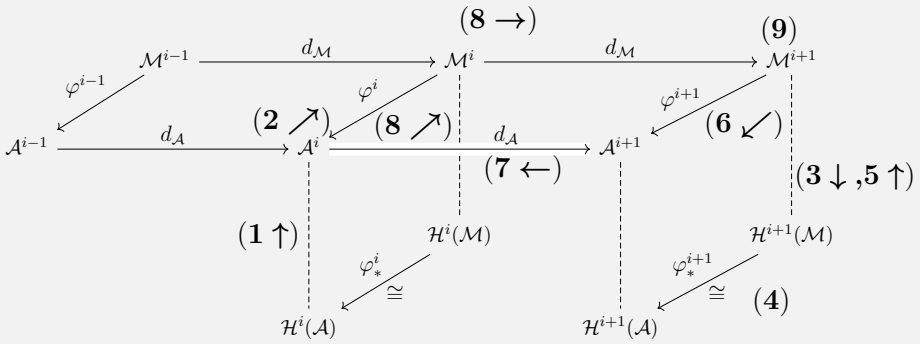
$$\mathcal{M}_3 = \bigwedge(x_0^1, x_1^1, y_0^1) \otimes \bigwedge(x_0^2) \text{ and } d_{\mathcal{M}}(y_0^1) = x_0^1 \wedge x_1^1.$$

In fact, in this case we can check that the obtained result is a minimal model for $\mathcal{A}$ in any degree.
We can summarize the whole process with the complete diagram:

$$\begin{array}{ccccc} e_6 & \leftarrow & x_0^1 & \rightarrow & 0 \\ e_5 & \leftarrow & x_1^1 & \rightarrow & 0 \\ e_4 & \leftarrow & y_0^1 & \rightarrow & x_0^1 \wedge x_1^1 \\ e_7 & \leftarrow & x_0^2 & \rightarrow & 0 \end{array} \qquad \text{(Eq. I.1.2)}$$

Graphical summary. The key aspects of the method applied at some degree i can be summarized in the following diagram:

Key steps of MMarco's implementation



(1) Lifts generators $\mathcal{H}^i(\mathcal{A}) \to \mathcal{A}^i$.
(2) Updates $(\mathcal{M}^i, \varphi^i)$ in order to make φ_*^i surjective.
(3) Gets a basis of $\mathcal{H}^{i+1}(\mathcal{M})$.
(4) Computes the kernel of the induced morphism φ_*^{i+1}.
(5) Lifts the generators of $\ker(\varphi_*^{i+1})$.
(6) For each generator $z \in \ker(\varphi_*^{i+1})$, gets its image $\varphi(z) \in \mathcal{A}^{i+1}$.
(7) For each $\varphi(z) \in \mathcal{A}^{i+1}$, finds $\theta \in \mathcal{A}^i$ such that $d_{\mathcal{A}}(\theta) = \varphi(z)$.
(8) Updates $(\mathcal{M}^i, \varphi^i)$ in order to make φ_*^i injective.
(9) $(\mathcal{M}, \varphi)$ is now an i-minimal model of $\mathcal{A}$.

Arrow notation: having the needed elements → compute/modify/link to.

Limitations of these methods. If the given $CDGA$ $\mathcal{A}$ has generators of degree zero, then the homogeneous parts of the algebra could be infinite dimensional. Since this method involves linear algebra computations on the homogenous parts of $\mathcal{A}$, the method does not work for this

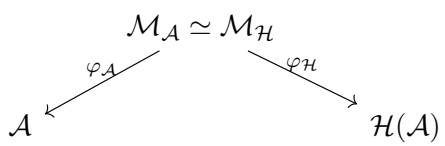
kind of algebras. The main motivation of this work is to adapt the method to overcome this limitation.

I.1.3. Formality

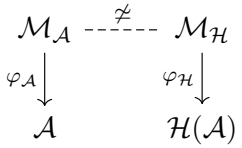
We conclude this section defining the concept of formality. Later, in section I.3.6, we will see that formality has an important implication in homotopy theory.

Definition (I.1.15) ((see [**26**])). A commutative differential graded algebra $\mathcal{A}$ is called *formal* if its minimal model and the model of its cohomology algebra (viewed as a $CDGA$ with trivial differential) are isomorphic (see Figure I.1a).

On the other hand, if the two models are not isomorphic, it is said that the algebra is not formal (see Figure I.1b).



(A) A formal algebra $\mathcal{A}$.



(B) A non-formal algebra $\mathcal{A}$.

Figure I.1. A $CDGA$ $\mathcal{A}$ is called formal if its minimal model is isomorphic to the minimal model of its cohomology algebra.

In [**47**], Manero and Marco proposed two criteria for i-formality, one necessary and one sufficient. These criteria can be checked effectively, and they conjectured that the fulfilment of both criteria is a necessary and sufficient condition to prove i-formality.

Section § I.2
Simplicial sets

In this section we are going to briefly introduce the concepts of simplicial objects and simplicial morphisms, a fundamental structure in the construction of the Sullivan's functor. In particular, this section is centered in the definition of *simplicial sets* and *simplicial set morphisms*, the combinatorial tool we will use to model both topological spaces and differential forms. We will begin defining the geometrical n-simplex and we will abstract the combinatorial structure in order to finally define the simplicial set structure. For a detailed introduction to simplicial sets we refer to [**27**], and for a deeper introduction to basic computational topology we suggest the following reading [**20**].

I.2.1. The geometrical n-simplex Δ^n.

We begin with the definition of the standard geometrical n-simplex:

Definition (I.2.1) (Geometrical n-simplex). The standard *geometrical n-simplex* Δ^n is the topological space defined as the set:

$$\Delta^n = \left\{ (\lambda_0, \lambda_1, \dots, \lambda_n) \in \mathbb{R}^{n+1} \;\middle|\; \lambda_i \geq 0 \text{ for all } i, \text{ and } \sum_{i=0}^{n} \lambda_i = 1 \right\}$$

equipped with the subspace topology inherited from $\mathbb{R}^{n+1}$.

The tuple $(\lambda_0, \lambda_1, \ldots, \lambda_n)$ are called the *barycentric coordinates* of Δ^n. A representation of the standard 2-simplex can be seen in Figure I.2.

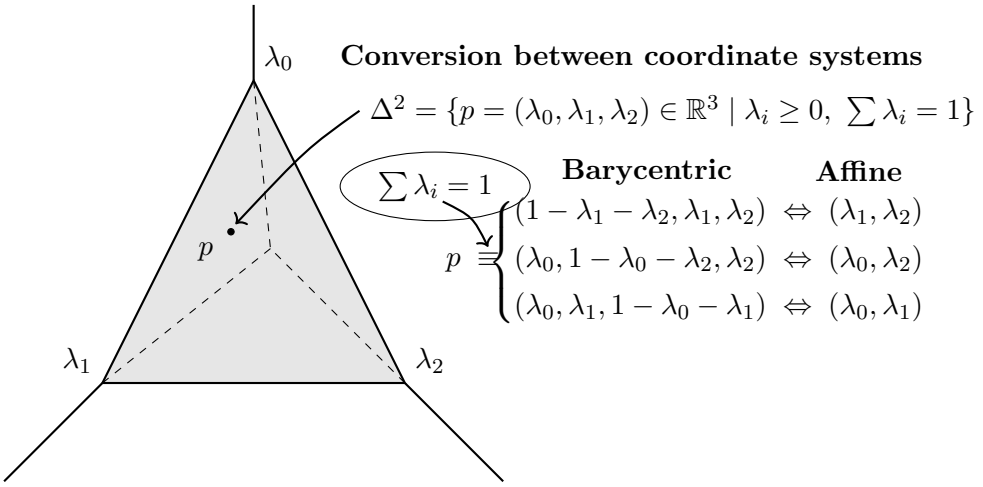


FIGURE I.2. The standard simplex Δ^2. This space is homeomorphic to the disk $\mathbb{D}^2$.

Due to the relation $\sum_{i=0}^{n} \lambda_i = 1$, for each λ_i, it is possible to represent Δ^n in $\mathbb{R}^n$ with affine coordinates, just projecting λ_i to the origin. A visual representation of this change of coordinates is shown in Figure I.3.

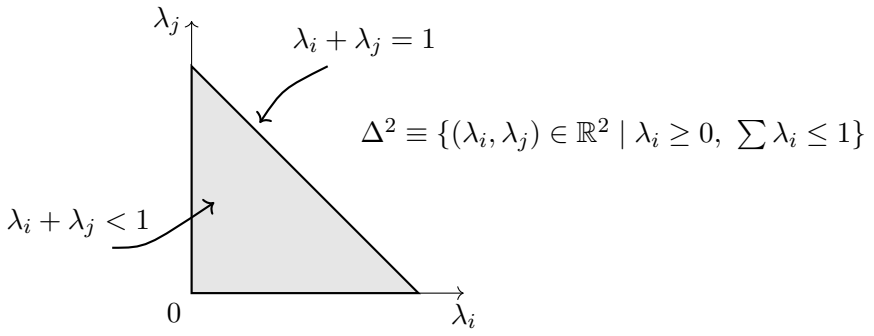


FIGURE I.3. Projection of the standard Δ^2-simplex onto the λ_i-λ_j axis.

A *face* of the geometrical n-simplex Δ^n is the convex set spanned by some subsets of $\{\lambda_0, \ldots, \lambda_n\}$. The *ith*-face of Δ^n is the $(n-1)$-simplex spanned by the points $\{\lambda_0, \ldots, \hat{\lambda}_i, \ldots, \lambda_n\}$.

A *geometrical simplicial complex* $\mathcal{K}^\Delta$ is defined as a collection of sets $\mathcal{K}^\Delta = \{\mathcal{K}_n^\Delta\}$, where the elements of the set $\mathcal{K}_0^\Delta$ contains an arbitrary number of geometrically independent points (called *vertices*), and an element $\sigma_n \in \mathcal{K}_n^\Delta$ is defined as the convex set spanned by a $(n+1)$-subset of vertices.

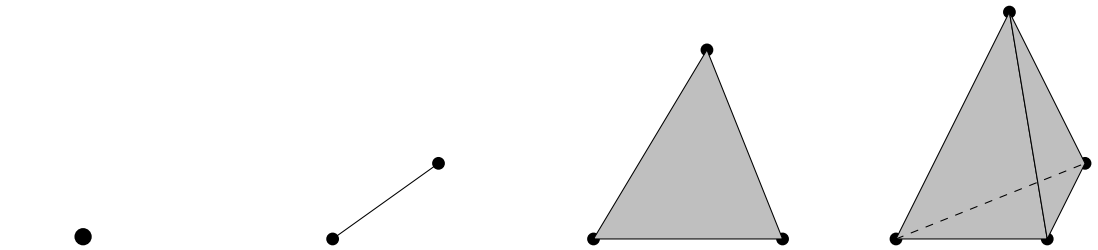
I.2.2. Simplicial Complexes

Taking the purely combinatorial structure of a geometrical simplicial complex, we can make the following generalization. An *abstract simplicial complex* $\mathcal{K}$ is a collection of sets

$\mathcal{K} = \{\mathcal{K}_n\}_{n\geq 0}$, where the elements of $\mathcal{K}_n$ are called *n-simplices*, and the collection must satisfy the following properties:

- every *n-simplex* $\sigma \in \mathcal{K}_n$ has exactly $n+1$ distinct faces.
- every face of a simplex $\sigma_n \in \mathcal{K}_n$ is in $\mathcal{K}_{n-1}$, and
- the intersection of any two simplices of $\mathcal{K}$, is either a face of each them or the empty set.

From now on we will refer to this structures just as simplicial complexes. A visual and common representation of n-simplices is shown in Figure I.4.



(A) 0-simplex (B) 1-simplex (C) 2-simplex (D) 3-simplex

FIGURE I.4. Building blocks for simplicial complexes.

Oriented simplices and boundary map

In a simplicial complex $\mathcal{K}$, the set $\mathcal{K}_0$ is a totally ordered set $v_0, \ldots, v_N$, and an *abstract* n-simplex can be represented as a sequence of vertices $(v_{i_0}, \ldots, v_{i_n})$. Then, for a simplex $\sigma_n \in \mathcal{K}_n$, the *ith*-face ∂_i^n is the $(n-1)$-simplex $\partial_i^n \sigma_n = (v_{i_0}, \ldots, \hat{v}_{i_j}, \ldots, v_{i_n}\}$. The ordering of vertices defines an orientation for every $(v_{i_0}, \ldots, v_{i_n}) \in \mathcal{K}$ determined by the order of the indices $i_0, \ldots, i_n$.

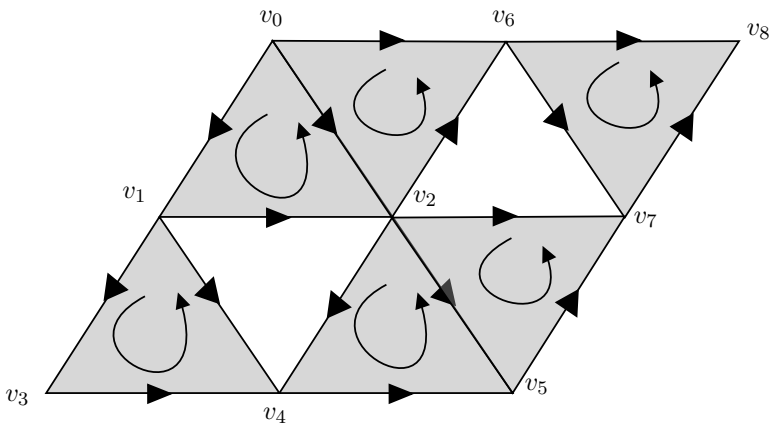


FIGURE I.5. Oriented simplicial complex

This orientation allows us to define a boundary operator acting on every simplex:

DEFINITION (I.2.2). The *boundary map* d_n is the linear operator defined as

$$d_n(\sigma) = \sum_{j=0}^{n} (-1)^j \partial_j^n(\sigma) \qquad \text{(Eq. I.2.3)}$$

for a given $\sigma \in \mathcal{K}_n$.

An important property of the boundary map is that it squares to zero, i.e., $d \circ d = 0$. In Figure I.5 it is represented the orientation of a simplicial complex of dimension 2.

I.2.3. Abstract simplicial structures

For the work presented here, we asumme that $\mathfrak{C}$ is a *concrete category* (see [**8**]): these are categories in which the objects are sets, possibly equipped with some structure, and the arrows are certain, possibly structure-preserving, functions. Examples of such categories are: $\mathfrak{Set}, \mathfrak{Rng}, \mathfrak{Top} \ldots$

Definition (I.2.3). A simplicial object $\mathcal{K}$ with values in a category $\mathfrak{C}$ is a sequence $\{\mathcal{K}_n\}_{n\geq 0}$ of objects in $\mathfrak{C}$, together with $\mathfrak{C}$-morphisms:

$$\begin{aligned} \partial_i^n &: \mathcal{K}_n \to \mathcal{K}_{n-1},\ 0 \leq i \leq n, \\ s_j^n &: \mathcal{K}_n \to \mathcal{K}_{n+1},\ 0 \leq j \leq n, \end{aligned} \tag{Eq. I.2.4}$$

where the morphism ∂_i (respectively, s_j) is called the i-face (respectively, j-degeneration), that satisfy the identities:

$$\begin{aligned} \partial_i^{(n-1)} \circ \partial_j^n &= \partial_{j-1}^{(n-1)} \circ \partial_i^n, && i < j, \\ s_i^{(n-1)} \circ s_j^n &= s_{j+1}^{(n-1)} \circ s_i^n, && i \leq j, \\ \partial_i^{(n-1)} \circ s_j^n &= \begin{cases} s_{j-1}^{(n-1)} \circ \partial_i^n, & i < j, \\ id, & i = j, j+1, \\ s_j^{(n-1)} \circ \partial_{i-1}^n, & i > j+1. \end{cases} \end{aligned} \tag{Eq. I.2.5}$$

If $\mathcal{K} = \{\mathcal{K}_n\}_{n\geq 0}$ is a simplicial object, then we say that the object of dimension n is $\mathcal{K}_n$.

Definition (I.2.4). Given two simplicial objects $\mathcal{K}, \mathcal{L}$ in a category $\mathfrak{C}$, a *simplicial morphism* $\Phi = \{\Phi_n : \mathcal{K}_n \to \mathcal{L}_n\}_{n\geq 0}$ is a sequence of $\mathfrak{C}$-morphisms such that each Φ_n commutes with face and degeneration morphisms.

I.2.4. Simplicial Coordinate System

Let $\mathcal{K} = \{\mathcal{K}_n\}_{n\geq 0}$, be a simplicial object, and let $\partial_i^n : \mathcal{K}_n \to \mathcal{K}_{n-1}$ be the i-th face morphism at dimension n. We define $\partial^n := \{\partial_i^n\}_{i\in\{0,\ldots,n\}}$ as the set of face morphisms from dimension n to dimension $n-1$. Now, let $p, q \in \mathbb{Z}$ so that $p, q \geq 0$ and $p + q = n$. Also, let $J := (j_1^{p+1}, \ldots, j_q^n)$ be a sequence of q non-negative integers satisfying $j^k \in \{0, \ldots, k\},\ \forall k \in \{p+1, \ldots, n\}$. For such sequence, consider the morphism:

$$\overline{\partial}_J : \mathcal{K}_n \to \mathcal{K}_p,$$

defined as the composition of q face morphisms determined by the sequence J in such a way that:

$$\overline{\partial}_J := \partial_{j_1}^{p+1} \circ \cdots \circ \partial_{j_q}^n. \tag{Eq. I.2.6}$$

Looking at the simplicial restrictions from definition (I.2.3):

$$\partial_i^{n-1} \circ \partial_j^n = \partial_{j-1}^{n-1} \circ \partial_i^n, \quad i < j,$$

it is easy to proof that, applying this rule, every morphism of type $\overline{\partial}_J$ can be uniquely identified with a strictly increasing sequence of indices of length q, $I = (i_1^{p+1} < \cdots < i_q^n)$, in such a way that:

$$\overline{\partial}_I = \overline{\partial}_J.$$

Now, Consider the set of multi-indices:

$$I_p^n := \{(i_1^{p+1}, \ldots, i_q^n) |\ i_1^{p+1} < \cdots < i_q^n \text{ and } 0 \leq i_l^k \leq k\}, \tag{Eq. I.2.7}$$

Then, for each $\mathfrak{C}$-object $\mathcal{K}_n$, we define the set of all unique face morphisms compositions:

$$\overline{\partial}_p^n := \{\overline{\partial}_I\}_{I \in I_p^n}, \tag{Eq. I.2.8}$$

The elements of this set are the $\mathfrak{C}$-morphisms:

$$\overline{\partial}_I : \mathcal{K}_n \to \mathcal{K}_p, \ \ I \in I_p^n, \tag{Eq. I.2.9}$$

such that $\overline{\partial}_I \neq \overline{\partial}_J$, if $I \neq J$, $\forall I, J \in I_p^n$.

Definition (I.2.5). An element $\gamma = \overline{\partial}_I \in \overline{\partial}_p^n$ is called a *face path* from dimension n to dimension p.

Definition (I.2.6). Given two face paths $\gamma_I \in \overline{\partial}_p^n$ and $\gamma_J \in \overline{\partial}_q^n$ so that $p+q = n$ and $|I \cap J| = k$ (considering I, J as sets of indices), we will say that the face paths I, J are $(k-1)$-*connected*. For the case $k = 0$, we will say that the paths are *non-connected*. If $I \cap J \neq \emptyset$ we will say that the paths γ_I and γ_J *cross* each other, *intersect* or are *crossing paths*. Otherwise, for non-connected paths, we will also say that the paths *do not cross* each other, *do not intersect* or *are not crossing paths*.

Definition (I.2.7). The set $\overline{\partial}_p^n$ is called *the set of face paths* from dimension n to dimension p.

Motivated by Eq. I.2.6, for an n-simplex $\sigma_n \in \mathcal{K}_n$ and a face path $\gamma \in \overline{\partial}_p^n$, when it is useful, we will say that the element $\gamma(\sigma_n) = \sigma_p \in \mathcal{K}_p$ is a q-face of σ_n, where the number q will emphasize the shift of dimension between σ_n and σ_p. Also, we will say that γ is the face path from σ_n to σ_p. Furthermore, the set:

$$\gamma^{-1}(\sigma_p) \subset \mathcal{K}_n, \tag{Eq. I.2.10}$$

is the set containing all n-simplices connected to σ_p by γ. As a face path is determined by a unique sequence of indices I, by abuse of notation, when there is no ambiguity, we will say that I is a face path when actually the face path is $\overline{\partial}_I$.

In many cases, given a simplicial object $\mathcal{K}$, it is useful to consider the subsimplicial object $\mathcal{NK} := \{\mathcal{NK}_n\}_{n \geq 0}$, where

$$\mathcal{NK}_n := \{\sigma \in \mathcal{K}_n |\ \sigma \text{ is a non-degenerated simplex}\}.$$

In that scenario, for a given n-simplex $\sigma_n \in \mathcal{NK}_n$ and a face path $\gamma \in \overline{\partial}_p^n$, we will say that

$$\gamma(\sigma_n) = \begin{cases} \sigma_p, & \text{if } \gamma \text{ is a face path from } \sigma_n \text{ to } \sigma_p \in \mathcal{NK}_p, \\ \emptyset, & \text{if } \sigma_n \notin \gamma^{-1}(\sigma_p), \ \ \forall \sigma_p \in \mathcal{NK}_p. \end{cases} \tag{Eq. I.2.11}$$

Definition (I.2.8). Given an n-simplex $\sigma \in \mathcal{K}_n$ and a face path $\gamma \in \overline{\partial}_p^n$, the set $\gamma^{-1} \circ \gamma(\sigma) \subset \mathcal{K}_n$ is the set containing all n-simplices connected to the same q-face $\gamma(\sigma) \in \mathcal{K}_p$ of σ by γ (recall that $q = n - p$).

I.2.5. Simplicial sets

DEFINITION (I.2.9). A *Simplicial set* $\mathcal{K}$ is a simplicial object in the category of *Sets* (for a detailed introduction, see [**27**]). Thus, it can be seen as a collection of sets $\mathcal{K} = \{\mathcal{K}_n\}_{n\geq 0}$ and maps between them, satisfying the simplicial restrictions (see Definition (I.2.3)). The elements of $\mathcal{K}_n$ are called n-simplices.

DEFINITION (I.2.10). If a simplicial set $\mathcal{K}$ has a finite number of non-degenerated simplices, then it is said that $\mathcal{K}$ is a finite simplicial set.

The simplicial set construction allows us to represent topological spaces in a more efficient way. As an example, consider the simplicial set $\mathcal{K} = \mathbb{S}^2$ of Figure I.6. This object can be described as the collection of sets:

$$\begin{aligned}\mathcal{K}_0 =&\{v_0\}\\ \mathcal{K}_1 =&\{s_0(v_0)\}\\ \mathcal{K}_2 =&\{\mathbb{D}^2, s_1(s_0(v_0))\}\end{aligned}$$

and, for $n > 2$, $\mathcal{K}_n$ contains only the corresponding degenerated simplices.

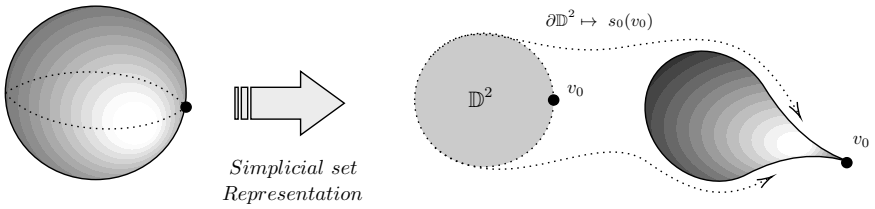


FIGURE I.6. Representation of the 2-Sphere as a simplicial set. This representation only contains 2 non-degenerated simplicies: a 2 simplex and a vertex. The boundary of the 2-simplex is collapsed to the degeneration of the vertex.

The face maps $\partial_i^n : \mathcal{K}_n \to \mathcal{K}_{n-1}$, represented as matrices are determined by the set maps $\partial_0^1 = \partial_1^1 = (1)$, and $\partial_0^2 = \partial_1^2 = \partial_2^2 = (1, 1)$.

Barycentric coordinates on simplicial sets. Although barycentric coordinates are defined for geometrical simplicial complexes, the abstraction carried out in the definition of simplicial sets breaks this correspondence between simplices and geometrical representation. However, it is possible to define a related meaning of barycentric coordinates in terms of face paths induced by the simplicial structure. This coordinates are defined locally as follows: for a p-simplex $\sigma_p \in \mathcal{K}_p$, its *barycentric coordinates at dimension p* are just the same as the standard Δ^p simplex:

$$\sigma_p = \left\{ (\lambda_0, \ldots, \lambda_p) \in \mathbb{R}^{p+1} \;\middle|\; \lambda_i \geq 0 \text{ for all } i, \text{ and } \sum_{i=0}^{p} \lambda_i = 1 \right\}$$

equipped with the subspace topology inherited from $\mathbb{R}^{p+1}$. Suppose now σ_p is a q-face of $\sigma_n \in \mathcal{K}_n$ determined by the face path $I = (i_1, \ldots, i_q)$, with $i_1 < \cdots < i_q$. Denote by $\lambda_0^p, \ldots, \lambda_p^p$ the barycentric coordinates of σ_p at dimension p, and by $\lambda_0^n, \ldots, \lambda_n^n$ the barycentric coordinates of σ_n at dimension n. Considering I as a subset of $[n] = \{0, \ldots, n\}$, let $J := (j_0, \ldots, j_p)$ be the tuple containing the elements in $I^c = [n] \setminus I$ (the complement of I in $[n]$) ordered in a

strictly increasing sequence. Then, we define the *barycentric coordinates of* σ_p *on* σ_n as the ones determined by the inclusion

$$\begin{array}{rccc} i_{J^I}: & \Delta^p & \hookrightarrow & \Delta^n \\ & \lambda_k^p & \to & \lambda_{j_k}^n, \end{array}$$

and $\lambda_i^n := 0,\ \forall i \in I$.

I.2.6. Algebraic Topological models and discrete Hodge decompositions

For the rest of the section, assume the ground ring is $\mathbb{Q}$ (see [**64, 32**]).

Briefly, an algebraic topological model is a useful construction that uses (co)chain contractions on chain complexes in order to obtain homological information in a computationally more efficient way (see [**36, 33, 34, 53**] for theoretical definitions, and [**51**] for concrete implementations over different simplicial structures).

Definition (I.2.11) ([**53**]). Let $(\mathcal{C}, d_{\mathcal{C}})$ be a chain complex. The *Algebraic Topological model* (or *AT*-model) of $(\mathcal{C}, d_{\mathcal{C}})$ is a chain complex $(\mathcal{M}, 0)$ (meaning $d_{\mathcal{M}} = 0$), such that $\mathcal{M} \cong \mathcal{H}(\mathcal{C})$, along with chain maps

$$\pi : \mathcal{C} \to \mathcal{M}$$

and

$$\iota : \mathcal{M} \to \mathcal{C}$$

satisfying $\pi \circ \iota = \mathrm{id}_{\mathcal{M}}$ and $\iota \circ \pi$ is chain homotopic to $\mathrm{id}_{\mathcal{C}}$. Furthermote, the chain homotopy ϕ must satisfy

- $\phi \circ \phi = 0$,
- $\pi \circ \phi = 0$,
- $\phi \circ \iota = 0$.

Definition (I.2.12). Taking the previous definition, such a chain homotopy ϕ is called a *chain contraction*.

An alternative definition for cochain complexes is found in [**13**]:

Definition (I.2.13). Let $(\mathcal{C}, d_{\mathcal{C}})$ and $(\mathcal{M}, d_{\mathcal{M}})$ be two cochain complexes. Then, a *contraction* (π, ι, ϕ) from $\mathcal{C}$ to $\mathcal{M}$ consists of chain maps:

$$\pi : \mathcal{C} \to \mathcal{M},$$
$$\iota : \mathcal{M} \to \mathcal{C},$$

and a chain homotopy

$$\phi : \mathcal{C}^* \to \mathcal{C}^{*-1},$$

such that

$$\pi \circ \iota = \mathrm{id}_{\mathcal{M}}$$

and

$$\iota \circ \pi - \mathrm{id}_{\mathcal{C}} = d_{\mathcal{C}} \circ \phi + \phi \circ d_{\mathcal{C}}.$$

A chain contraction provides a strong relation between the cochain complexes $\mathcal{C}$ and $\mathcal{M}$; for example, their cohomology groups are isomorphic.

On the other hand, let $\mathcal{K}$ be a finite oriented simplicial set. For the rest of the document, assume that $\mathcal{C}_*(\mathcal{K})$ (respectively $\mathcal{C}^*(\mathcal{K})$) denotes the (co)chain complex $\mathcal{C}_*(\mathcal{NK})$ (respectively $\mathcal{C}^*(\mathcal{NK})$) defined over the subsimplicial set $\mathcal{NK} := \{\mathcal{NK}_n\}_{n\geq 0}$, where

$$\mathcal{NK}_n := \{\sigma \in \mathcal{K}_n |\ \sigma \text{ is a non-degenerated simplex}\}.$$

Let $n \geq 0$ be a non-negative integer, and $d^n_{\mathcal{C}} : \mathcal{C}^n(\mathcal{K}) \to \mathcal{C}^{n+1}(\mathcal{K})$, the matrix representation of the coboundary operator on $\mathcal{C}^*(\mathcal{K})$. Then ([**18, 32**]),

Definition (I.2.14). The *nth-combinatorial Laplacian* (or *nth-Laplacian matrix*) is the linear operator $L_n : \mathcal{C}^n(\mathcal{K}) \longrightarrow \mathcal{C}^n(\mathcal{K})$ given by

$$L_n = L_n^{down} + L_n^{up}, \qquad \text{(Eq. I.2.12)}$$

where

$$L_n^{down} = d^{n-1}_{\mathcal{C}} \cdot (d^{n-1}_{\mathcal{C}})^T$$

and

$$L_n^{up} = (d^n_{\mathcal{C}})^T \cdot d^n_{\mathcal{C}}.$$

This operator has interesting properties and, in particular, we will find useful the following ones:

Definition (I.2.15) (Spectrum, [**32**] p. 40). Let $\mathcal{K}$ be a finite oriented simplicial complex, and let n be an interger with $0 \leq n \leq dim(\mathcal{K})$. Then, the *nth-Laplacian spectrum* of $\mathcal{K}$, denoted by $Spec(L_n)$, is the multiset of eigenvalues of L_n.

Theorem (I.2.16) (Orientation independent, [**32**] p. 40). *Let $\mathcal{K}$ be finite oriented simplicial complex, and let n be an interger with $0 \leq n \leq dim(\mathcal{K})$. Then, $Spec(L_n)$ is independent of the choice of orientation for the nth-simplices of that complex.*

Looking at the kernel of the combinatorial Laplacian, a powerful property used to compute efficiently Betti numbers ([**28**]) is the following:

Theorem (I.2.17) (Combinatorial Hodge Theorem, [**64**]). *Let $\mathcal{K}$ be a finite simplicial complex. Then,*

$$\mathcal{H}^n(\mathcal{K}) \cong \ker(L_n),$$

for each integer n with $0 \leq n \leq dim(\mathcal{K})$, where the isomorphism is as vector spaces over $\mathbb{Q}$.

Combining theorems (I.2.15) and (I.2.17), we see that $\mathcal{H}^n(\mathcal{K})$ is isomorphic to the eigenspace of L_n associated to the eigenvalue $\lambda = 0$. Looking at the non-zero eigenvalues of the operator, it is remarkable the following theorem in our context:

Theorem (I.2.18) ([**32**], p. .46). *Let $\mathcal{K}$ be a finite simplicial complex, and let n be an integer with $0 \leq n \leq dim(\mathcal{K})$. Then, $Spec(L_n)$ contains no non-integer rational numbers.*

Finally, the last but most important property (for the purpose of this thesis) of the combinatorial laplacian is presented. Notice that each $\mathcal{C}^n(\mathcal{K})$ has the structure of an Euclidean vector space of dimension $|\mathcal{K}_n|$, and therefore, orthogonality is well defined in terms of the underlying scalar product. Then:

Theorem (I.2.19) (discrete Hodge decomposition [**18**]). *Let $\mathcal{K}$ be a finite simplicial complex and consider the nth-combinatorial laplacian as the sum $L_n = L_n^{down} + L_n^{up}$, for some $0 \leq n \leq dim(\mathcal{K})$. Then, the vector space $\mathcal{C}^n(\mathcal{K})$ can be decomposed in three orthogonal subspaces*

$$C^n(\mathcal{K}) = H_n \oplus X_n \oplus Y_n$$

where

$$\begin{aligned} H_n &:= \ker(L_n^{up}) \cap \ker(L_n^{down}), \\ X_n &:= \ker(L_n^{up}) \cap Im(L_n^{down}), \\ Y_n &:= Im(L_n^{up}) \cap \ker(L_n^{down}). \end{aligned}$$

This theorem defines three orthogonal subspaces of $\mathcal{C}^n(\mathcal{K})$ such that: (1) H_n is the subspace of harmonic *nth*-cochains (cocycles that are not coboundaries) such that $H_n \cong \mathcal{H}^n(\mathcal{C}^*(\mathcal{K}))$; (2) the subspace X_n is generated by the exact *nth*-cochains (coboundaries), and (3) the space Y_n is the subspace of coexact *nth*-cochains (this space is the complement of $\ker(d_{\mathcal{C}}^n)$. Then, if $dim(H_n) > 0$, the basis of H_n is a representative basis of the cohomology vector space $\mathcal{H}^n(\mathcal{C})$. The rest of the section is dedicated to describe how to obtain an AT-model and a discrete Hodge decomposition of the cochain complex $\mathcal{C}^*(\mathcal{K})$. It is important to remark that, although the previous theorems have been stated as in the original works, taking into account that our ground ring is $\mathbb{Q}$, the definitions and theorems are valid also for simplicial sets.

Computation of AT-models and Hodge decompositions

Let $\mathcal{K}$ be a simplicial set. For some dimension n, the computation of an AT-model can be done as follows: first, compute the discrete laplacian

$$L_n = L_n^{down} + L_n^{up},$$

and, applying theorem (I.2.19), obtain the three orthogonal subspaces:

$$C^n(\mathcal{K}) = H_n \oplus X_n \oplus Y_n.$$

Having this decomposition, it is easy to obtain an AT-model of the simplicial cochain complex. Recall the definition of a cochain contraction given in (I.2.13). In our case, the cochain complex $\mathcal{M}$ is precisely the harmonic cochain subcomplex $(H_n, 0)$ endowed with trivial differential. We just need to define the chain maps π, ι and the chain homotopy ϕ. This can be done easily considering the hodge decomposition of the cochain spaces

$$\cdots \to \mathcal{C}^{n-1}(\mathcal{K}) \xrightarrow{d_{\mathcal{C}}^{n-1}} \mathcal{C}^n(\mathcal{K}) \xrightarrow{d_{\mathcal{C}}^{n}} \mathcal{C}^{n+1}(\mathcal{K}) \xrightarrow{d_{\mathcal{C}}^{n+1}} \cdots .$$

Let dim_h, dim_x, dim_y be the dimensions of the spaces H_n, X_n, Y_n respectively. Choose, as basis of $\mathcal{C}^n(\mathcal{K})$, the row space of the matrix $\boldsymbol{B_n} = (H_n|X_n|Y_n)^T$, built as a 3-block matrix where the block H_n has size $(dim_n \times dim_h)$ and its columns are the basis elements of the subspace (equivalently for the other blocks X_n, Y_n). Then, we define the projection map $\mathcal{C}^n(\mathcal{K}) \to H_n$ as the linear map defined by the 3-block matrix $\pi_n = (1|0|0)$ of size $dim_h \times dim_n$, where the first block is the identity matrix of order dim_h, and the other two blocks are the zero matrices of sizes $dim_h \times dim_x$ and $dim_h \times dim_y$ respectively. The inclusion map $\iota : H_n \to \mathcal{C}^n(\mathcal{K})$ is defined just as the linear map determined by the matrix $\iota_n = \pi_n^T$.

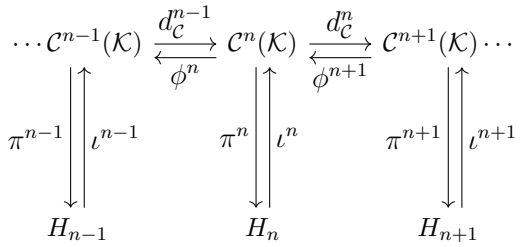


FIGURE I.7. Representation of the chain maps involved in the AT-model of $\mathcal{C}^*(\mathcal{K})$.

Finally, notice that the subspaces Y_n and X_{n+1} are isomorphic by construction, because Y_n is the subspace of coexact n-cochains, and X_{n+1} is the subspace of $(n+1)$-coboundaries. Using this idea, consider the linear isomorphism $E_n : Y_n \to X_{n+1}$ determined by the matrix $E_n = d_{\mathcal{C}}^n \cdot D_n^T \cdot D_n$, where $D_n = (0|0|1)$ is another 3-block matrix where the first two blocks are the zero matrices of dimensions $dim_y \times dim_h$ and $dim_y \times dim_x$ respectively, and the third block is the identity matrix of order dim_y. Then, E_n correspond with the change-of-basis matrix between the vector subspaces Y_n and X_{n+1} on $\mathcal{C}^n(\mathcal{K})$ and $\mathcal{C}^{n+1}(\mathcal{K})$ respectively. Actually, what we are doing is the same as π_n and ι_n, i.e., to use the linear map $D_n : \mathcal{C}^n(\mathcal{K}) \to Y_n$ as a projection, and the transpose $D_n^T : Y_n \to \mathcal{C}^n(\mathcal{K})$ as a inclusion. We conclude the computations defining the chain homotopy ϕ as the linear map determined by the matrix

$$\phi_n = E_n^T = (d_{\mathcal{C}}^n \cdot D_n^T \cdot D_n)^T = D_n^T \cdot D_n \cdot (d_{\mathcal{C}}^n)^T.$$

Taking a look at Figure I.7, it is easy to see that all maps satisfy:

$$\mathrm{id}_{\mathcal{C}} - \iota \circ \pi = d_{\mathcal{C}} \circ \phi + \phi \circ d_{\mathcal{C}}.$$

Making a simple reordering of the terms:

$$\mathrm{id}_{\mathcal{C}} = \iota \circ \pi + d_{\mathcal{C}} \circ \phi + \phi \circ d_{\mathcal{C}},$$

it is (even more) explicit the discrete Hodge decomposition as the direct sum of the subspaces generated by

$$\mathcal{C}^*(\mathcal{K}) = \langle \iota \circ \pi \rangle \oplus \langle d_{\mathcal{C}} \circ \phi \rangle \oplus \langle \phi \circ d_{\mathcal{C}} \rangle.$$

Last but not least, notice that it is possible to revert the computations and obtain a discrete Hodge decomposition from a given AT-model, just defining the subspaces H_n, X_n and Y_n as the spaces:

$$\begin{aligned} H_n &:= \mathrm{Im}(\iota \circ \pi), \\ X_n &:= \mathrm{Im}(d_{\mathcal{C}} \circ \phi), \\ Y_n &:= \mathrm{Im}(\phi \circ d_{\mathcal{C}}). \end{aligned}$$

SECTION § I.3
The functor $\mathcal{K} \rightsquigarrow \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$

In this section are included all the mathematical objects and algorithms previously defined in the literature and needed for the development and understanding of the work presented in the next section. For a more extensive description see [**22, 23, 21, 30, 37**].

I.3.1. The Singular Simplicial Set $\mathcal{S}_*(\mathcal{X})$

DEFINITION (I.3.1) (Singular Simplicial Set). Let $\mathcal{X}$ be a topological space. The *singular simplicial set* of $\mathcal{X}$, is the simplicial set $\mathcal{S}(\mathcal{X}) = \{\mathcal{S}_n(\mathcal{X})\}_{n\geq 0}$ where, for each $n \geq 0$, the set of n-simplices is:

$$\mathcal{S}_n(\mathcal{X}) = \{\sigma : \Delta^n \to X \mid \sigma \text{ is a continuous map}\}$$

I.3.2. The cochain algebra $(\mathcal{A}_{\mathcal{PL}})_n$

First, consider the free graded commutative algebra $\bigwedge(t_0, \ldots, t_n, y_0, \ldots, y_n)$, where the generators t_i have degree 0, and the generators y_j have degree 1. The differential d is given by

$$d(t_i) = y_i,$$

$$d(y_j) = 0.$$

The cochain algebra $(\mathcal{A}_{\mathcal{PL}})_n$ is defined as the quotient:

$$(\mathcal{A}_{\mathcal{PL}})_n = \frac{\bigwedge(t_0, \ldots, t_n, y_0, \ldots, y_n)}{\langle \sum t_i - 1, \sum y_j \rangle}. \tag{Eq. I.3.13}$$

Notice that this algebra is actually isomorphic to $\bigwedge(t_1, \ldots, t_n, y_1, \ldots, y_n)$. That is, the relations allow us to express t_0 in terms of $t_1, \ldots, t_n$ and y_0 in terms of $y_1, \ldots, y_n$. After eliminating t_0, y_0, the expression of each element is unique. However, it is sometimes more convenient to use also t_0, y_0 to write certain formulas, so we will use one expression or the other depending on the context.

I.3.3. The simplicial cochain algebra $\mathcal{A}_{\mathcal{PL}}$

We define now a simplicial object in the category of cochain algebras, called the **simplicial cochain algebra** $\mathcal{A}_{\mathcal{PL}} = \{(\mathcal{A}_{\mathcal{PL}})_n\}_{n\geq 0}$ as follows:

- For each $n \geq 0$, the cochain algebra $(\mathcal{A}_{\mathcal{PL}})_n$ is the one defined above.
- The face and degeneration morphisms are the unique cochain algebra morphisms

$$\begin{array}{ll} \partial_i : & (\mathcal{A}_{\mathcal{PL}})_n \to (\mathcal{A}_{\mathcal{PL}})_{n-1}, \quad 0 \leq i \leq n, \\ s_j : & (\mathcal{A}_{\mathcal{PL}})_n \to (\mathcal{A}_{\mathcal{PL}})_{n+1}, \quad 0 \leq j \leq n, \end{array} \tag{Eq. I.3.14}$$

satisfying:

$$\partial_i(t_k) = \begin{cases} t_k, & k < i \\ 0, & k = i \\ t_{k-1}, & k > i \end{cases}, \qquad s_j(t_k) = \begin{cases} t_k, & k < j \\ t_k + t_{k+1}, & k = j \\ t_{k+1}, & k > j \end{cases} \tag{Eq. I.3.15}$$

The structure of the simplicial cochain algebra $\mathcal{A}_{\mathcal{PL}}$ can be visualized in figure I.8, where horizontally is represented the simplicial structure for a given dimension n (subscript), and vertically the graded structure of the algebra (representing the degree p, with superscript). This object has a triangular structure due to the vanishing of the elements when $p > n$.

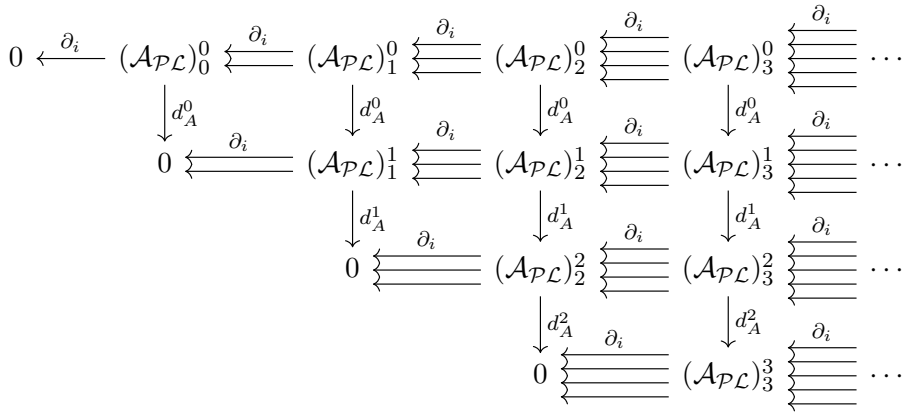


FIGURE I.8. The structure of the simplicial cochain algebra $(\mathcal{A}_{\mathcal{PL}})_n$.

I.3.4. The construction $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$

Let $\mathcal{K}$ be a simplicial set, and let $\mathcal{A}_{\mathcal{PL}} = \{(\mathcal{A}_{\mathcal{PL}})_n\}_{n\geq 0}$ be the simplicial cochain algebra defined earlier. Then

$$\mathcal{A}_{\mathcal{PL}}(\mathcal{K}) = \bigoplus_{p\geq 0} \mathcal{A}^p_{\mathcal{PL}}(\mathcal{K})$$

is the cochain algebra defined as follows:

- The homogenous part $\mathcal{A}^p_{\mathcal{PL}}(\mathcal{K})$ is the set of simplicial set morphisms from $\mathcal{K}$ to $\mathcal{A}_{\mathcal{PL}}$ where an element $\Phi^p \in \mathcal{A}^p_{\mathcal{PL}}(\mathcal{K})$ is defined as a collection of simplicial maps $\Phi^p = \{\Phi^p_n : \mathcal{K}_n \longrightarrow (\mathcal{A}_{\mathcal{PL}})^p_n\}_{n\geq 0}$ whose elements satisfy the following restrictions:

$$\begin{aligned} \Phi^p_{n-1}(\partial_i(\sigma)) &= \partial_i(\Phi^p_n(\sigma)), \\ \Phi^p_{n+1}(s_j(\sigma)) &= s_j(\Phi^p_n(\sigma)), \end{aligned} \qquad \text{(Eq. I.3.16)}$$

  for all $\sigma \in \mathcal{K}_n$.
- Addition, scalar multiplication, product and the differential are induced by the corresponding operations in the algebras $(\mathcal{A}_{\mathcal{PL}})_n$.

We denote by $\mathcal{A}^p_{\mathcal{PL}}(\mathcal{K})_n$ the subset of simplicial maps from $\mathcal{K}_n$ to $(\mathcal{A}_{\mathcal{PL}})^p_n$. The structure of the cochain algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, at some general dimension n and degree p, can be visualized in figure I.9. The horizontal direction corresponds to the simplicial structure for a given dimension n (subscript), and the vertical direction corresponds to the graded structure of the algebra (representing the degree p with superscript). As the simplicial cochain algebra $(\mathcal{A}_{\mathcal{PL}})_n$ is at the ground of this structure, all the elements of $\mathcal{A}^p_{\mathcal{PL}}(\mathcal{K})_n$ will vanish when $p > n$. This is because it is not possible to assign non-zero polynomial forms of degree greater than the dimension of the simplices.

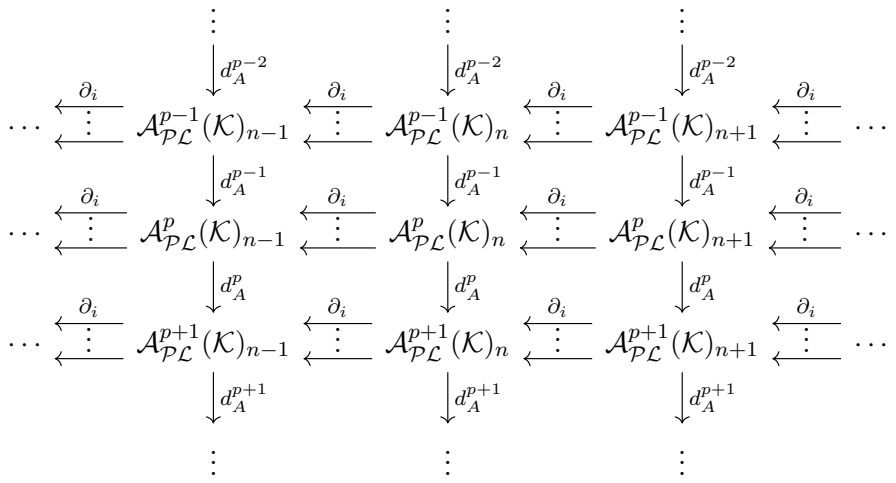


FIGURE I.9. The structure of the cochain algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. Each row represents the set of homogeneous elements $\Phi^p \in \mathcal{A}^p_{\mathcal{PL}}(\mathcal{K})$ and each column represents the complex structure defined over the elements $\Phi^*_n \in \mathcal{A}^*_{\mathcal{PL}}(\mathcal{K})_n$.

I.3.5. The quasi-isomorphism $\mathcal{A}_{\mathcal{PL}}(\mathcal{S}_*(\mathcal{X})) \simeq \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$

DEFINITION (I.3.2). Consider a continuous map $f : \mathcal{X} \to \mathcal{Y}$ between pointed topological spaces. This map is called a *weak homotopy equivalence* if it induces isomorphisms

$$\pi_n(f) : \pi_n(\mathcal{X}, x_0) \to \pi_n(\mathcal{Y}, f(x_0))$$

for all $n \geq 0$ and all choices of basepoint x_0. The spaces $\mathcal{X}$ and $\mathcal{Y}$ have the same weak homotopy type if there is a finite chain of weak homotopy equivalences

$$\mathcal{X} \leftarrow \mathcal{Z}_1 \to \cdots \leftarrow \mathcal{Z}_n \to \mathcal{Y}.$$

DEFINITION (I.3.3). A cellular model or *CW model* for a space $\mathcal{Y}$ is a $\mathcal{CW}$ complex $\mathcal{X}$ together with a weak homotopy equivalence $f : \mathcal{X} \to \mathcal{Y}$.

It is known that every space $\mathcal{Y}$ has a $\mathcal{CW}$ model and this model is unique up to homotopy equivalence. Two important theorems are in the foundations of Rational Homotopy Theory:

THEOREM (I.3.4) (Whitehead [**39**, p. 346]). *If a map $f : \mathcal{X} \to \mathcal{Y}$ between connected $\mathcal{CW}$ complexes induces isomorphisms $\pi_*(f) : \pi_*(\mathcal{X}, x_0) \to \pi_*(\mathcal{Y}, y_0)$, $\forall n$, then f is a homotopy equivalence.*

THEOREM (I.3.5) (Whitehead-Serre [**21**, p. 94]). *Let $f : \mathcal{X} \to \mathcal{Y}$ be a map between simply connected spaces. Then the following assertions are equivalent for a subring $\mathcal{R}$ of $\mathbb{Q}$:*

- *$\pi_*(f) \otimes_{\mathbb{Z}} \mathcal{R}$ is an isomorphism*
- *$\mathcal{H}_*(f, \mathcal{R})$ is an isomorphism*

In this work, instead of using $\mathcal{CW}$ models, for computational reasons, these structures are replaced by either simplicial complexes or simplicial sets. Because Rational Homotopy theory is built upon the construction of $\mathcal{CW}$ models and its singular homology, two important theorems have to be mentioned in order to validate the use of simplicial complexes for the purpose of this work.

The first theorem asserts the existence of a homotopy equivalent simplicial complex $\mathcal{Y}$ for every $\mathcal{CW}$ complex $\mathcal{X}$:

THEOREM (I.3.6) ([**39**, p. 182]). *Every $\mathcal{CW}$ complex $\mathcal{X}$ is homotopy equivalent to a simplicial complex $\mathcal{Y}$, which can be chosen to be of the same dimension as $\mathcal{X}$, finite if $\mathcal{X}$ is finite, and countable if $\mathcal{X}$ is countable.*

The second theorem is important because it allows us to work with the simplicial homology of a given simplicial complex $\mathcal{X}$, as if we were working with the singular homology, object used in [**21**] for the building of the Rational Homotopy Theory's corpus:

THEOREM (I.3.7) (Equivalence of Simplicial and Singular Homology [**39**, p. 128]). *Given a simplicial complex, $\mathcal{X}$, the simplicial homology and the singular homology of $\mathcal{X}$ are isomorphic.*

THEOREM (I.3.8) ([**21**, p. 58]). *If $\varphi : \mathcal{X} \to \mathcal{Y}$ is a weak homotopy equivalence, then*

$$\mathcal{C}_*(\varphi) : \mathcal{C}_*(\mathcal{X}, \mathbb{K}) \to \mathcal{C}_*(\mathcal{Y}, \mathbb{K})$$

is a quasi-isomorphism.

THEOREM (I.3.9) ([**39**, p. 196]). *If a chain map between chain complexes of free abelian groups induces an isomorphism on homology groups, then it induces an isomorphism on cohomology groups with any coefficient group $\mathbb{G}$.*

COROLLARY (I.3.10). *If $\varphi : \mathcal{X} \to \mathcal{Y}$ is a weak homotopy equivalence, then*

$$\mathcal{C}^*(\varphi) : \mathcal{C}^*(\mathcal{X}, \mathbb{K}) \to \mathcal{C}^*(\mathcal{Y}, \mathbb{K})$$

is a quasi-isomorphism.

THEOREM (I.3.11) ([**29**, p. 284]). *A simplicial map $f : \mathcal{Y} \to \mathcal{X}$ induces a chain homomorphism*

$$\mathcal{C}(f) = \{\mathcal{C}_n(f) = (\mathcal{C}(f)_n, n \in \mathbb{N}\} : \mathcal{C}(\mathcal{Y}) \to \mathcal{C}(\mathcal{X}),$$

obtaining the chain complex functor $\mathcal{C}(-)$ from the category of simplicial sets to the category of chain complexes.

PROPOSITION (I.3.12) ([**29**, p. 284]). *If $f_\lambda : \mathcal{Y} \to \mathcal{X}$, with $\lambda = 0, 1$, are homotopic simplicial maps, then $(f_0)_* = (f_1)_*$. Consequently, simplicial hotomopy equivalences induce isomorphisms on homology.*

DEFINITION (I.3.13) ([**29**, p. 152]). Geometric realization functor $|\cdot| : \mathcal{SiSets} \longrightarrow \mathcal{Top}$

PROPOSITION (I.3.14) ([**29**, p. 156]). *The functor $|\cdot|$ has the right adjoint $\mathcal{S} : \mathcal{Top} \longrightarrow \mathcal{SiSets}$.*

Let $\mathcal{X} = \{\mathcal{X}_n\}$ be a simplicial set, $|\mathcal{X}|$ its geometric realization, $\mathcal{T}$ a topological space, and $\mathcal{S}(\mathcal{T})$ the simplicial set of singular simplices of a topological space. We also define the maps $\eta_\mathcal{X}$ and $j_\mathcal{T}$ as:

DEFINITION (I.3.15) ([**29**, p. 156]). The unit map $\eta_\mathcal{X} : \mathcal{X} \longrightarrow \mathcal{S}(|\mathcal{X}|)$ is the map given by the simplicial embedding of $\mathcal{X}$ onto $\mathcal{S}(|\mathcal{X}|)$, associating to a simplex $x \in \mathcal{X}$, the singular simplex

$$\eta_\mathcal{X}(x) : \Delta^n \longrightarrow |\mathcal{X}|, \;\; t \to [x, t],$$

where t is any point in the standard n-simplex Δ^n.

Definition (I.3.16) ([**29**, p. 156]). The co-unit map $j_{\mathcal{T}} : |\mathcal{S}(\mathcal{T})| \longrightarrow \mathcal{T}$ is the map given by is called the simplicial embedding associating to a simplex $x \in \mathcal{X}$, the singular simplex

$$j_{\mathcal{T}}([x,t]) = x(t)$$

for every singular simplex $x : \Delta^n \longrightarrow \mathcal{T}$ and every point $t \in \Delta^n$.

Proposition (I.3.17) ([**29**, p. 157]). *The adjointness of (I.3.15) and (I.3.16) gives a bijective correspondence between maps $g : |\mathcal{X}| \to \mathcal{T}$ and simplicial maps $f : \mathcal{X} \to \mathcal{S}(\mathcal{T})$. The adjoint of the map g is the simplicial map $g' = \mathcal{S}(g) \circ \eta_{\mathcal{X}}$, and the adjoint of the simplicial map f is the map $f' : j_{\mathcal{T}} \circ |f|$.*

In fact, [**29**, p. 157] the reason that these processes of forming adjoints are inverse to each other is due to the fundamental equations beteen unit and co-unit:

$$j_{|\mathcal{X}|} \circ |\eta_{\mathcal{X}}| = \mathbf{1}_{|\mathcal{X}|}, \qquad \mathcal{S}(j_{\mathcal{T}}) \circ \eta_{\mathcal{S}(\mathcal{T})} = \mathbf{1}_{\mathcal{S}(\mathcal{T})}.$$

Proposition (I.3.18) ([**29**, p. 189]). *The map $j_{\mathcal{T}}$ is a weak homotopy equivalence, for every space $\mathcal{T}$.*

Theorem (I.3.19) (Corollary 4.5.31, [**29**, p. 190]). **1)** *The co-unit map $j_{\mathcal{T}} : |\mathcal{S}(\mathcal{T})| \to \mathcal{T}$ is a homotopy equivalence, for any $\mathcal{CW}$-complex $\mathcal{T}$.*

2) *A map $f : \mathcal{U} \to \mathcal{T}$ is a weak homotopy equivalence iff $|\mathcal{S}(f)| : |\mathcal{S}(\mathcal{U})| \to |\mathcal{S}(\mathcal{T})|$ is a homotopy equivalence.*

3) *Let, $\mathcal{Y}, \mathcal{X}$ be Kan sets. Then, a simplicial map $f : \mathcal{Y} \to \mathcal{X}$ is a simplicial homotopy equivalence iff $|f| : |\mathcal{Y}| \to |\mathcal{X}|$ is a homotopy equivalence.*

4) *A map $f : \mathcal{U} \to \mathcal{T}$ is a weak homotopy equivalence iff $\mathcal{S}(f) : \mathcal{S}(\mathcal{U}) \to \mathcal{S}(\mathcal{T})$ is a simplicial homotopy equivalence.*

Proposition (I.3.20) ([**29**, p. 187]). *Any simplicial set $\mathcal{X}$ can be considered as a simplicial subset of the singular set $\mathcal{S}(|\mathcal{X}|)$, via the injective simplicial map $\eta_{\mathcal{X}}$.*

Lemma (I.3.21) ([**21**], p. 118). *If $\mathcal{K}$ is any simplicial set then any $\sigma \in \mathcal{K}_n$ determines a unique simplicial set map $\sigma_* : \Delta[n] \to \mathcal{K}$ such that $\sigma_*(c_n) = \sigma$.*

Let $\mathcal{K}$ be a simplicial set. By theorems (I.3.6), (I.3.19) and proposition (I.3.20), we can deduce the following results:

1) The continuous map

$$j_{\mathcal{K}} : |\mathcal{S}(|\mathcal{K}|)| \to |\mathcal{K}|,$$

is a homotopy equivalence.

2) The inclusion given by

$$\eta_{\mathcal{K}} : \mathcal{K} \to \mathcal{S}(|\mathcal{K}|),$$

is a simplicial homotopy equivalence.

3) The induced morphism by the contravariant functor $\mathcal{A}_{\mathcal{PL}}(-)$:

$$\mathcal{A}_{\mathcal{PL}}(\eta_{\mathcal{K}}) : \mathcal{A}_{\mathcal{PL}}(\mathcal{S}(|\mathcal{K}|)) \to \mathcal{A}_{\mathcal{PL}}(\mathcal{K}),$$

is a quasi-isomorphism of cochain algebras.

Now, consider a topological space $\mathcal{X}$, and (following the definition (I.3.3) and theorem (I.3.6)) a simplicial set $\mathcal{K}$ together with a continuous map φ, such that $\varphi : |\mathcal{K}| \to \mathcal{X}$ is a weak homotopy equivalence. By theorem (I.3.19) we get that:

$$\mathcal{S}(\varphi) : \mathcal{S}(|\mathcal{K}|) \to \mathcal{S}(\mathcal{X})$$

is a simplicial hotomopy equivalente, and then

$$\mathcal{S}(\varphi) : \mathcal{A}_{\mathcal{PL}}(\mathcal{S}(\mathcal{X})) \to \mathcal{A}_{\mathcal{PL}}(\mathcal{S}(|\mathcal{K}|))$$

is a quasi-isomorphism of cochain algebras.
Summing up these results, we get that:

THEOREM (I.3.22). *The simplicial set morphism* $\mathcal{S}(\varphi) \circ \eta_{\mathcal{K}} : \mathcal{K} \to \mathcal{S}(\mathcal{X})$ *induces by the functor* $\mathcal{A}_{\mathcal{PL}}(-)$ *the quasi-isomorphism of cochain algebras:*

$$\mathcal{A}_{\mathcal{PL}}(\varphi) : \mathcal{A}_{\mathcal{PL}}(\mathcal{S}(\mathcal{X})) \to \mathcal{A}_{\mathcal{PL}}(\mathcal{K}).$$

Finally, we arrive to the point we were looking for that guarantee us the effectiveness of working with simplicial models $\mathcal{K}$ instead of the singular simplicial set of a topological space:

COROLLARY (I.3.23). *The simplicial sets* $\mathcal{K}$ *and* $\mathcal{S}(\mathcal{X})$ *have isomorphic minimal models.*

Given a simply connected space $\mathcal{X}$, a simplicial complex (or simplicial set) $\mathcal{Y}$ homotopy equivalent to a $\mathcal{CW}$ model of such space, and combining Theorems (I.3.5) and (I.3.7), we can study the rational homotopy type of $\mathcal{X}$ through $\mathcal{Y}$.

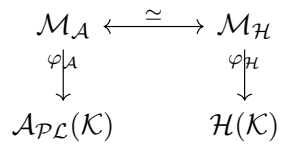
Given a simplicial set $\mathcal{K}$, we will construct a CDGA, denoted by $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. The minimal model of the topological space modeled by $\mathcal{K}$ is defined as the minimal model of this algebra. In the next sections are presented all the elements needed for the construction of this algebra.

I.3.6. Formality

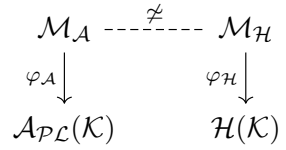
We conclude the chapter mentioning an important invariant in homotopy theory. Recall from section (I.1.15) the definition of formality for *CDGAs*. Then, we make the following definition of formality for topological spaces:

DEFINITION (I.3.24). A Sullivan space is called *formal* if the minimal model of the space and the model of its cohomology algebra are isomorphic. On the other hand, if the two models are not isomorphic, it is said that the space is *not formal*.

Although the definition of Sullivan spaces has not been introduced, the reader can think about simply-connected spaces as an example of such spaces (for a proper definition see [**22, 26**]). The important idea behind formality is that, for a Sullivan space $\mathcal{X}$, its minimal model $(\mathcal{M}_{\mathcal{X}}, d_{\mathcal{M}})$ fully captures the rational homotopy type of $\mathcal{X}$. Furthermore, if $\mathcal{X}$ is formal, the minimal model of its cohomology algebra $(\mathcal{M}_{\mathcal{H}}, d_{\mathcal{H}})$ and $(\mathcal{M}_{\mathcal{X}}, d_{\mathcal{M}})$ are isomorphic. Consequently, it carries the same homotopical information. In that case, it is said that the rational homotopy type of $\mathcal{X}$ is a *formal* consequence of its cohomology (see Figure I.10a). However, if a space is not formal, the minimal model of its cohomology does not give as information about the rational homotopy type of such space (see Figure I.10b).



(A) A formal algebra $\mathcal{A}$.



(B) A non-formal algebra $\mathcal{A}$.

FIGURE I.10. If the minimal model of the space $\mathcal{M}_{\mathcal{A}}$ and the model of the cohomology algebra M_H are not isomorphic, it is said that the space is not formal.

The main contribution of this thesis consist in the effective computation of the minimal model of a simplicial set (viewed as a cellular model of a topological space). A direct consequence of this contribution is the possibility to apply the i-formality criteria defined at [**47**] for the effective computation of the formality decision problem for simplicial sets.

II

Revisiting the PL de Rham theory

In this chapter we start in II.1 introducing the concept of Whitney forms as a key elements in the development of this thesis. We follow in II.2 studying their algebraic properties in relation to the algebra of polynomial differential forms $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, giving a proof of the well known Dupont's formula for the cochain product. As a result of the deductive procedure, we conjecture a generalization of such formula for the product of an arbitrary sequence of cochains. Finally, at section **??** we define the Dupont's contraction $s : \mathcal{A}^*_{PL}(\mathcal{K}) \to \mathcal{A}^{*-1}_{PL}(\mathcal{K})$, a key component for the description of the main contribution of this thesis, i.e., an effective method (with a concrete implementation) for the computation of the minimal model of a simplicial set. This method will be described in the following chapter.

Section § II.1
Whitney elementary forms

George de Rham [**15**] in the 1930's proved that integration provides a natural direct quasi-isomorphism of cochain complexes that connects the algebra of polynomial differential forms (in this context $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$) with the corresponding cochain complex $\mathcal{C}(\mathcal{K})$ associated to $\mathcal{K}$:

$$\oint : \mathcal{A}_{\mathcal{PL}}(\mathcal{K}) \to \mathcal{C}(\mathcal{K})$$

as follows. Let $\int_n$ be the linear map:

$$\int_n : (\mathcal{A}_{\mathcal{PL}})^n_n \to \mathbb{Q}$$

by setting

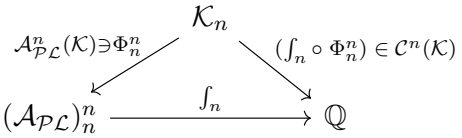
$$\begin{aligned}\int_n t_1^{k_1} \dots t_n^{k_n} dt_1 \wedge \dots \wedge dt_n = \int_0^1 dt_1 \int_0^{1-t_1} dt_2 \dots \\ \dots \int_0^{1-\sum_1^{n-1} t_i} t_1^{k_1} \dots t_n^{k_n} dt_n = \\ = \frac{k_1! k_2! \dots k_n!}{(k_1 + \dots + k_n + n)!}.\end{aligned} \qquad \text{(Eq. II.1.17)}$$

This map represents the integration $\int_{\Delta^n} \omega$ of an n-form $\omega \in (\mathcal{A}_{\mathcal{PL}})^n_n$ over the standard n-simplex Δ^n. Using this map, we can integrate elements of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ as follows: consider a homogeneous

element $\Phi^n \in \mathcal{A}^n_{\mathcal{PL}}(\mathcal{K})$, and $\Phi^n_m : \mathcal{K}_m \to (\mathcal{A}_{\mathcal{PL}})^n_m$ its restriction to simplicies of dimension m. We define the integral of the homogeneous element Φ^n as the map:

$$\begin{array}{rcl} \oint : \mathcal{A}^n_{\mathcal{PL}}(\mathcal{K}) & \to & \mathcal{C}^n(\mathcal{K}) \\ \Phi^n & \to & (\int_n \circ \Phi^n_n). \end{array}$$

Furthermore, the integral of the element Φ^n is the element of $\mathcal{C}^n(\mathcal{K})$ determined by the integration map $\int_n$ over the set of n-simplicies:



Then, the map is extended to all $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ by linearity in order to define the integration of a generic (not neccesarily homogeneous) element $\Psi \in \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. As it is represented in figure II.1, this map commutes with the differential, i.e., $\oint \circ d_{\mathcal{A}} = d_{\mathcal{C}} \circ \oint$, and, the most important, it defines a quasi-isomorphism of cochain complexes:

$$\begin{array}{ccc} \mathcal{A}^{p-1}_{\mathcal{PL}}(\mathcal{K}) & \xrightarrow{d^{p-1}_{\mathcal{A}}} & \mathcal{A}^{p}_{\mathcal{PL}}(\mathcal{K}) \\ \downarrow \oint^{p-1} & & \downarrow \oint^{p} \\ \mathcal{C}^{p-1}(\mathcal{K}) & \xrightarrow{d^{p-1}_{\mathcal{C}}} & \mathcal{C}^{p}(\mathcal{K}) \end{array}$$

FIGURE II.1. The commutative diagram between $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ and $\mathcal{C}(\mathcal{K})$.

Latter, Hassler Whitney [**62**, sections IV.27 and VII.11], in his seminal work on geometric integration, proved that, for a discrete simplicial set $\mathcal{K}$, it is possible to define a linear map:

$$\mathcal{W} : \mathcal{C}(\mathcal{K}) \to \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$$

satifying:

$$\begin{array}{rcl} \mathcal{W}(\beta|_{\sigma_1})(\sigma_2) & = & 0 \in (\mathcal{A}_{\mathcal{PL}})_{|\sigma_2|}, \quad if \quad \sigma_2 \notin St(\sigma_1), \\ \oint \circ \mathcal{W} & = & id_{\mathcal{C}}, \\ d_{\mathcal{A}} \circ \mathcal{W} & = & \mathcal{W} \circ d_{\mathcal{C}}, \\ \mathcal{W}(1_{\mathcal{C}}) & = & 1_{\mathcal{A}}, \end{array} \qquad \text{(Eq. II.1.18)}$$

where

- $\beta \in \mathcal{C}(\mathcal{K})$,
- $\beta|_\sigma$ means the evaluation of the cochain β at simplex σ,
- $\sigma_1, \sigma_2 \in \mathcal{K}$,
- $|\sigma|$ denotes the dimension of the simplex,
- $St(\sigma)$ is the star of the simplex,
- $1_{\mathcal{C}}$ is the constant map 1 in $\mathcal{C}^0(\mathcal{K})$, and
- $1_{\mathcal{A}}$ is the constant map 1 in $\mathcal{A}^0_{\mathcal{PL}}(\mathcal{K})$.

Furthermore, Whitney gave us an explicit construction of this map. The elements $\omega \in \mathrm{Im}(\mathcal{W})$ are called *constant forms*, *Whitney elementary forms*, or just *Whitney forms* . Whitney forms are a mathematical concept used in several areas, from pure and applied mathematics to physics. In particular, Whitney forms are at the core of the relatively young field of Finite Elements Exterior Calculus (FEEC, [**6, 4, 5**]). Their definition is given below. Subsequently, based on the recently published work of Lohi and Kettunen [**43**], where they give a detailed description

and applications of Whitney forms in other contexts, in the following section are summarized the key properties of Whitney forms.

II.1.1. Whitney elementary forms

The key idea behind Whitney forms is the following: consider two simplices $\sigma_p \in \mathcal{K}_p$ and $\sigma_n \in \mathcal{K}_n$ of dimensions p and n respectively, so that σ_p is a q-face of σ_n determined by the face path $I = (i_1, \dots, i_q)$, where $i_1 < \cdots < i_q$, (for the definition of a face path see I.2.4). Let $(\lambda_0^p, \dots, \lambda_p^p)$ and $(\lambda_0^n, \dots, \lambda_n^n)$ be the local barycentric coordinates of σ_p and σ_n at their respective dimensions. From the definition of barycentric coordinates on simplicial sets (see I.2.5), we know that the inclusion map $\sigma_p \hookrightarrow \sigma_n$ identified by the tuple $J = (j_0, \dots, j_p)$ (with $j_0 < \cdots < j_p$) determines an affine map between such simplices where the image of σ_p on σ_n, as an affine subspace, is given by the linear span of the image of its barycentric coordinates,i.e.:

$$\begin{array}{rccc} i_J: & \sigma_p & \hookrightarrow & \sigma_n \\ & \lambda_k^p & \to & \lambda_{j_k}^n, \end{array}$$

and $\lambda_i^n := 0$, $\forall i \in I$. Due to the relation between the tuples I and J, notice that the inclusion map i_J determines a unique face path $\overline{\partial}_I : \mathcal{K}_n \to \mathcal{K}_p$ such that $\overline{\partial}_I(\sigma_n) = \sigma_p$, and vice versa. For a graphical illustration of this homomorphism see Figure II.2. As I and J are complementary sets over $\{0, \dots, n\}$, from now on, if there is no room for ambiguity, we will use the notation I indistinctly for both face paths and the index set of barycentric coordinates determined by such path.

Now, consider the map $\omega_n^n : \mathcal{K}_n \to (\mathcal{A}_{\mathcal{PL}})_n^n$ given by

$$\omega_n^n(\sigma) = (n!) \sum_{i=0}^{n} (-1)^i \cdot t_i \cdot y_0 \wedge \cdots \wedge \hat{y_i} \wedge \cdots \wedge y_n. \qquad \text{(Eq. II.1.19)}$$

where $\hat{y_i}$ means the variable is ommitted. Due to the relations

$$\sum_{i=0}^{n} t_i = 1, \text{ and } \sum_{j=0}^{n} y_j = 0,$$

implicit in the definition of $(\mathcal{A}_{\mathcal{PL}})_n$, it is possible to check by direct computations that the image of ω_n is equivalent to the form:

$$\omega_n^n(\sigma) = (n!) \cdot y_1 \dots y_n. \qquad \text{(Eq. II.1.20)}$$

In particular, notice that, by definition, $\omega_0^0(\sigma \in \mathcal{K}_0) = 1$. Looking at the integration formula, it is easy to see that $(\int_n \circ\, \omega_n^n) = 1_{\mathbb{Q}}$. This motivates us to make the following definiton:

Definition (II.1.1). The map ω_n^n is called the *unit volume map* and its image is called the *unit volumen form*.

Recall from definitions (I.2.7) and (I.2.5) that $\overline{\partial}_p^n$ is the set of simplicial paths from dimension n to dimension p and, as we said before, for a given n-simplex σ_n, an element $\gamma = \overline{\partial}_I \in \overline{\partial}_p^n$ determines a unique inclusion map $i_J : \sigma_p \hookrightarrow \sigma_n$ of a p-simplex $\sigma_p = \gamma(\sigma_n)$ on σ_n. This inclusion is uniquely represented by the subset of barycentric coordinates $\lambda_J^n = (\lambda_{j_0}^n, \dots, \lambda_{j_p}^n)$, where $j_k < j_l$ if $k < l$ and each $j_k \in [n]$. Then, for a given non-negative integers p, n such that $p < n$, consider the map $\omega_n^p : \mathcal{K}_n \times \overline{\partial}_p^n \to (\mathcal{A}_{\mathcal{PL}})_n^p$ given by

$$\omega_n^p(\sigma_n, \gamma) = (p!) \sum_{i=0}^{p} (-1)^i \cdot t_{j_i} \cdot y_{j_0} \wedge \cdots \wedge \hat{y}_{j_i} \wedge \cdots \wedge y_{j_p}. \qquad \text{(Eq. II.1.21)}$$

where as before, $\hat{y}_i$ means the variable is ommitted. Notice that ω_n^p and ω_n^n are equivalent when $p = n$.

Definition (II.1.2). The map ω_n^p described above is called the *Whitney propagation map* (or just the *propagation map*).

Now we can define the Whitney map. Let $\beta \in \mathcal{C}^p(\mathcal{K})$ be a p-cochain, and denote by $\beta(\sigma_p) \in \mathbb{Q}$ the evaluation of the cochain β at the p-simplex σ_p. When working with non-degenerated simplices, for a face path $\gamma \in \overline{\partial}_p^n$, and a simplex $\sigma_n \in \mathcal{K}_n$, due to Eq. I.2.11, if $\gamma(\sigma_n) = \emptyset$, we set $\beta(\gamma(\sigma_n)) := 0$. Then, we define the Whitney map as follows:

Definition (II.1.3). The *Whitney map* $\mathcal{W}^p : \mathcal{C}^p(\mathcal{K}) \to \mathcal{A}^p_{\mathcal{PL}}(\mathcal{K})$ is the linear map defined as follows: for a simplex $\sigma_n \in \mathcal{K}_n$, $\mathcal{W}^p(\beta)(\sigma_n) = 0$ if $n < p$, and, for $n \geq p$:

$$\mathcal{W}^p(\beta)(\sigma_n) = \sum_{\gamma \in \overline{\partial}_p^n} \beta(\gamma(\sigma_n)) \cdot \omega_n^p(\sigma_n, \gamma). \qquad \text{(Eq. II.1.22)}$$

Notice that the summatory on Eq. II.1.22 runs for all face paths from dimension n to dimension p. We conclude with the definition of Whitney forms:

Definition (II.1.4). The elements on the image of the Whitney map $\mathcal{W}$ are called *Whitney forms*.

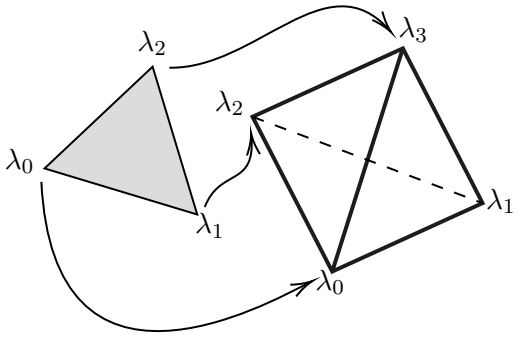


Figure II.2. Representation of the inclusion $i : \Delta^2 \hookrightarrow \Delta^3$ determining the face ∂_1 of Δ^3.

In [**43**], three equivalent definitions of Whitney forms are described. Also, the authors compiled and unified notation for a list of known properties of Whitney forms in different contexts. In order to formulate the key properties we need for the rest of the document, we skecth out below the main properties but for an extensive description and proofs we refer the reader to the citated article and the classical work of Hassler Whitney [**62**, sections IV.27 and VII.11].

Properties of Whitney forms

With some abuse of notation, let $\mathcal{W}^*(\mathcal{K}) = \{\mathcal{W}^p(\mathcal{C}^p(\mathcal{K}))\}_{p \geq 0}$ be the graded module formed by the image of the Whitney map at each degree. This graded module is actually a cochain subcomplex $\mathcal{W}^p(\mathcal{K}) \subset \mathcal{A}^p_{\mathcal{PL}}(\mathcal{K})$ due to fact that the differential $d_{\mathcal{A}}(\mathcal{W}^p) \subset \mathcal{W}^{p+1}$ is closed on $\mathcal{W}^*(\mathcal{K})$ and determines the exact sequence (see [**43**], proposition 4.7):

$$\cdots \longrightarrow \mathcal{W}^{p-1} \xrightarrow{d^{p-1}_{\mathcal{A}}} \mathcal{W}^p \xrightarrow{d^p_{\mathcal{A}}} \mathcal{W}^{p+1} \longrightarrow \cdots .$$

Moreover, an important property of this subcomplex is the following:

Proposition (II.1.5) (Isomorphism of cochain complexes,[**43**]). *The cochain complex* $\mathcal{W}^*(\mathcal{K})$ *is isomorphic to* $\mathcal{C}^*(\mathcal{K})$.

This isomorphism implies that the following diagram:

$$\begin{array}{ccc} \mathcal{W}^{p-1}(\mathcal{K}) & \xrightarrow{d^{p-1}_{\mathcal{A}}} & \mathcal{W}^{p}(\mathcal{K}) \\ \oint\downarrow \uparrow \mathcal{W} & \circlearrowleft & \oint\downarrow \uparrow \mathcal{W} \\ \mathcal{C}^{p-1}(\mathcal{K}) & \xrightarrow[d^{p-1}_{\mathcal{C}}]{} & \mathcal{C}^{p}(\mathcal{K}) \end{array}$$

commutes in both directions and, restricting the integration map $\oint$ to the image of $\mathcal{W}$, we have that $\mathcal{W} \circ \oint = 1_{\mathcal{W}}$ and $\oint \circ \mathcal{W} = 1_{\mathcal{C}}$. In other words, the Whitney map $\mathcal{W} : \mathcal{C}(\mathcal{K}) \to \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ is a quasi-isomorphism of cochain complexes. Later we will see that, when $\oint$ is not restricted to Whitney forms, $\mathcal{W} \circ \oint \simeq 1_{\mathcal{A}}$ and Dupont found a simplicial contraction from $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ to $\mathcal{C}(\mathcal{K})$ that allows us to transfer an algebraic structure to the simplicial cochain complex (for the description of Dupont's contraction go to section II.3).
Now, recall that the algebra

$$(A_{PL})_n := \frac{\bigwedge(t_0, \ldots, t_n, y_0, \ldots, y_n)}{\langle \sum t_i - 1, \sum y_j \rangle}$$

is isomorphic to the exterior algebra $Ext_y = \frac{\bigwedge(y_0,\ldots,y_n)}{\langle \sum y_j \rangle}$ with coefficients in the polynomial ring $R_t := \frac{\mathbb{Q}[t_0,\ldots,t_n]}{\langle \sum t_i - 1 \rangle}$ so that

$$(A_{PL})_n \cong R_t \otimes Ext_y.$$

Taking into account this definition, an important property is the following:

Proposition (II.1.6) (Constant forms, [**43**]). *Whitney forms are first order polynomials in each cell.*

This statement says that, looking at the polynomial differential form assigned to each simplex σ by the Whitney map, all monimials have coeffients in R_t of degree at most 1. This can be easly seen by taking a look at equations Eq. II.1.20 and Eq. II.1.21. For a Whitney form of degree n, the differential form assigned to an n-simplex σ_n is a volume forms with constant coefficient (this is the reason why sometimes are called constant forms). This is not the general case, when a p-form is propagated to simplices of higher dimensions. However, when a constant cochain β is non-zero only at the horn of a simplex, the polynomial differential forms assigned to simplices of higher dimensions are also constant forms.

Section § II.2
Products of Whitney forms

In [**19**], using the Whitney map, Dupont showed that it is possible to transfer an algebraic structure to the simplicial cochain complex induced by the wedge product on $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ as follows: for $\alpha \in \mathcal{C}^p(\mathcal{K})$ and $\beta \in \mathcal{C}^q(\mathcal{K})$ two cochains of degrees p and q respectively, the product $\alpha \sqcup\!\sqcup \beta \in \mathcal{C}^{p+q}(\mathcal{K})$ is defined as the element

$$\alpha \sqcup\!\sqcup \beta = \oint (\mathcal{W}(\alpha) \wedge \mathcal{W}(\beta)).$$

In this section, we are going to give the direct computations Dupont mentioned in the cited paper in order to proof the following proposition:

PROPOSITION (II.2.1). *For cochains $\alpha \in \mathcal{C}^p(\mathcal{K})$ and $\beta \in \mathcal{C}^q(\mathcal{K})$, the element $\alpha \sqcup\!\sqcup \beta$ is the cochain:*

$$(\alpha \sqcup\!\sqcup \beta)(\sigma) = \frac{p!q!}{(p+q+1)!} \sum_{I,J} sgn(J,I) \cdot \alpha(\overline{\partial}_I \sigma) \cdot \beta(\overline{\partial}_J \sigma)$$

where $\sigma \in \mathcal{K}_{p+q}$, the multi-indices $I \in I_p^n$ and $J \in I_q^n$ run for all q- and p-face paths respectively, such that $|I \cap J| = 0$ (i.e., all face paths from dimension n to dimensions p and q that do not cross each other), and $sgn(J,I)$ is the sign of the permutation taking the tuple $J \frown I$ to $(0, \dots, \hat{m}, \dots, n)$, where $m = I^c \cap J^c$ is the unique index that acts as a pivoting barycentric coordinate for each pair of face paths (I, J).

In the original paper, and using our notation, Dupont expressed the condition $|I \cap J| = 0$ as $|I^c \cap J^c| = 1$, meaning that the product only runs for pairs of faces sharing just one barycentric coordinate. Also, he defined the sign $sgn(J,I)$ as $(-1)^{p-s+r}$ times the sign $sgn(I^c, J^c)$ computed as the permutation sign of taking the tuple $I^c \frown J^c$ to $(0, \dots, n)$, and using the notation $I^c \cap J^c = i_s = j_r$ to identify the shared index between I^c and J^c. This definitions are equivalent. Furthermore, our statement changes this narrative saying that the condition runs for non-crossing face paths. Later we will see that both conditions are equivalent and, actually, the summatory runs for every pair of face paths but the product is zero when more than one coordinate is shared (or equivalently, the product is zero if the face paths cross each other (I.2.6)).

Remark **(II.2.2)** (Path integral perspective)**.** Informally, the reason of changing narrative is because using these terms, the product $\alpha \sqcup\!\sqcup \beta$ suggests that it can be defined as a path integral that computes the sum of the joint contribution of the elements α and β at dimension n, taking into account all possible and unique paths to reach the simplex σ. In this direction, as the work of Dupont is based on the topological structure of the underlying space, it could be interesting to explore the potential relation of this idea with the works [**12, 61, 41**] that use simplicial structures as models in the context of statistical physics, characterization of complex systems dynamics and the study of phase transitions.

The rest of the section is structured as follows: first, a change of notation and some definitions are introduced in the context of Dupont's simplicial computations; then, some combinatorial results that will be useful later are listed; and finally, in section II.2.1 the proof of proposition (II.2.1) is described.

Notation

First of all, if there is no room for ambiguity, the notation $I = (i_1, \dots, i_q)$ will be used interchangeably to define I as an strictly increasing sequence of q non-negative integers, a tuple or a set. Recall from I.2.4 that $\overline{\partial}_p^n$ is the set of face paths from dimension n to dimension p. An element $\gamma = \overline{\partial}_I \in \overline{\partial}_p^n$ is called a *q-face path* from dimension n to dimension p. As a q-face path $\overline{\partial}_I = \partial_{i_1} \circ \cdots \circ \partial_{i_q}$ is determined uniquely by the strictly increasing sequence $I = (i_1, \dots, i_q)$, by abuse of notation, at some points we will refer to the face path I instead of $\overline{\partial}_I$.

On the other hand, notice that Dupont's formula for the induced product at simplicial cochain level is a local product, in the sense that, for cochains $\alpha \in \mathcal{C}^p(\mathcal{K})$ and $\beta \in \mathcal{C}^q(\mathcal{K})$, and

$\sigma_1, \sigma_2 \in \mathcal{K}_{p+q}$ the computations of $(\alpha \sqcup\!\sqcup \beta)(\sigma_1)$ and $(\alpha \sqcup\!\sqcup \beta)(\sigma_2)$ are carried out independently. For that reason, and for the rest of the section, we fix the following notation: let $\alpha \in \mathcal{C}^p(\mathcal{K})$ and $\beta \in \mathcal{C}^q(\mathcal{K})$ be two cochains of degrees p and q respectively; $\sigma_p \in \mathcal{K}_p$, $\sigma_q \in \mathcal{K}_q$ and $\sigma_n \in \mathcal{K}_n$ are three simplices such that $n = p + q$. Then, we will denote by $\omega_\alpha = \mathcal{W}(\alpha)$ and $\omega_\beta = \mathcal{W}(\beta)$ the Whitney forms associated to each cochain.

Definition (II.2.3). The element ω_α is called the *Whitney form of degree p and weight α*. If there is no ambiguity, we will omit the degree and we will just say that ω_α is the *Whitney form of weight α*.

If σ_p, σ_q and σ_n are three simplices such that $n = p + q$ and

$$\sigma_p = \overline{\partial}_I \sigma_n,$$
$$\sigma_q = \overline{\partial}_J \sigma_n,$$

for some face paths $I = (i_1, \ldots, i_q)$ and $J = (j_1, \ldots, j_p)$, we will denote by $\omega_I = \omega_n^p(\sigma_n, \overline{\partial}_I)$ and $\omega_J = \omega_n^q(\sigma_n, \overline{\partial}_J)$ the propagation of the unit volume forms $\omega_p^p(\sigma_p)$ and $\omega_q^q(\sigma_q)$ to the simplex σ_n along face paths I and J respectively. As Dupont's formula is applied locally, this notation will be useful when considering all computations carried out at the same simplex σ_n. Also, if we consider the sequences I, J as subsets of $[n] = \{0, \ldots, n\}$, then the multi-indices $I^c = (i_0, \ldots, i_p)$ and $J^c = (j_0, \ldots, j_q)$ determine the inclusions $i_{I^c}(\sigma_p) = \langle \lambda_{i_0}, \ldots, \lambda_{i_p} \rangle$ and $i_{J^c}(\sigma_q) = \langle \lambda_{j_0}, \ldots, \lambda_{j_q} \rangle$ on $\sigma_n = \langle \lambda_0, \ldots, \lambda_n \rangle$. Now, consider the Whitney form of weight α, ω_α. For a concrete n-simplex σ_n, we will use for convenience both sides of the following equality

$$\omega_\alpha|_{\sigma_n} = \sum_{I \in I_p^n} \alpha(\partial_I \sigma_n) \cdot \omega_I,$$

to denote the element $\mathcal{W}(\alpha)(\sigma_n)$. We make the following definition:

Definition (II.2.4). Given $\alpha \in \mathcal{C}^p(\mathcal{K})$, a q-face path I, and a simplex $\sigma \in \mathcal{K}_{p+q}$, we will say that $\alpha(\overline{\partial}_I \sigma)$ is the *weight* of the face path I at σ.

Definition (II.2.5). We call the *unit* cochain of degree p to the cochain $\alpha \in \mathcal{C}^p(\mathcal{K})$ such that

$$\alpha(\sigma_p) = 1, \forall \sigma_p \in \mathcal{K}_p.$$

Definition (II.2.6). Given a unit cochain $\alpha \in \mathcal{C}^p(\mathcal{K})$, we define the *Whitney unit form* of degree p at dimension n as the element

$$\omega_1|_p^n = \sum_{I \in I_p^n} \omega_I.$$

Definition (II.2.7). For a general cochain $\alpha \in \mathcal{C}^p(\mathcal{K})$, and an n-simplex σ_n with $n > p$, the element $\omega_\alpha|_{\sigma_n}$ is called the *Whitney form of weight α at σ_n*.

Combinatorial computations

In order to facilitate the understanding of the next section, some easy but useful and necessary combinatorial computations to proof proposition (II.2.1) are put together here. Let $\sigma_n \in \mathcal{K}_n$ be an n-simplex and $p, q \in [n]$ such that $p + q = n$. Then,

Proposition (II.2.8). *The number of q-face paths from dimension n to dimension p is given by the formula:*

$$\binom{n+1}{q} = \frac{(n+1)!}{q!(n+1-q)!}. \qquad \text{(Eq. II.2.23)}$$

Proposition (II.2.9). *The number of q-face paths from dimension n to dimension p coincides with the number of unique subsets of barycentric coordinates with cardinality $p+1$ over the set $\{\lambda_0, \ldots, \lambda_n\}$.*

The proof of both propositions is the following:

Proof. By hypothesis, $n = p + q$, and by definition, a q-face path from dimension n to dimension p is determined by a sequence of length q of distinct elements from the set $[n] = \{0, \ldots, n\}$. This set $[n]$ contains exactly $n + 1$ elements. Then, the number of q-face paths is given by the combinatorial number

$$\binom{n+1}{q} = \binom{n+1}{p+1} = \frac{(p+q+1)!}{q!(p+1)!}, \qquad \text{(Eq. II.2.24)}$$

where the expresion in the middle is exactly the number of unique non-empty $p+1$ subsets over the set $\{\lambda_0, \ldots, \lambda_n\}$. □

Proposition (II.2.10). *Consider the face paths $I = (i_1, \ldots, i_q) \in I^n_p$ and $J = (j_1, \ldots, j_p) \in I^n_q$. Then, the following statements are equivalent:*

1) $|I \cap J| = 0$
2) $|I^c \cap J^c| = 1$

Proof. Recall that $[n] = \{0, \ldots, n\}$ and consequently, $|[n]| = n + 1$. On the other hand, $I, J \subseteq [n]$ are subsets of $[n]$ with cardinalities $|I| = q$ and $|J| = p$ respectively. Suppose that $|I \cap J| = 0$. Then, $|I \cup J| = |I| + |J| = n$ and, with basic set operations we deduce that:

$$|I^c \cap J^c| = |(I \cup J)^c| = |[n]| - |I \cup J| = n + 1 - n = 1.$$

□

Proposition (II.2.11). *The number of non-connected face paths I, J from dimensions p and q to dimension n is determined by the formula*

$$\frac{(n+1)!}{p!q!}.$$

Proof. The multinomial coefficient

$$\binom{n}{k_1, \ldots, k_m} = \frac{n!}{k_1! \cdots k_m!}$$

is precisely the number of possible combinations of non intersecting subsets $k_1, \ldots k_m$ such that $\sum k_i = n$. As an n-simplex has $(n + 1)$ barycentric coordinates, for each of them we have this number of face paths. So, multiplying the expression above we obtain

$$(n+1)\binom{n}{p, q} = \frac{(n+1)!}{p!q!}.$$

□

II.2.1. Dupont's formula

Dupont, at [**19**], chose the following words to introduce the formula for the cochain product: *A direct calculation shows that...*, but he did not explicitly show such calculation. The rest of the section is dedicated to prove proposition (II.2.1).

First, consider three simplices $\sigma_p, \sigma_q, \sigma_n$ of dimensions p, q and n respectively, in such a way that, $n = p + q$, $\overline{\partial}_I(\sigma_n) = \sigma_p$ and $\overline{\partial}_J(\sigma_n) = \sigma_q$, for some face paths $\overline{\partial}_I \in I_p^n$ and $\overline{\partial}_J \in I_q^n$. Then,

PROPOSITION (II.2.12). *If $|I \cap J| \neq 0$ then $\omega_I \wedge \omega_J = 0$.*

PROOF. Let ω_I and ω_J be the elements

$$\begin{aligned} \omega_I &= p! \sum_{k=0}^{p} (-1)^k t_{i_k} y_{i_0} \wedge \ldots \wedge \hat{y}_{i_k} \wedge \ldots \wedge y_{i_p}, \\ \omega_J &= q! \sum_{l=0}^{q} (-1)^l t_{j_l} y_{j_0} \wedge \ldots \wedge \hat{y}_{j_l} \wedge \ldots \wedge y_{j_q}, \end{aligned}$$

where the multi-indices $I^c = (i_0, \ldots, i_p)$ and $J^c = (j_0, \ldots, j_q)$ are determined by the inclusions $i_{I^c}(\sigma_p) = \langle \lambda_{i_0}, \ldots, \lambda_{i_p} \rangle$ and $i_{J^c}(\sigma_q) = \langle \lambda_{j_0}, \ldots, \lambda_{j_q} \rangle$ on $\sigma_n = \langle \lambda_0, \ldots, \lambda_n \rangle$. Consider the product $\omega_I \wedge \omega_J$:

$$(p! \sum_{k=0}^{p} (-1)^k t_{i_k} y_{i_0} \wedge \ldots \wedge \hat{y}_{i_k} \wedge \ldots \wedge y_{i_p}) \wedge (q! \sum_{l=0}^{q} (-1)^l t_{j_l} y_{j_0} \wedge \ldots \wedge \hat{y}_{j_l} \wedge \ldots \wedge y_{j_q}). \quad \text{(Eq. II.2.25)}$$

As $|I \cap J| + |(I \cap J)^c| = |I \cap J| + |I^c \cup J^c| = n + 1$, if $|I \cap J| > 0$, then $|I^c \cup J^c| < n + 1$. By hypothesis, $p + q = n$ so

$$|I^c \cup J^c| = |I^c| + |J^c| - |I^c \cap J^c| = (p + 1) + (q + 1) - |I^c \cap J^c| < n + 1$$

and we have that $|I^c \cap J^c| > 1$. This means that, at Eq. II.2.25, all factors share at least one exterior variable y_k, so all products vanish. □

PROPOSITION (II.2.13). *Let I, J face paths as defined above. If $I^c \cap J^c = \{m\}$, then*

$$\omega_I \wedge \omega_J = sgn(J, I) \frac{p!q!}{n!} t_m \omega_n^n$$

where $sgn(J, I)$ is the sign of the permutation taking the concatenation of the tuples $J \frown I$ to $(0, \ldots, \hat{m}, \ldots, n)$.

PROOF. Let $I \in I_p^n$ and $J \in I_q^n$ be a q- and p-face paths respectively, such that $I^c = (i_0, \ldots, i_p)$ and $J^c = (j_0, \ldots, j_q)$ are the corresponding indices of the barycentric coordinates of the faces of σ_n determined by such paths. It is easy to see (repeating the reasoning made in the previous proof) that $|I^c \cap J^c| = 1$ implies $|I \cap J| = 0$. Assume that $I^c \cap J^c = i_s = j_r = m$. In that case, notice that the face paths I, J are precisely those corresponding with the sequences

$$I = J^c \setminus \{j_r\} = \{j_0, \ldots, \hat{j}_r, \ldots, j_q\},$$

and

$$J = I^c \setminus \{i_s\} = \{i_0, \ldots, \hat{i}_s, \ldots, i_p\}.$$

This will be useful later. Then, consider the Whitney forms

$$\begin{aligned}\omega_I &= p!\sum_{k=0}^{p}(-1)^k t_{i_k} y_{i_0}\wedge\ldots\wedge\hat{y}_{i_k}\wedge\ldots\wedge y_{i_p},\\ \omega_J &= q!\sum_{l=0}^{q}(-1)^l t_{j_l} y_{j_0}\wedge\ldots\wedge\hat{y}_{j_l}\wedge\ldots\wedge y_{j_q}.\end{aligned}$$

Looking at their product $\omega=\omega_I\wedge\omega_J$:

$$(p!\sum_{k=0}^{p}(-1)^k t_{i_k} y_{i_0}\wedge\ldots\hat{y}_{i_k}\ldots\wedge y_{i_p})\wedge(q!\sum_{l=0}^{q}(-1)^l t_{j_l} y_{j_0}\wedge\ldots\hat{y}_{j_l}\ldots\wedge y_{j_q}),\qquad\text{(Eq. II.2.26)}$$

it is easy to see that most of the terms will vanish due to both expressions share the exterior variable $y_m=y_{i_s}=y_{j_r}$. This expression can be simplified in such a way that

$$\begin{aligned}\omega =& p!q!\left[((-1)^s t_m y_{i_0}\wedge\ldots\hat{y}_{i_s}\ldots\wedge y_{i_p})(\sum_{l=0}^{q}(-1)^l t_{j_l} y_{j_0}\wedge\ldots\hat{y}_{j_l}\ldots\wedge y_{j_q})\right]+\\ &+p!q!\left[(\sum_{\substack{k=0\\k\neq s}}^{p}(-1)^k t_{i_k} y_{i_0}\wedge\ldots\hat{y}_{i_k}\ldots\wedge y_{i_p})((-1)^r t_m y_{j_0}\wedge\ldots\hat{y}_{j_r}\ldots\wedge y_{j_q})\right]\end{aligned}$$

We can take the factor t_m outside both summands and simplify a little bit the expression such that:

$$\begin{aligned}\omega =& p!q!t_m\sum_{j_l\in J^c}(-1)^{l+s} t_{j_l} y_{i_0}\wedge\ldots\hat{y}_{i_s}\ldots\wedge y_{i_p}\wedge y_{j_0}\wedge\ldots\hat{y}_{j_l}\ldots\wedge y_{j_q}+\\ &+p!q!t_m\sum_{\substack{i_k\in I^c\\i_k\neq i_s}}(-1)^{k+r} t_{i_k} y_{i_0}\wedge\ldots\hat{y}_{i_k}\ldots\wedge y_{i_p}\wedge y_{j_0}\wedge\ldots\hat{y}_{j_r}\ldots\wedge y_{j_q}\end{aligned}$$

Then, if we analyze both summand separately, due to the relation $\sum y_k=0$, it is easy to see that both summands can be expressed in such a way that the variable $y_m=y_{i_s}=y_{j_r}$ does not appear in any term, just adjusting the corresponding sign. So, applying this relation we obtain the following expression:

$$\begin{aligned}\omega =& p!q!t_m\sum_{j_l\in J^c}(-1)^{s+r} t_{j_l} y_{i_0}\wedge\ldots\hat{y}_{i_s}\ldots\wedge y_{i_p}\wedge y_{j_0}\wedge\ldots\hat{y}_{j_r}\ldots\wedge y_{j_q}+\\ &+p!q!t_m\sum_{\substack{i_k\in I^c\\i_k\neq i_s}}(-1)^{s+r} t_{i_k} y_{i_0}\wedge\ldots\hat{y}_{i_s}\ldots\wedge y_{i_p}\wedge y_{j_0}\wedge\ldots\hat{y}_{j_r}\ldots\wedge y_{j_q}\end{aligned}$$

The sign $(-1)^{s+r}$ does not depend on the index set of the summatory, and also, notice that $s+r=m$, that is, the global index of the variable y_m (as I^c and J^c are sequences ordered in strictly increasing order, the sum of the relative locations of i_s and j_r give us the number of elements in the set $\{0,\ldots,n\}$ lower than m. Then, getting the sign outside the summatory and simplifying the expression we obtain that:

$$\omega = p!q!(-1)^m t_m\sum_{k=0}^{n} t_k y_{i_0}\wedge\ldots\hat{y}_{i_s}\ldots\wedge y_{i_p}\wedge y_{j_0}\wedge\ldots\hat{y}_{j_r}\ldots\wedge y_{j_q}$$

Now, taking a look at the indices of the exterior variables, we see that the sequence

$$(i_0,\ldots,\hat{i}_s,\ldots,i_p,j_0,\ldots,\hat{j}_r,\ldots,j_q)$$

is precisely the concatenation of the tuples $J \frown I$. Defining the sign $sgn(J, I)$ as the permutation sign taking the sequence $J \frown I$ to $(0, \dots, \hat{m}, \dots, n)$, and reordering the terms we obtain the expression:

$$\omega = p!q!(-1)^m sgn(J, I) t_m \sum_{k=0}^{n} t_k y_0 \wedge \dots \hat{y}_m \dots \wedge y_n.$$

Finally, using the sign $(-1)^m$, the relation $\sum t_i = 1$ and the definition of Whitney forms we arrive to the following result:

$$\omega_I \wedge \omega_J = p!q! sgn(J, I) \cdot t_m \cdot y_1 \wedge \dots \wedge y_n. = \frac{p!q!}{n!} sgn(J, I) \cdot t_m \cdot \omega_n^n. \qquad \text{(Eq. II.2.27)}$$

□

Remark **(II.2.14).** With patience, we can see that the sign defined by Dupont, i.e. $(-1)^{p-s+r}$ times the sign of the permutation

$$(0, \dots, n) \frown (i_0, \dots, \hat{i}_s, \dots, i_p, i_s = j_r, j_0, \dots, \hat{j}_r, \dots, j_q),$$

is equivalent to the sign $sgn(J, I)$ defined as the permutation sign from taking the sequence $J \frown I$ to $(0, \dots, \hat{m}, \dots, n)$.

At this point, knowing that $\omega_I \wedge \omega_J = sgn(J, I) \frac{p!q!}{n!} t_m \omega_n^n$, it is easy to see that:

Proposition (II.2.15). *For two face paths $I = I_p^n$ and $J = I_q^n$ such that $I \cap J = \emptyset$, the integral of the product $\omega_I \wedge \omega_J$ is defined by the formula*

$$\oint (\omega_I \wedge \omega_J) = sgn(J, I) \frac{p!q!}{(n+1)!}.$$

Proof. It is a direct consequence of proposition (II.2.13). □

Finally, we arrive to the proof of proposition (II.2.1):

Proof of Dupont's formula. Consider the Whitney forms ω_α and ω_β of weights $\alpha \in \mathcal{C}^p(\mathcal{K})$, $\beta \in \mathcal{C}^q(\mathcal{K})$. Using propositions (II.2.13), (II.2.15) and (II.2.11) we deduce that, at some n-simplex σ_n, the cochain product $\alpha \sqcup\!\sqcup \beta$ is defined as

$$\begin{aligned}
(\alpha \sqcup\!\sqcup \beta)(\sigma_n) &= \oint (\omega_\alpha|_{\sigma_n} \wedge \omega_\beta|_{\sigma_n}) = \\
&= \sum_{I,J} \alpha(\overline{\partial}_I \sigma_n) \cdot \beta(\overline{\partial}_J \sigma_n) \cdot \oint (\omega_I \wedge \omega_J) = \\
&= \sum_{I,J} \alpha(\overline{\partial}_I \sigma_n) \cdot \beta(\overline{\partial}_J \sigma_n) \cdot \frac{p!q!}{(n+1)!} sgn(J, I) = \\
&= \frac{p!q!}{(n+1)!} \sum_{I,J} sgn(J, I) \alpha(\overline{\partial}_I \sigma_n) \cdot \beta(\overline{\partial}_J \sigma_n).
\end{aligned}$$

□

A visual representation of the product of two unit Whitney forms is shown in Figure II.3. The non-connected paths conditions is visualized as their product do not vanish only when the multiplication takes place between a 2-simplex with the 1-simplices inside the horn of the opposite vertex.

Just to finish this section, Dupont's formula is defined for the product of two cochains of arbitrary degrees p and q, at a $(p+q)$-simplex. This formula captures the combinatorial structure of the wedge product at $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, but not the underlying affine structure induced by the simplicial construction. As we have seen, for two differential forms ω_1, ω_2, the cochain $\oint(\omega_1 \wedge \omega_2)$ corresponds to $(\oint \omega_1) \sqcup\!\sqcup (\oint \omega_2)$. However, some information is lost when the integration map is applied. The product $\omega_1 \wedge \omega_2$ carries extra information reflected at each polynomial differential form $\theta \in (\mathcal{A}_{\mathcal{PL}})_n$ assigned to each simplex. In order to explain what information is lost by the integral, in the next section it is described explicitely the generalization of Dupont's formula for the case of the product of three cochains. As far as we know, this is the first time such formula is presented.

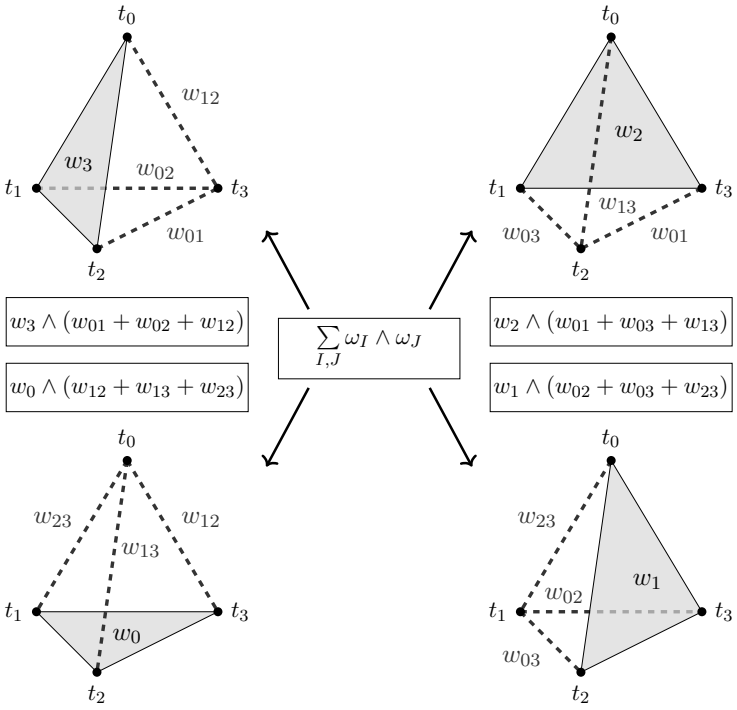


FIGURE II.3. Representation of the product of the unit Whitney products of degrees 1 and 2 on the standard Δ^3 simplex.

II.2.2. Generalization of Dupont's formula for the case of three factors

In this section, we are going to generalize Dupont's formula for the case of the product of three cochains $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{C}(\mathcal{K})$. However, as we will see, the generalization of such formula is not trivial. In general, given three differential forms $\omega_1, \omega_2, \omega_3 \in \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, the products $(\omega_1 \wedge \omega_2 \wedge \omega_3)$ and $((\mathcal{W} \circ \oint)(\omega_1 \wedge \omega_2) \wedge \omega_3)$ are not equal because the projection onto $\mathcal{C}(\mathcal{K})$ removes a key piece of information. In the same way, a direct calculation shows that the map defined by Dupont's formula:

$$(\alpha_1 \sqcup\!\sqcup (\alpha_2 \sqcup\!\sqcup \alpha_3)) \neq ((\alpha_1 \sqcup\!\sqcup \alpha_2) \sqcup\!\sqcup \alpha_3).$$

is not associative. In other words, in general

$$\alpha_1 \sqcup\!\sqcup (\alpha_2 \sqcup\!\sqcup \alpha_3) \neq \oint (\mathcal{W}(\alpha_1) \wedge \mathcal{W}(\alpha_2) \wedge \mathcal{W}(\alpha_3)).$$

As we are going to see, when we apply Dupont's formula $(\oint \omega_1) \sqcup (\oint \omega_2)$ we lose precisely that information related to the product of face paths. For that reason, in order to find an expression that generalizes such formula, it is necessary to define the multilinear map:

$$\sqcup_n : \mathcal{C}^*(\mathcal{K}) \times \overset{n}{\cdots} \times \mathcal{C}^*(\mathcal{K}) \longrightarrow \mathcal{C}^*(\mathcal{K}).$$

For the rest of the section, we are going to limit the description of the map $\sqcup_n$ to the case of the product of three cochains $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{C}(\mathcal{K})$ of arbitrary degrees:

$$\sqcup_3 : \mathcal{C}^*(\mathcal{K}) \times \mathcal{C}^*(\mathcal{K}) \times \mathcal{C}^*(\mathcal{K}) \longrightarrow \mathcal{C}^*(\mathcal{K}),$$

such that

$$\sqcup_3(\alpha_1, \alpha_2, \alpha_3) = \oint (\mathcal{W}(\alpha_1) \wedge \mathcal{W}(\alpha_2) \wedge \mathcal{W}(\alpha_3)).$$

First, consider four simplices $\sigma_p, \sigma_q, \sigma_r, \sigma_n \in \mathcal{K}$ of dimensions p, q, r and n respectively, in such a way that $n = p + q + r$, and assume that

$$\overline{\partial}_I(\sigma_n) = \sigma_p, \quad \overline{\partial}_J(\sigma_n) = \sigma_q, \quad \overline{\partial}_K(\sigma_n) = \sigma_r,$$

for some face paths $\overline{\partial}_I \in I^n_p$, $\overline{\partial}_J \in I^n_q$ and $\overline{\partial}_K \in I^n_r$. Also, let

$$\begin{aligned} I^c &= (i_0, \ldots, i_p), \\ J^c &= (j_0, \ldots, j_q), \\ K^c &= (k_0, \ldots, k_r), \end{aligned}$$

be the subsets of indices, respect to the set $[n] = \{0, \ldots, n\}$ representing the barycentric coordinates of each face on σ_n. On the proof of Dupont's formula we saw that, given two face paths I, J from dimensions p and q to dimension $p + q$, the product $\omega_I \wedge \omega_J$ is not zero if and only if $|I \cap J| = 0$ or, equivalently, $|I^c \cap J^c| = 1$. However, this conditions change when more than two factors are considered. Let

$$\omega_I \wedge \omega_J \wedge \omega_K$$

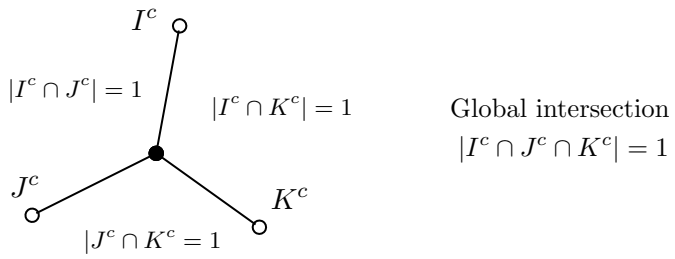
be the product of the propagation of three unit Whitney forms at some simplex $\sigma_n \in \mathcal{K}_n$, according to the face paths I, J and K defined above. In that case, knowing that

$$|I^c \cup J^c \cup K^c| = |I^c| + |J^c| + |K^c| - |I^c \cap J^c| - |I^c \cap K^c| - |J^c \cap K^c| + |I^c \cap J^c \cap K^c| = n + 1,$$

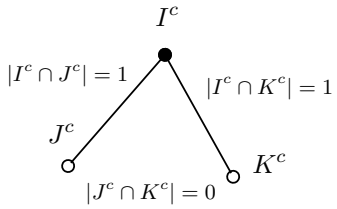
we deduce the following equation that must be satisfied:

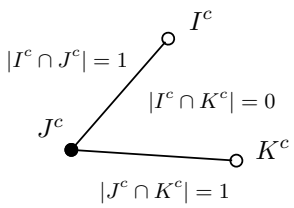
$$|I^c \cap J^c| + |I^c \cap K^c| + |J^c \cap K^c| - |I^c \cap J^c \cap K^c| = 2.$$

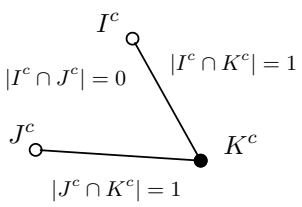
Looking at all possible configurations, we distinguish two scenarios: (1) when the simplices σ_p, σ_q and σ_r share a common barycentric coordinate (star configuration):



and (2) when one simplex acts as a bridge between the other two (non-intersecting) simplices. In that case, the global intersection $|I^c \cap J^c \cap K^c| = 0$ and there are three possible configurations:







As we see, the conditions

$$\begin{aligned} |I^c \cap J^c| &= 1, \\ |I^c \cap K^c| &= 1, \\ |J^c \cap K^c| &= 1, \end{aligned}$$

only hold in one particular configuration out of four, so other products must be taken into account.

Equivalently, we can deduce which relations will have the face paths I, J, K depending on the configuration of the simplices σ_p, σ_q and σ_r. The condition $|I \cap J \cap K| = 0$ holds in any case, due to the fact that $|I^c \cup J^c \cup K^c| = n + 1$. However, knowing that

$$|I \cup J \cup K| = n + 1 - |I^c \cap J^c \cap K^c|$$

and

$$|I \cup J \cup K| = |I| + |J| + |K| - |I \cap J| - |I \cap K| - |J \cap K| + |I \cap J \cap K|,$$

the pairwise intersections vary depending on the value of $|I^c \cap J^c \cap K^c|$.
In the case $|I^c \cap J^c \cap K^c| = 1$, we see that

$$|I \cup J \cup K| = |I| + |J| + |K| - |I \cap J| - |I \cap K| - |J \cap K| + |I \cap J \cap K| = n,$$

and substituting the known values we get the relation

$$|I \cap J| + |I \cap K| + |J \cap K| = n.$$

Then, we deduce that

$$\begin{aligned} |I \cap J| &= r, \\ |I \cap K| &= q, \\ |J \cap K| &= p. \end{aligned}$$

But, for the other configurations, the cardinalities of each intersection have to satisfy the equation

$$|I \cup J \cup K| = |I| + |J| + |K| - |I \cap J| - |I \cap K| - |J \cap K| + |I \cap J \cap K| = n - 1,$$

or, equivalently

$$|I \cap J| + |I \cap K| + |J \cap K| = n - 1.$$

As each scenario depends on which pair of simplices do not intersect, after analyzing all cases we get the following relations between the face paths that form the triple:

Case $|J^c \cap K^c| = 0$:

$$\begin{aligned} |I \cap J| &= r, \\ |I \cap K| &= q, \\ |J \cap K| &= p - 1. \end{aligned}$$

Case $|I^c \cap K^c| = 0$:

$$\begin{aligned} |I \cap J| &= r, \\ |I \cap K| &= q-1, \\ |J \cap K| &= p. \end{aligned}$$

Case $|I^c \cap J^c| = 0$:

$$\begin{aligned} |I \cap J| &= r-1, \\ |I \cap K| &= q, \\ |J \cap K| &= p. \end{aligned}$$

Then, in an attempt to find an explicit expression for the product $\sqcup_3$, this characterization describes which triples of face paths (I, J, K) have to be taken into account. Analyzing the combinatorial structure of these restrictions, we can count the total number of non-zero products depending on each configuration. In the case that $|I^c \cap J^c \cap K^c| = 1$, we can see that there are

$$(n+1)\binom{n}{p,q,r}$$

triples of face paths satisfying that condition. From the other three configurations, we can see that there are

$$(3-1)!\binom{n+1}{2}\left[\tbinom{n-1}{p-1,q,r} + \tbinom{n-1}{p,q-1,r} + \tbinom{n-1}{p,q,r-1}\right]$$

different triples of paths, where the factor $\binom{n+1}{2}$ comes from selecting two different pivots, one for each intersection of simplices, and the factor $(3-1)!$ refers to the permutation of such pivots. Summing up, the total number of triples corresponding to non-vanishing products of face paths are

$$\#(paths) = (n+1)\binom{n}{p,q,r} + (3-1)!\binom{n+1}{2}\left[\tbinom{n-1}{p-1,q,r} + \tbinom{n-1}{p,q-1,r} + \tbinom{n-1}{p,q,r-1}\right]. \quad \text{(Eq. II.2.28)}$$

Following the same reasoning that the one used to prove Dupont's formula, we deduce that, given an arbitrary sequence of three cochains $\alpha_1, \alpha_2, \alpha_3$ of degrees p, q, r respectively, where $n := p+q+r$, and $f = 3$ represents the number of factors, the cochain product

$$\sqcup_3(\alpha_1, \alpha_2, \alpha_3)(\sigma_n)$$

is determined by the formula:

$$\frac{p!q!r!}{(n+f-1)!} \sum_{I,J,K} \Upsilon(I,J,K) \cdot \alpha_1(\overline{\partial}_I \sigma_n) \cdot \alpha_2(\overline{\partial}_J \sigma_n) \cdot \alpha_3(\overline{\partial}_K \sigma_n), \quad \text{(Eq. II.2.29)}$$

where $\sigma_n \in \mathcal{K}_n$; (I, J, K) is a multi-face path satisfying one of the four configurations described above, and $\Upsilon(I, J, K)$ is the generalization of Dupont's signature $sgn(J, I)$ for a sequence of three paths that is computed as follows:

Case 1: $|I^c \cap J^c \cap K^c| = 1$:

$$\Upsilon(I,J,K) = 2\upsilon$$

where υ is the permutation sign of the triple concatenation $(J \cap K) \frown (I \cap K) \frown (I \cap J)$ necessary to order the elements in a strictly increasing sequence, and the coefficient 2 comes from the fact that:

$$\omega_I \wedge \omega_J \wedge \omega_K = \frac{p!q!r!}{n!} t_i^2 \omega_n^n$$

where t_i represents the variable associated to the pivoting coordinate, and

$$\int (\frac{p!q!r!}{n!} t_i^2 \omega_n^n) = \frac{2!p!q!r!}{(n+2)!}.$$

Case 2: $|I^c \cap J^c \cap K^c| = 0$ **and** ($|J^c \cap K^c| = 0$ **or** $|I^c \cap K^c| = 0$):
Defining the tuples

$$\begin{aligned} \vartheta_1 &= J \setminus \{I \cap J\}, \\ \vartheta_2 &= I \setminus \{I \cap J\}, \\ \vartheta_3 &= I \cap J. \end{aligned}$$

and being $sgn(a)$ the permutation sign of ordering the tuple a in a strictly increasing sequence, then

$$\Upsilon(I, J, K) = sgn(\vartheta_1 \frown \vartheta_2) \cdot sgn(K \frown \vartheta_3).$$

Case 3: $|I^c \cap J^c \cap K^c| = 0$ **and** $|I^c \cap J^c| = 0$:
Defining the tuples

$$\begin{aligned} \vartheta_1 &= K \setminus \{J \cap K\}, \\ \vartheta_2 &= J \setminus \{J \cap K\}, \\ \vartheta_3 &= J \cap K. \end{aligned}$$

and again, being $sgn(a)$ the permutation sign of ordering the tuple a in a strictly increasing sequence, then

$$\Upsilon(I, J, K) = sgn(\vartheta_1 \frown \vartheta_2) \cdot sgn(\vartheta_3 \frown I).$$

Notice that, in cases 2 and 3, $|\Upsilon| = 1$. This is due to the fact that, in both cases, there are two pivots, one for each non-empty pairwise intersection and it is easy to deduce that

$$\omega_I \wedge \omega_J \wedge \omega_K = \frac{p!q!r!}{n!} t_i t_j \omega_n^n,$$

where t_i and t_j represent the variables associated to such pivots. Computing the integral of the product we see that

$$\int (\frac{p!q!r!}{n!} t_i t_j \omega_n^n) = \frac{1!1!p!q!r!}{(n+1+1)!}.$$

This information is precisely what is lost when we apply the integration map and the reason why this product at the cochain level is not associative:

$$\alpha_1 \sqcup\!\sqcup (\alpha_2 \sqcup\!\sqcup \alpha_3) \neq \oint (\mathcal{W}(\alpha_1) \wedge \mathcal{W}(\alpha_2) \wedge \mathcal{W}(\alpha_3)).$$

As far as we know, the problem of finding a general closed formula for the product of an arbitrary number of cochains is still open ([**63**], [**1, 50, 58, 49**]). Although this problem was not the aim of this thesis, in our attempt to find the most effective solution for our computations, we came across with this beautiful and stimulating scenario. A deeper search on the literature must be done in order to see the progress in this direction, but we want to state the following conjectures. The first one is related to find a closed formula for the product of an arbitrary number of factors: given an arbitrary sequence of cochains $\alpha_1, \ldots, \alpha_f$ of degrees $k_1, \ldots, k_f$ respectively, in such a way that $\sum k_i = n$,

Conjecture (II.2.16). The cochain product

$$\sqcup_f(\alpha_1, \cdots, \alpha_f)(\sigma_n)$$

is determined by the formula:

$$\frac{k_1! \cdots k_f!}{(n+f-1)!} \sum_{I_1,\ldots,I_f} \Upsilon(I_1, \ldots, I_f) \cdot \alpha_1(\overline{\partial}_{I_1}(\sigma_n)) \cdots \alpha_f(\overline{\partial}_{I_f}(\sigma_n)),$$

where the magnitude $|\Upsilon(I_1, \ldots, I_f)| = m_1! \cdots m_g!$ is determined by the tuple $(m_1, \ldots, m_g)$, for $1 \leq g \leq f-1$, with m_i being the incidence number minus 1 of the pivoting vertex v_i such that $\sum m_i = f-1$, and the sign of $\Upsilon(I_1, \ldots, I_f)$ is determined by the orientation of the simplex that $\overline{\partial}_{I_1}(\sigma_n), \ldots, \overline{\partial}_{I_f}(\sigma_n)$ expand.

The second conjecture is the following: at ([**63**], [**1, 50, 58, 49**]) it is shown that the wedge product transfers a non-associative algebraic structure to the simplicial cochain complex $\mathcal{C}^*(\mathcal{K})$.

Conjecture (II.2.17). There exists a chain homotopy $\phi : \mathcal{C}^*(\mathcal{K}) \to \mathcal{C}^{*-1}(\mathcal{K})$ in such a way that the products $\sqcup = \{\sqcup_n\}_{n\geq 0}$ together with the homotopy ϕ completely describe the transferred algebraic structure mentioned before.

We insist in the preliminary condition of that statement because it is something that appeared naturally in our computations at the end of the thesis, but maybe it is something known in other fields of expertise. A further search on the literature maybe will answer the question. In any case, we will see in section III.4 that it is very easy to implement the original Dupont's formula in a Computer Algebra Software. If it is true that it is possible to generalize such formula, then it would open the door to see if the method could work directly at the cochain level and if the computational complexity would decrease substantially. Also, if this generalization is not known, this formula could bring to us new insights to explore the intrinsic relation between geometry, topology, algebra and combinatorics.

Just to finish, Dupont do not just give us this explicit formula, but also gave us a contraction $s : \mathcal{A}^*_{\mathcal{PL}(\mathcal{K})} \longrightarrow \mathcal{A}^{*-1}_{\mathcal{PL}}(\mathcal{K})$ such that

$$\mathrm{id}_A - \mathcal{W} \circ \oint = d \circ s + s \circ d.$$

In the next and last section of the chapter is described that contraction as a key tool for the effective computation of the Sullivan minimal model of a simplicial set.

Section § II.3
Dupont's contraction

Based on Whitney's work, Dupont [**19**] found an explicit simplicial cochain contraction of the algebra of polynomial differential forms $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ to the cochain complex $\mathcal{C}(\mathcal{K})$:

$$id_{\mathcal{A}} - \mathcal{W} \circ \oint = s \circ d_{\mathcal{A}} + d_{\mathcal{A}} \circ s, \qquad \text{(Eq. II.3.30)}$$

defined by some operator s:

$$s : \mathcal{A}^*_{\mathcal{PL}}(\mathcal{K}) \to \mathcal{A}^{*-1}_{\mathcal{PL}}(\mathcal{K}). \qquad \text{(Eq. II.3.31)}$$

More recent works proved that, using that contraction, an algebraic structure is transfered from $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ to $\mathcal{C}(\mathcal{K})$. However, the induced algebraic structure on $\mathcal{C}(\mathcal{K})$ is not a $CDGA$ [**46, 58, 13**]. In [**13**], the authors gave a concrete specification for the construction of the Dupont contraction

Eq. II.3.30. This description is the following: for each vertex e_i of the n-simplex Δ^n, define the dilation map

$$\phi_i : [0,1] \times \Delta^n \to \Delta^n$$

by the formula

$$\phi_i(u, t_0, \ldots, t_n) = ((1-u)t_0, \ldots, (1-u)t_i + u, \ldots, (1-u)t_n).$$

A visual representation of the dilation maps applied for the standard 2-simplex Δ^2 is represented in Figure II.4. Now, let

$$(\pi_i)_* : \mathcal{A}^*_{\mathcal{PL}}([0,1] \times \Delta^n) \to \mathcal{A}^{*-1}_{\mathcal{PL}}(\Delta^n)$$

be integration over the first factor. Now, let $\omega \in (\mathcal{A}_{\mathcal{PL}})_n$ and define the operator

$$h^i_n : \mathcal{A}^*_{\mathcal{PL}}(\Delta^n) \to \mathcal{A}^{*-1}_{\mathcal{PL}}(\Delta^n)$$

by the formula

$$h^i_n \omega = (\pi_i)_* \phi_i^* \omega,$$

where $\phi_i^*\omega$ is the pullback of ω respect to ϕ_i. Let

$$\epsilon^i_n : \mathcal{A}^*_{\mathcal{PL}}(\Delta^n) \to \mathbb{K}$$

be evaluation at the vertex e_i. The Poincaré Lemma states that

$$\mathrm{id}_{\mathcal{A}} - \epsilon^i_n = d \circ h^i_n + h^i_n \circ d.$$

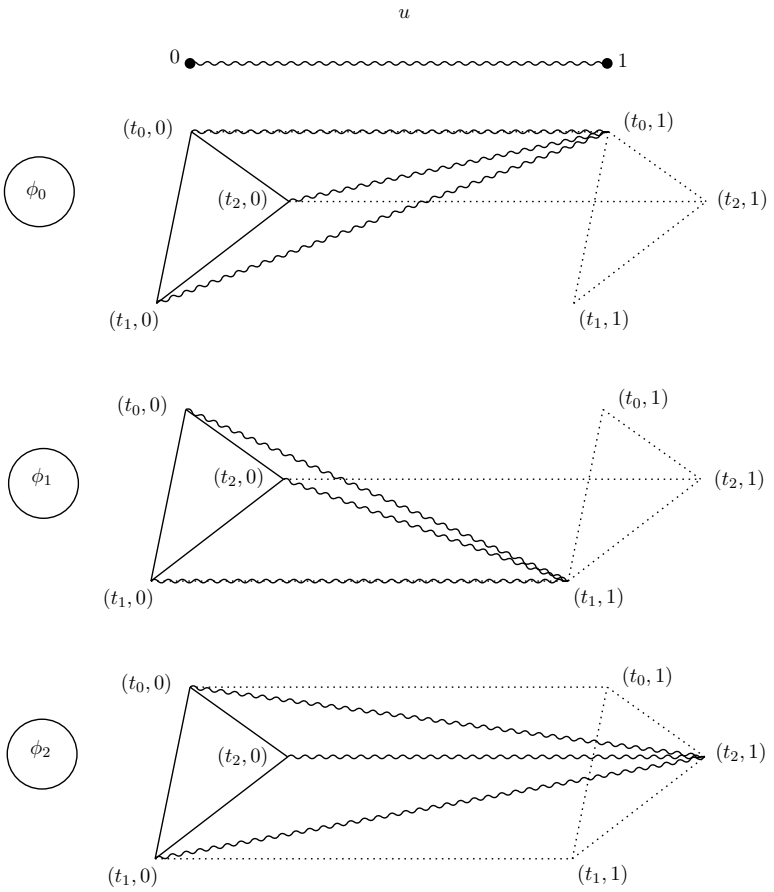


FIGURE II.4. The dilation maps ϕ_i applied to Δ^2.

Now, let $J^k := \{\{i_0, \ldots, i_{k-1}\} \mid 0 \leq i_0 < \cdots < i_{k-1} \leq n\}$, be the set of all subsets of $\{0, \ldots, n\}$ with cardinality k.

THEOREM (II.3.1) (Dupont 76' ,[**19**]). *The operator*

$$s_n(\omega) = \sum_{k=0}^{n-1} \sum_{I \in J^k} \mathcal{W}_n^I(1) \wedge (h^{i_k} \circ \cdots \circ h^{i_0})(\omega)$$

defines a simplicial contraction from $\mathcal{A}^*_{\mathcal{PL}}(\mathcal{K})$ *to* $\mathcal{C}^*(\mathcal{K})$*:*

$$\mathrm{id}_{\mathcal{A}} - \mathcal{W} \circ \oint = d \circ s + s \circ d.$$

THEOREM (II.3.2) (Getzler 09', [**31**]). *The side conditions*

$$\begin{aligned} \oint \circ s &= 0 \\ s \circ s &= 0 \end{aligned}$$

hold.

This explicit description is the core of the solution we found for the most difficult problem we had trying to make effective our adaptation of the **MMarco**'s implementation. It shows that it is possible to effectively compute that contraction, in the same way we implement other $CDGA$ morphisms between $(\mathcal{A}_{\mathcal{PL}})$.

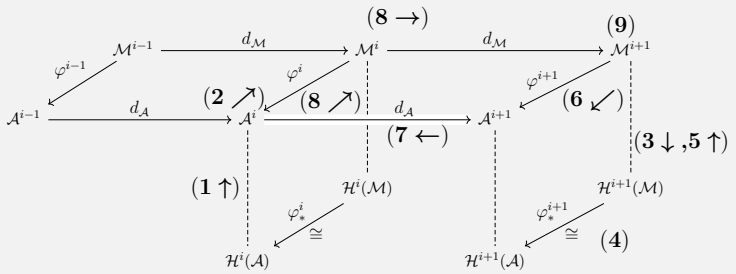
In the next chapter, the main contribution of this thesis is presented.

III

The minimal model of a finite simplicial set

In this chapter, the main contribution of this thesis is presented, i.e., the effective computation of the minimal model of a finite simplicial set $\mathcal{K}$. This model is defined as the minimal model of the algebra of polynomial differential forms $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. Our approach is an adaptation of **MMarco**'s method for the computation of the s-minimal model of a finitely generated connected $CDGA$ (see section I.1.2). The key aspects of the original method (at some degree i) can be summarized in the following diagram:

Key steps of MMarco's implementation



(1) Lifts generators $\mathcal{H}^i(\mathcal{A}) \to \mathcal{A}^i$.
(2) Updates $(\mathcal{M}^i, \varphi^i)$ in order to make φ_*^i surjective.
(3) Gets a basis of $\mathcal{H}^{i+1}(\mathcal{M})$.
(4) Computes the kernel of the induced morphism φ_*^{i+1}.
(5) Lifts the generators of $\ker(\varphi_*^{i+1})$.
(6) For each generator $z \in \ker(\varphi_*^{i+1})$, gets its image $\varphi(z) \in \mathcal{A}^{i+1}$.
(7) For each $\varphi(z) \in \mathcal{A}^{i+1}$, finds $\theta \in \mathcal{A}^i$ such that $d_{\mathcal{A}}(\theta) = \varphi(z)$.
(8) Updates $(\mathcal{M}^i, \varphi^i)$ in order to make φ_*^i injective.
(9) $(\mathcal{M}, \varphi)$ is now an i-minimal model of $\mathcal{A}$.

Arrow notation: having the needed elements → compute/modify/link to.

After analyzing carefully the flow diagram of this algorithm, we identified specifically the need to solve these problems in order to adapt the method for the non-connected $CDGA$ $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$:

(P1) Effective representation of the $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ elements.
(P2) Difficulty to compute a basis of $\mathcal{H}^*(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$ ⇒ **[Steps 1, 3, 4]**

(P3) Given an exact element $\varphi(z) \in \mathcal{A}^{i+1}_{\mathcal{PL}}(\mathcal{K})$, a lack of a general and effective method to compute a primivite $\theta \in \mathcal{A}^{i}_{\mathcal{PL}}(\mathcal{K})$, i.e., a differential form such that $d_{\mathcal{A}}(\theta) = \varphi(z) \Rightarrow$ **[Step 7]**.

The source of these problems lies in the fact that linear algebra computations involve maps between vector spaces and $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, as a graded vector space, is not finitely generated. In the next sections, our approach for the solution of these problems is described.

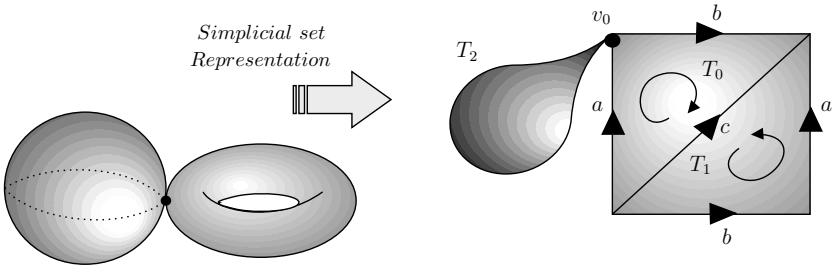


FIGURE III.1. Representation of the space $\mathbb{S}^2 \vee \mathbb{T}$ as a simplicial set.

In order to ilustrate how the new method works, we are going to consider the space $\mathcal{K} = \mathbb{S}^2 \vee \mathbb{T}$, the wedge of a 2-sphere with a torus $\mathbb{S}^1 \times \mathbb{S}^1$ (see Figure III.1). Despite its simplicity, this example allows us to introduce and describe the underlying computational steps carried out by the algorithm. Let $\mathcal{A}$ be a *CDGA*. Recall from (I.1.7), that a minimal model of a $\mathcal{A}$ consist of a pair $(\mathcal{M}, \varphi)$ where $\mathcal{M}$ is a Sullivan minimal algebra, and $\varphi : \mathcal{M} \to \mathcal{A}$ is a quasi-isomorphism of *CDGAs*. If $\mathcal{K}$ is a simplicial structure (e.g., a simplicial set), the minimal model of $\mathcal{K}$ is defined as the minimal model of the associated algebra of polynomial differential forms $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$.

SECTION § III.1
Effective representation of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ elements

As we mentioned at the beginning, consider the simplicial set $\mathcal{K} = \mathbb{S}^2 \vee \mathbb{T}$ (see Figure III.1). This object can be described as the collection of sets:

$$\begin{aligned}\mathcal{K}_0 =& \{v_0\}\\ \mathcal{K}_1 =& \{a, b, c, s_0(v_0)\}\\ \mathcal{K}_2 =& \{T_0, T_1, T_2, s_0(a), s_0(b), s_0(c), s_1(a), s_1(b), s_1(c), s_1(s_0(v_0))\}\end{aligned}$$

and, for $n > 2$, $\mathcal{K}_n$ contains only the corresponding degenerated simplices (for the definition of simplicial sets see section I.2.5). As it is known that degenerated simplices does not carry cohomological information [**21, 27**], the first decision of the algorithm is to ignore those simplices. Then, our simplicial set is determined by the three sets:

$$\begin{aligned}\mathcal{K}_0 =& \{v_0\},\\ \mathcal{K}_1 =& \{a, b, c\},\\ \mathcal{K}_2 =& \{T_0, T_1, T_2\}.\end{aligned}$$

The face maps of this object are determined by the set maps:

$$\partial_0^1 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}, \quad \partial_1^1 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

$$\partial_0^2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \partial_1^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \quad \partial_2^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The method begins with the computation of the simplicial cohomology of $\mathcal{K}$. This is done as follows: first, consider the simplicial cochain complex

$$\mathcal{C}^0(\mathcal{K}) \xrightarrow{d_{\mathcal{C}}^0} \mathcal{C}^1(\mathcal{K}) \xrightarrow{d_{\mathcal{C}}^1} \mathcal{C}^2(\mathcal{K}) \xrightarrow{d_{\mathcal{C}}^2} 0.$$

Recall from section I.2.6, that $\mathcal{C}_*(\mathcal{K})$ (respectively $\mathcal{C}^*(\mathcal{K})$) denotes the (co)chain complex $\mathcal{C}_*(\mathcal{NK})$ (respectively $\mathcal{C}^*(\mathcal{NK})$) defined over the subsimplicial set $\mathcal{NK} := \{\mathcal{NK}_n\}_{n\geq 0}$, where

$$\mathcal{NK}_n := \{\sigma \in \mathcal{K}_n |\ \sigma \text{ is a non-degenerated simplex}\}.$$

The differentials are defined as the linear maps $d_{\mathcal{C}}^n = \sum_{i=0}^{n+1} (-1)^i (\partial_i^{n+1})^T$, so, applying the formula, we get

$$d_{\mathcal{C}}^0 = (0,0,0)^T, \quad d_{\mathcal{C}}^1 = \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix},$$

and $d_{\mathcal{C}}^n = 0$ for $n \geq 2$.

Once the simplicial cochain complex is obtained, we can compute its cohomology in order to find the first non-trivial cohomology group. The method works for connected simplicial sets (i.e., with $\mathcal{H}^0 = \mathbb{Q}$). In the case of non-connected simplicial sets, the method works applying it separately for each connected component. As in our example, $\mathcal{K}$ is connected, we start computing the cohomology for degrees $n \geq 1$. For doing so, we start computing an AT-model and the discrete hodge decomposition of the simplicial cochain complex as it is explained at section I.2.6. At dimension 1, we get the following decomposition:

$$\mathcal{C}^1(\mathcal{K}) = H_1 \oplus X_1 \oplus Y_1$$

where

$$\begin{aligned} H_1 =& \mathbb{Q}\langle (1,0,1), (0,1,1) \rangle, \\ X_1 =& 0, \\ Y_1 =& \mathbb{Q}\langle (1,0,0) \rangle. \end{aligned}$$

By definition, $H_1 \cong \mathcal{H}^1(\mathcal{C}(\mathcal{K})) \cong \mathcal{H}^1(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$, and each element on the basis of this subspace is precisely a representative of a unique cohomology class on $\mathcal{C}^1(\mathcal{K})$. As we see, the subspace H_1 has dimension 2 so we proceed computing the corresponding $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ elements. Recall that an element of $\Phi \in \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ is a collection of simplicial set maps

$$\Phi = \{\Phi_n : \mathcal{K}_n \to (\mathcal{A}_{\mathcal{PL}})_n\}_{n\geq 0},$$

being $\mathcal{K}_n$ the set of simplices of $\mathcal{K}$ of dimension n, and Φ_n must commute with face and degeneration morphisms of both $\mathcal{K}$ and $\mathcal{A}_{\mathcal{PL}}$. As set maps, an easy way to represent these objects is just as a collection of pairs $(\sigma \mapsto p)$ where, for each pair, σ is a non-degenerated

simplex, and p is a non-zero polynomial differential form. Although it is not possible to have represented all elements of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, with this approach it is easy to represent a finite set of elements.

Following our example, once we have identified the first non-trivial cohomology group, and obtained a basis of representatives (in our example $H_1 = \mathbb{Q}\langle(1,0,1),(0,1,1)\rangle$, we use the Whitney map to lift each representative to $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. Let h_0^1, h_1^1 be the harmonic cochains. Then, using the representation we mentioned before, the lifted elements we obtain are:

$$\mathcal{W}(h_0^1) = \begin{cases} a & \mapsto & y_1 \\ c & \mapsto & y_1 \\ T_0 & \mapsto & y_1 + y_2 \\ T_1 & \mapsto & y_2 \end{cases}, \quad \mathcal{W}(h_1^1) = \begin{cases} b & \mapsto & y_1 \\ c & \mapsto & y_1 \\ T_0 & \mapsto & y_2 \\ T_1 & \mapsto & y_1 + y_2 \end{cases}.$$

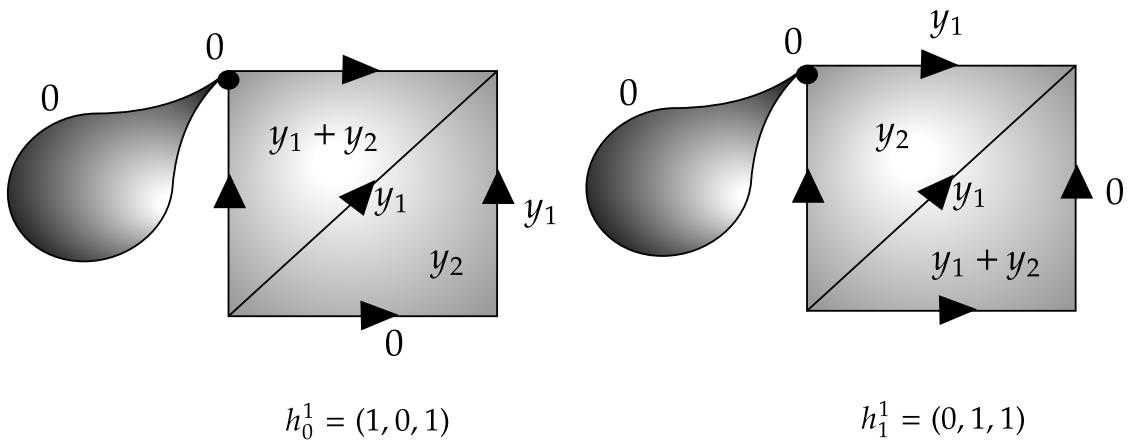


FIGURE III.2. Representatives of $\mathcal{H}^1(\mathcal{A}_{\mathcal{PL}}(\mathbb{S}^2 \vee \mathbb{T}))$. The elements h_0^1 and h_1^1 are the cochain representatives of $\mathcal{H}^1(\mathcal{C}(\mathcal{K}))$ and the above drawings represent the Whitney forms of those elements. This are the representatives of $\mathcal{H}^1(\mathcal{A}_{\mathcal{PL}}(\mathbb{S}^2 \vee \mathbb{T}))$.

A graphical representation of these elements is shown in Figure III.2. Notice that $\mathcal{W}(h_0^1)$ and $\mathcal{W}(h_1^1)$ have constant coefficients even at dimension higher than 1. This is because h_0^1 and h_1^1 are precisely the unit cochains on the horns $\Lambda_1(T_0)$ and $\Lambda_0(T_1)$. This kind of cochains always lift to Whitney forms that assign forms with constant coefficients on higher dimensions than degree.

(1) First step of the algorithm

At this point, we have solved the problem of finding the generators of the first non-trivial cohomology group $\mathcal{H}^1(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$ and also we have obtained the corresponding representatives on $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. Then, we can proceed with the construction of the minimal model as in the original method (see initial step I.1.2). In order to simplify notation, let $\omega_0^1 = \mathcal{W}(h_0^1)$ and $\omega_1^1 = \mathcal{W}(h_1^1)$ be the Whitney forms computed above. For completeness, let us remind the notation used to describe the method (see I.1.2):

- x_j^i will denote a generator of $\mathcal{M}$ of degree i, whose differential is zero.
- y_j^i will denote a generator of $\mathcal{M}$ of degree i, with nonzero differential.

Each time we add a new generator to $\mathcal{M}$, we will use the following diagram

$$b_j^i \quad \xleftarrow{\varphi} \quad y_j^i \quad \xrightarrow{d_{\mathcal{M}}} \quad z_j^{i+1}$$

to denote that we have added the generator y_j^i, with $d_{\mathcal{M}}(y_j^i) = z_j^{i+1} \in \mathcal{M}^{i+1}$ and $\varphi(y_j^i) = b_j^i \in \mathcal{A}_{\mathcal{PL}}^i(\mathcal{K})$.

In our example, after computing ω_0^1, ω_1^1, we initialize the minimal model $(\mathcal{M}, d_{\mathcal{M}})$ with generators x_0^1, x_1^1 and the quasi-isomorphism φ following the diagram

$$\begin{array}{ccccc} \omega_0^1 & \leftarrow & x_0^1 & \rightarrow & 0 \\ \omega_1^1 & \leftarrow & x_1^1 & \rightarrow & 0. \end{array}$$

Doing so, we have that $\varphi : \mathcal{M} \to \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ induces an isomorphism $\mathcal{H}^j(\mathcal{M}) \to \mathcal{H}^j(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$ for all $j \leq 1$.

(2) Increase degree

After getting the induced isomorphism up to degree 1, we proceed computing the cohomology of $\mathcal{M}$ at degree 2. As it has only two generators of degree 1, the graded vector space $\mathcal{M}^2$ is generated only by the element $x_0^1 \wedge x_1^1$. Our objective at this step is to see if

$$\varphi_*^2 : \mathcal{H}^2(\mathcal{M}) \to \mathcal{H}^2(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$$

is injective. As it is not possible to compute $\mathcal{H}^2(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$ with the classical method (quotient of vector spaces), we need to find a way to use the simplicial cochain complex as a proxy. This solution is described in the following section.

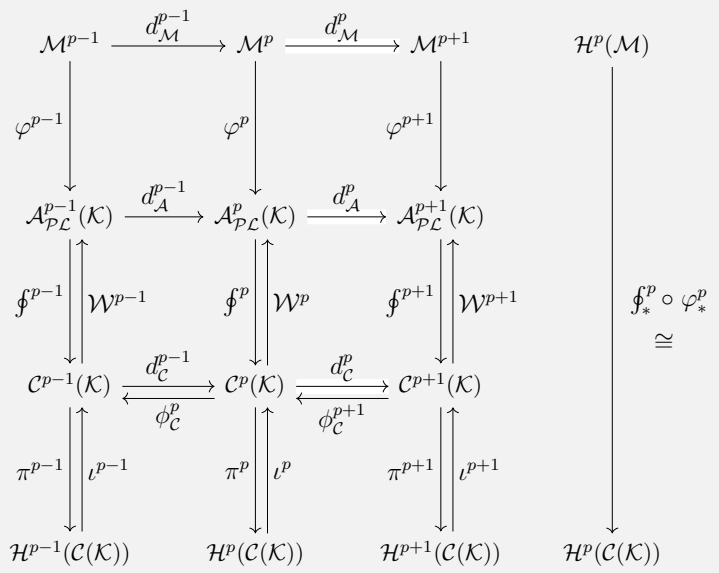
Section § III.2
$\mathcal{C}(\mathcal{K})$ as a proxy to $\mathcal{H}(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$

The solution we found addressing problem **(P2)** involve the use of the simplicial cochain complex $\mathcal{C}^*(\mathcal{K})$ and the simplicial cohomology $\mathcal{H}^*(\mathcal{C}(\mathcal{K}))$ as a proxy between the cohomology of the minimal model $(\mathcal{M}, \varphi)$ and the cohomology of the algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. This approach is based on the fact that $\pi : \mathcal{C}(\mathcal{K}) \to \mathcal{H}(\mathcal{C}(\mathcal{K}))$ and $\oint : \mathcal{A}_{\mathcal{PL}}(\mathcal{K}) \to \mathcal{C}(\mathcal{K})$ are quasi-isomorphisms (in particular, $\pi_* = id_{\mathcal{H}}$) and the induced morphism $\oint_* \circ \varphi_*$:

$$\mathcal{H}(\mathcal{M}) \underset{\cong}{\xrightarrow{\varphi_*}} \mathcal{H}(\mathcal{A}_{\mathcal{PL}}(\mathcal{K})) \underset{\cong}{\xrightarrow{\oint_*}} \mathcal{H}(\mathcal{C}(\mathcal{K})).$$

must be so. The key idea of this approach is the combination of the discrete Hodge decomposition and the AT-model maps π, ι, ϕ, with the quasi-isomorphisms $\oint$, $\mathcal{W}$, in order to move back and forth from $\mathcal{H}(\mathcal{C}(\mathcal{K}))$, $\mathcal{C}(\mathcal{K})$ and $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ without the need of computing $\mathcal{H}(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$.

The use of an AT-Model as a solution to problem (P2)



The morphism $(\oint^{i+1}_* \circ \varphi^{i+1}_*)$ is explicitely computed as follows: for each cohomology class $[z^{i+1}_j] \in \mathcal{H}(\mathcal{M}^{i+1})$, we obtain its representative $z^{i+1}_j \in \mathcal{M}^{i+1}$. Then, we compute its image by $(\pi^{i+1} \circ \oint^{i+1} \circ \varphi^{i+1})$ so that the element

$$\theta^{i+1}_j = (\pi^{i+1} \circ \oint^{i+1} \circ \varphi^{i+1})(z^{i+1}_j) \in H_{i+1},$$

is the harmonic component of the cochain $(\oint \circ \varphi)(z^{i+1})$. This element acts precisely as the coordinate vector of the cohomology class $[\varphi^{i+1}(z^{i+1}_j)]$ at the vector space $\mathcal{H}^{i+1}(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$. We call this element the *harmonic coordinates* of $\varphi(z^{i+1}_j)$. Doing the same for all representatives, we build the induced isomorphism $(\oint^{i+1}_* \circ \varphi^{i+1}_*)$ as it is represented in Figure III.3. This leads us to the following step of the algorithm.

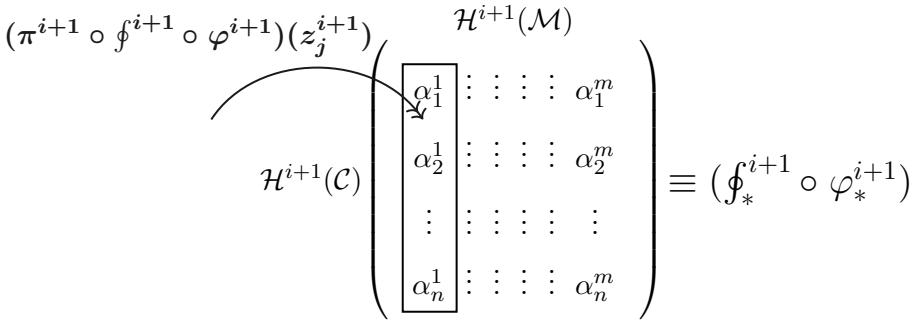


FIGURE III.3. The linear map $(\oint^{i+1}_* \circ \varphi^{i+1}_*)$. The *jth*-column corresponds to the harmonic coordinates of the generator $[z^{i+1}_j] \in \mathcal{H}^{i+1}(\mathcal{M})$ in the vector space $\mathcal{H}^{i+1}(\mathcal{C})$. At step (2.1) of the method, we compute the kernel of this map. If it is not trivial, new generators are added to the model in construction.

Step (2.1): Adding generators of the form y_j^i. The right kernel of the matrix described above give us the cocycles on $\mathcal{M}^{i+1}$ that must be killed in order to make φ_* injective. Then, we proceed iteratively, as in the original method, adding generators of the form y_j^i until the obtained that the induced map in φ_*^{i+1} is injective: if the map $(\oint_*^{i+1} \circ \varphi_*^{i+1})$ is not injective, take $[z_0^{i+1}], \ldots, [z_{k-1}^{i+1}]$ a basis of $\ker(\oint_*^{i+1} \circ \varphi_*^{i+1})$, and consider representatives $\varphi(z_0^{i+1}), \ldots \varphi(z_{k-1}^{i+1}) \in \mathcal{A}_{\mathcal{PL}}^{i+1}(\mathcal{K})$. Since $[z_j^{i+1}]$ is in the kernel of the cohomology map, $\varphi(z_j^{i+1})$ must correspond to a trivial cohomology class, that is, there must be $b_j^i \in \mathcal{A}_{\mathcal{PL}}^i(\mathcal{K})$ such that $d_{\mathcal{A}}(b_j^i) = \varphi(z_j^{i+1})$. So we add the following generators:

$$\begin{array}{ccccc} b_0^i & \leftarrow & y_0^i & \rightarrow & z_0^{i+1} \\ & & \vdots & & \\ b_{k-1}^i & \leftarrow & y_{k-1}^i & \rightarrow & z_{k-1}^{i+1} \end{array}$$

Note that, after adding these generators, new elements of $\ker(\oint_*^{i+1} \circ \varphi_*^{i+1})$ could have been added. We can iterate this process if required adding new generators until the map $\oint_*^{i+1} \circ \varphi_*^{i+1}$ is injective.

This step of the algorithm has been the main contribution of this thesis due to the difficulty to solve the problem of finding a primitive θ of an exact form $d_{\mathcal{A}}(\theta) \in \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. Because of that, we will dedicate next section to explain how we solve this problem. In our example, the unique generator of $\mathcal{M}^2$ $(x_0^1 \wedge x_1^1)$ corresponds precisely with the cohomology generator of degree 2 of the Torus, so the morphism is injective at this stage, and the algorithm continues to the following step.

Step (2.2): Adding generators of the form x_i^k. If the map $\oint_*^{i+1} \circ \varphi_*^{i+1}$ is not surjective, get a basis $[a_0^{i+1}], \ldots, [a_j^{i+1}], [a_{j+1}^{i+1}], \ldots, [a_{l_k}^{i+1}]$, of the harmonic subspace H_{i+1} where $[a_{j+1}^{i+1}], \ldots, [a_{l_k}^{i+1}]$ is a basis of $\mathrm{Im}(\oint_{i+1}^* \circ \varphi_{i+1}^*)$. Then, we obtain the elements $a_0^{i+1}, \ldots, a_j^{i+1} \in \mathcal{A}_{\mathcal{PL}}^{i+1}(\mathcal{K})$ as the image of each $[a_m^{i+1}]$ by the morphism $(\mathcal{W} \circ \iota)$. With this elements, add new generators to $\mathcal{M}$ such that

$$\begin{array}{ccccc} a_0^k & \leftarrow & x_0^k & \rightarrow & 0 \\ & & \vdots & & \\ a_j^k & \leftarrow & x_j^k & \rightarrow & 0. \end{array}$$

We repeat these two steps until $k = i$. Then we repeat step (2.1) one last time to get injectivity in $\oint_*^{i+1} \circ \varphi_*^{i+1}$.

To conclude our example, at step (2.2) the method finds that the map $\oint_*^{i+1} \circ \varphi_*^{i+1}$ is not surjective and add a new generator x_0^2 such that

$$w_0^2 \leftarrow x_0^2 \rightarrow 0.$$

This new element ω_0^2 is represented at Figure III.4 (right). As the space $\mathbb{S}^2 \vee \mathbb{T}$ has dimension 2, for degrees $d > 2$ the cohomology of the space is trivial and the algorithm continues as the original **MMarco**'s method. In particular, the method only will add new generators at the injectivity step (2.1).

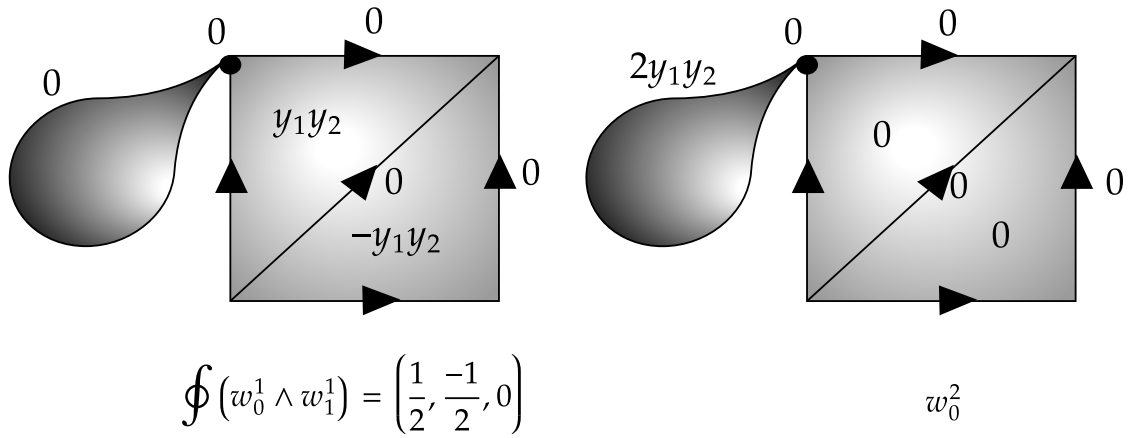


FIGURE III.4. Representatives of $\mathcal{H}^2(\mathcal{A}_{\mathcal{PL}}(\mathbb{S}^2 \vee \mathbb{T}))$. On the left, the product $\omega_0^1 \wedge \omega_1^1$ is precisely a representative of the cohomology generator of degree 2 of the Torus. On the right, the representative of the cohomology generator of the sphere. Both elements represent a basis of $\mathcal{H}^2(\mathcal{A}_{\mathcal{PL}}(\mathbb{S}^2 \vee \mathbb{T}))$.

The next section is dedicated exclusively to describe how the method solves the problem of finding primitives of exact polynomial differential forms, a key step of the algorithm in order to ensure that φ is a quasi-isomorphism of *CDGAs*.

SECTION **§ III.3**

Primitives of exact differential forms on simplicial sets

The main problem we found in the attempt to implement an effective method to compute the minimal model of the algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ is presented here. At **step 7** of **MMarco**'s method (see I.1.2), when the algorithm finds that, at some degee $i+1$, the induced morphism

$$\varphi_*^{i+1} : \mathcal{H}^{i+1}(\mathcal{M}) \to \mathcal{H}^{i+1}(\mathcal{A})$$

is not injective, for each representative $z \in \mathcal{M}^i$ of the $\ker(\varphi_*^{i+1})$ basis, the method needs to find an element $\theta \in \mathcal{A}^i$ such that $d_A(\theta) = \varphi(z)$. In their implementation, the authors formulate the problem as a the inverse problem of solving the system of linear equations where the coefficients matrix is the linear map $d_{\mathcal{A}}^i : \mathcal{A}^i \to \mathcal{A}^{i+1}$. In order to compute that matrix, the method needs to compute a basis for both vector spaces and the coordinates of each element $\varphi(z) \in \mathcal{A}^{i+1}$. The problem of this approach for the algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ is precisely that our vector spaces have not finite type and we can not build such matrix.

In order to overcome this obstacle, during the development of a solution, we found three methods, each one improving some aspect of the previous one. The aim of presenting different solutions has two reasons. The first one is to show our reasoning in cronological order. Our initial approach was basically the brute force, but the attempt to find a smarter solution led us to a better understanding of the underlying structure of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ and we proceed until we found the third method. This method is actually used by default in our implementation. The other reason is because we found in the literature related to fields apparently far from the Sullivan's

theory of minimal models, such as Finite Element Exterior Calculus (or FEEC, [**6, 4, 5**]), that the first method seems to be useful to solve other problems but, as far as we know, there is no general purpose implementations of such method. We consider interesting to present in the same document the different solutions we found as a way to establish bridges between different fields, with the hope that it could be useful for some readers interested in related problems.

III.3.1. Method 1: *Finite Element Exterior Calculus*

In the field of *Finite Element Exterior Calculus* (or FEEC) it is used the algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ with other notation and motivation:

NOTATION (III.3.1) (Arnold, [**6, 4, 5**]). At this section, $\mathcal{P}_r\Lambda^k_n$ will denote the elements of $(\mathcal{A}_{\mathcal{PL}})^k_n$ with polynomial coefficients of degree *up to* r in the polynomial ring $\mathbb{Q}\langle t_0, \ldots, t_n\rangle$. On the other hand, $\mathcal{H}_r\Lambda^k_n$ will denote the elements of $(\mathcal{A}_{\mathcal{PL}})^k_n$ with homogeneous coefficients of degree exactly r. If the dimension of this algebra has no relevance, we will omit the subindex n.

Consider the following property of the cochain algebra $(\mathcal{A}_{\mathcal{PL}})_n$ using the notation defined above:

PROPOSITION (III.3.2) ([**6, 4, 5**]). *Given an element $\omega \in \mathcal{P}_{r+1}\Lambda^k$, its differential lies in the polynomial space $d(\omega) \in \mathcal{P}_r\Lambda^{k+1}$, i.e., the differential decreases the coefficients degree by* 1 *and increases the form degree by* 1.

Furthermore, the cochain algebra $\mathcal{H}_*\Lambda_n = \{\mathcal{H}_*\Lambda^k_n\}_{0\leq k\leq n}$ has a mixed structure ([**42**]), i.e., there exists a differential of degree -1, called the *koszul differential*:

$$\kappa : \mathcal{H}_r\Lambda^{k+1}_n \to \mathcal{H}_{r+1}\Lambda^k_n$$

such that

$$\kappa \circ \kappa = 0$$

and

$$\kappa(\omega \wedge \theta) = \kappa(\omega) \wedge \theta + (-1)^p \omega \wedge \kappa(\theta),$$

for polynomial differential forms $\omega \in \mathcal{H}_*\Lambda^p_n$ and $\omega \in \mathcal{H}_*\Lambda^q_n$. This differential defines a chain complex structure on $(\mathcal{H}_*\Lambda_n, \kappa)$, called the *Koszul complex*. Moreover, κ is a contraction for the homogeneous polynomial de Rham complex ([**6**]) and both differentials are related by the equation [**42, 4**]:

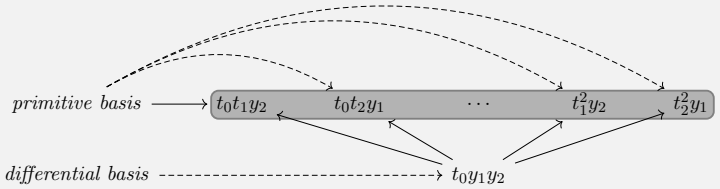
$$((d \circ \kappa) + (\kappa \circ d))(\omega) = (r + k) \cdot \omega$$

for $\omega \in \mathcal{H}_r\Lambda^k$. However, although the differential d is compatible with the underlying relations of the algebras $(\mathcal{A}_{\mathcal{PL}})_n$, it is not the case for κ (different representatives of the same object in the algebra map to different objects). It is a problem that does not guarantee that the simplicial constraints are met when it is applied to the algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$.

In order to address the problem of finding a primitive of an exact differential form of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, the first method we present consist on take advantage of the property (III.3.2). As Whitney forms are first order polynomial diffeerential forms, this property give us an upper bound of the polynomial space in order to truncate the differential matrix by the degree of the coefficients.

Key idea: Truncate the differential matrix

For each polynomial differential form $\omega \in \mathcal{P}_r\Lambda_n^k$ assigned to each simplex $\sigma \in \mathcal{K}_n$, first compute a primitive basis of all polynomial differential forms with coefficients up to degree $r+1$.



Then, the differential matrix is built adding to the linear system the simplicial restrictions in order to guarantee that the solution satisfies the patch conditions.

We compute a primitive basis locally, i.e., for each simplex. Then, we add to the linear system the compatibility conditions in order to satisfy the face map restrictions. This restrictions can be described as follows:

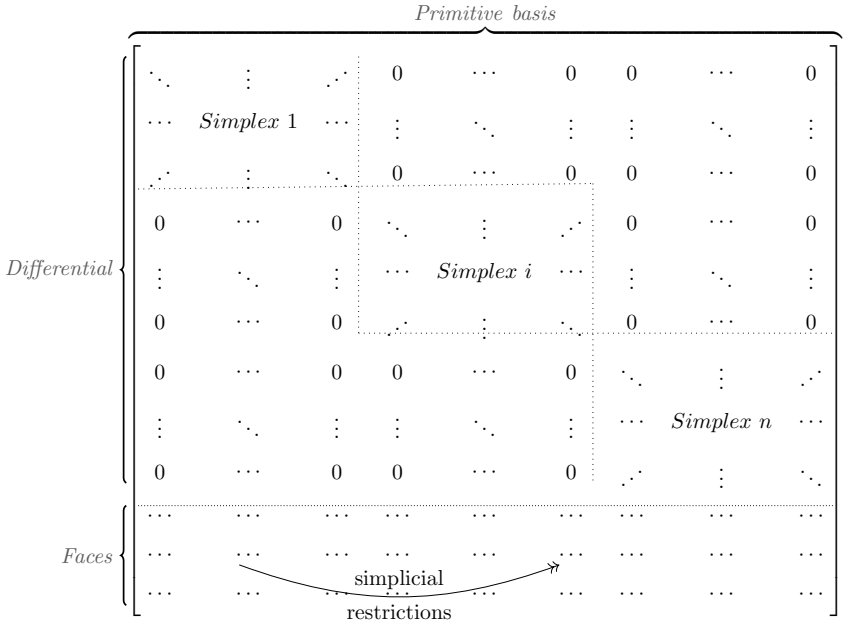


FIGURE III.5. Representation of the coefficients matrix of the inverse linear system problem described in this section.

Let $\Phi \in \mathcal{A}^p_{\mathcal{PL}(\mathcal{K})}$ be an exact form. This condition guarantees us that $\exists \Psi \in \mathcal{A}^{p-1}_{\mathcal{PL}(\mathcal{K})}$ so that $d(\Psi) = \Phi$. Let $\sigma_1, \sigma_2 \in \mathcal{K}$ so that $\partial_i(\sigma_1) = \partial_j(\sigma_2)$, for some pair of faces i, j. Consider the differential forms $\omega_1 = \Phi(\sigma_1)$ and $\omega_2 = \Phi(\sigma_2)$. The primitives $\theta_1 = \Psi(\sigma_1)$ $\theta_2 = \Psi(\sigma_2)$ must satisfy the conditions $d(\Psi(\sigma)) = \Phi(\sigma)$ and $\partial_i(\Psi(\sigma_1)) = \partial_j(\Psi(\sigma_2))$, for every $\sigma_1, \sigma_2 \in K$.

So by solving the complete system of linear equations we obtain the desired element Ψ. In Figure III.5 it is represented the structure of the coefficients matrix of such system. The upper half of the matrix is a block matrix, related to compute locally (for each simplex) a

primitive of the corresponding differential form. The bottom half of the matrix represents the affine structure of the simplicial object. Each row indicates which basis elements at different simplices correspond with the same polynomial. Just to finish with this method, we refer the reader to previous publications [**3, 2**] where we described some examples and applications (using an old-fashioned notation).

III.3.2. Method 2: The combinatorial model

Although the first method solves our problem, the computational cost of building the differential matrix makes it, in practice, untractable for large simplicial sets. In order to improve the performance and reduce the computational complexity, we detect that one of the bottlenecks of the previous method is the computation of the simplicial restrictions. This part of the method implies the addition of one equation to the linear system for every pair of simplices $\sigma_1, \sigma_2 \in \mathcal{K}$ such that $\sigma_1 \cap \sigma_2 \neq \emptyset$ and for every element of the primitive basis. In order to avoid this step, we modified the previous method following the ideas from [**25**]. First, we introduce the combinatorial model, and later we will describe how we use this representation to solve our problem.

The combinatorial model of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. In [**25**], Felix et al. verified the assertion made by Sullivan, that the rational de Rham algebra

$$\mathcal{A}_{\mathcal{PL}}(\mathcal{K}) = \bigoplus_{p\geq 0} \mathcal{A}^p_{\mathcal{PL}}(\mathcal{K})$$

of a finite simplicial set $\mathcal{K}$ has an explicit and direct combinatorial description, closely related to the Stanley-Reisner face ring of a simplicial complex. In particular, for a finite simplicial set $\mathcal{K}$, the authors proved the isomorphism between the rational algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ and other algebra, namely $\mathcal{A}(\mathcal{K})$, defined as follows:

Let $[n]$ be the underlying simplicial set of the standard simplex Δ^n (adding the corresponding degenerated simplices). This simplicial set has exactly $(n+1)$ non-degenerated vertices. Let $(\mathcal{A}_{\mathcal{PL}})_n$ be the algebra of polynomial differential forms of $[n]$. Recall that

$$(\mathcal{A}_{\mathcal{PL}})_n = \frac{\Lambda(t_0, \ldots, t_n, y_0, \ldots, y_n)}{\langle \sum_{i=0}^n t_i - 1, \sum_{j=0}^n y_j \rangle}$$

and the differential is defined as $d(t_i) = y_i$, and $d(y_j) = 0$. Now, consider the simplicial set $\mathcal{K} \subseteq [n]$ as a subset of $[n]$ such that n is the least dimension for which the inclusion:

$$\iota : \mathcal{K} \hookrightarrow [n]$$

determines a simplicial set morphism so that the image of a non-degenerated simplex is also a non-degenerated simplex. Let J be the set of non-degenerated simplices of $[n]$ that are not in the image of the inclusion $\iota : \mathcal{K} \hookrightarrow [n]$:

$$J = \{\sigma \in [n] | \sigma \text{ is non-degenerated and } \nexists \tau \in \mathcal{K} \text{ s.t } \iota(\tau) = \sigma)\}.$$

Then, each p-simplex $\sigma \in J$ is uniquely determined by its barycentric coordinates $\lambda_0, \ldots, \lambda_p$ corresponding to the oriented sequence of its vertices. The inclusion $\iota : \mathcal{K} \to [n]$ determines a simplicial set morphism of cochain algebras between $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ and $\mathcal{A}_{\mathcal{PL}}([n])$. Let fix some notation: for each p-simplex $\sigma = (\lambda_{i_0}, \ldots, \lambda_{i_p}) \in [n]$, let

- $t_\sigma = t_{\lambda_{i_0}} \cdots t_{\lambda_{i_p}} \in (\mathcal{A}_{\mathcal{PL}})_n$
- $y_\sigma = y_{\lambda_{i_0}} \cdots y_{\lambda_{i_p}} \in (\mathcal{A}_{\mathcal{PL}})_n$

be the monomials of degrees 0 and $(p+1)$ respectively, determined by the barycentric coordinates of the simplex σ. With this notation, consider the ideal on $(\mathcal{A}_{\mathcal{PL}})_n$

$$I_K = \langle \{ t_\sigma y_\tau \in (\mathcal{A}_{\mathcal{PL}})_n | \sigma, \tau \notin J, \text{ but } \sigma \star \tau \in J \} \rangle$$

where, for non-degenerated simplices $\sigma = (\lambda_{i_0}, \ldots, \lambda_{i_p})$ and $\tau = (\lambda_{j_0}, \ldots, \lambda_{j_q})$, $\sigma \star \tau$ is the simplex of $[n]$ determined by the non-decreasing ordered sequence of the set $\{\lambda_{i_0}, \ldots, \lambda_{i_p}\} \cup \{\lambda_{j_0}, \ldots, \lambda_{j_q}\}$. Then, Felix et al. defined at [**25**] the algebra:

$$\mathcal{A}(\mathcal{K}) = \frac{(\mathcal{A}_{\mathcal{PL}})_n}{I_K},$$

and verified that this algebra is isomorphic to the algebra of polynomial differential forms $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. The idea behind this definition is represented in Figure III.6. The ideal represents the geometrical points of the standard simplex $|\Delta^n| \subset \mathbb{R}^{n+1}$ that are not in the intersection $|\mathcal{K}| \cap |\Delta^n|$, being $|\mathcal{K}|$ the geometrical realization of the simplicial set $\mathcal{K}$ viewed as a simplicial subset of Δ^n.

Our solution. Our second approach to find a primitive of an exact differential form is based on the same idea of the previous one, in the sense that we need to solve a system of linear equations where the differential matrix is truncated by the degree of the coefficients. However, in this case the domains and codomains of the differential are build in a different way.

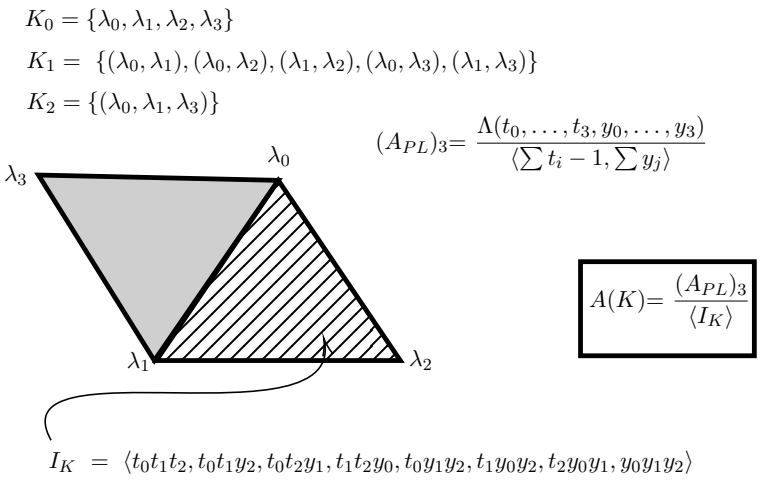


FIGURE III.6. Graphical representation of the combinatorial algebra of polynomial differential forms $\mathcal{A}(\mathcal{K})$.

Considering the property of the cochain algebra $(\mathcal{A}_{\mathcal{PL}})_n$ mentioned in proposition (III.3.2), and the fact that Whitney forms are first order polynomial differential forms (see (II.1.6)), given two Whitney forms $\omega_1, \omega_2 \in \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, their product will be a simplicial set map, assigning to each simplex, a polynomial differential form of degree up to 2. If the product $\omega_1 \wedge \omega_2$ is an exact element, a primitive $d(\theta) = \omega_1 \wedge \omega_2$ will be a simplicial set map assigning to each simplex, a polynomial differential form with coefficients of degree up to 3. Applying this argument iteratively, we deduce that $\omega_1, \omega_2 \in \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ are elements with coefficients degree up to r and s respectively for which $\omega_1 \wedge \omega_2$ is an exact forms, then a primitive of such element will have coefficients of degree up to $r + s + 1$. So, using that argument, our second method proceeds as follows:

Let $\mathcal{K}$ be a simplicial set, and let $d(\theta) \in \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ be an exact element of degree p. Assume $d(\theta)$ is homogeneous (for non-homogeneus elements, just apply the method for each homogeneous component and the solution will be the sum of all homogeneous primitives). First, we identify the maximum coefficients degree over all differential forms in the image of $d(\theta)$. Let r be such degree.

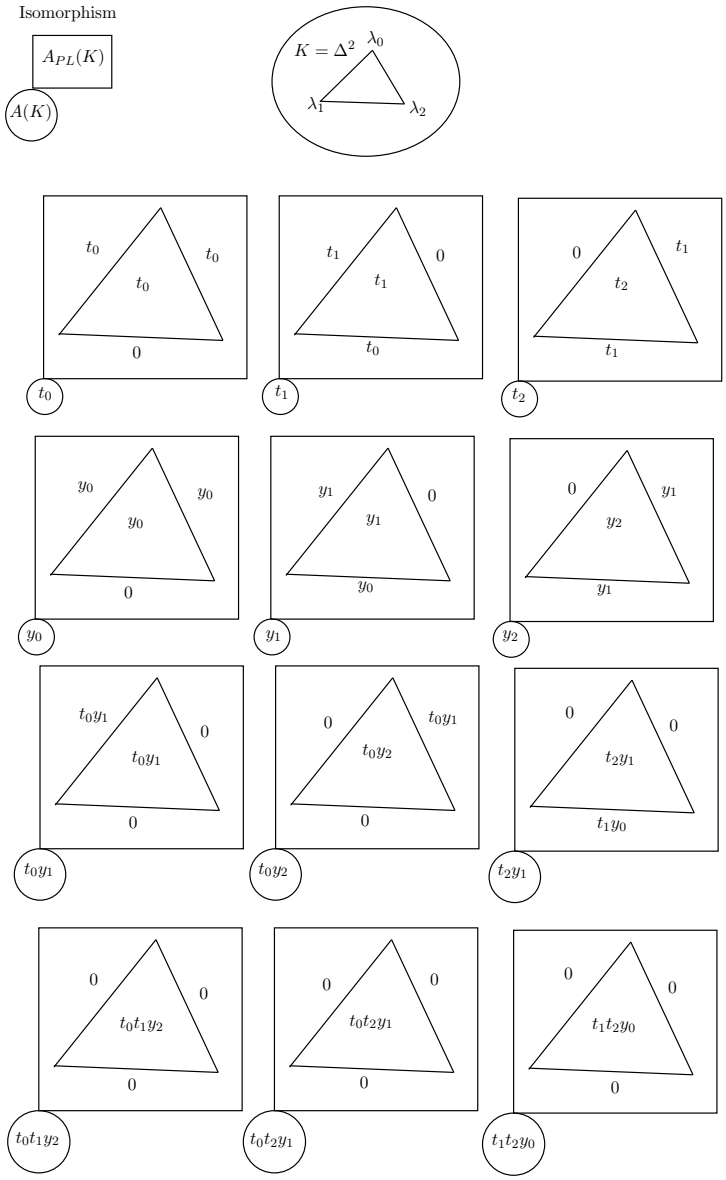
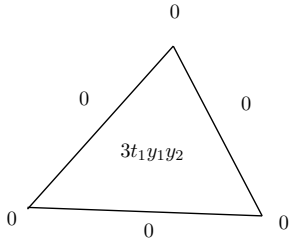


FIGURE III.7. Ilustrative representation of the isomorphism $\mathcal{A}(\Delta^2) \cong \mathcal{A}_{\mathcal{PL}}(\Delta^2)$ respect to elements of degree 1 with coefficients up to degree 2. It is easy to check the relations $\sum t_i = 1$ and $\sum y_j = 0$ at both sides of the isomorphism. Furthermore, the naturality of this isomorphism shows up noticing that products respect the structure.

Then, consider the combinatorial model $\mathcal{A}(\mathcal{K}) = (\mathcal{A}_{\mathcal{PL}})_n / I_{\mathcal{K}}$ described at the beginning of the section. Using the notation of the previous section, compute the set of elements $\mathcal{A}(\mathcal{K}) \cap \mathcal{P}_{r+1}\Lambda_n^p$. Now, as we do not deal with differential gröbner basis [**48**], once we have a generator set, we

come back and consider those elements in $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. This elements form a generating set of the subspace of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ where our primitive can be found. So, using these elements build the differential matrix. The element $\theta \in \mathcal{A}^{p-1}(\mathcal{K})$ will be a linear combination of such differential forms. In order to ilustratively sketch the isomorphism $\mathcal{A}(\mathcal{K}) \cong \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, in Figure III.7 it is shown the relation between the subset of generators of $\mathcal{A}(\Delta^2)$ and their counterparts in $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$.

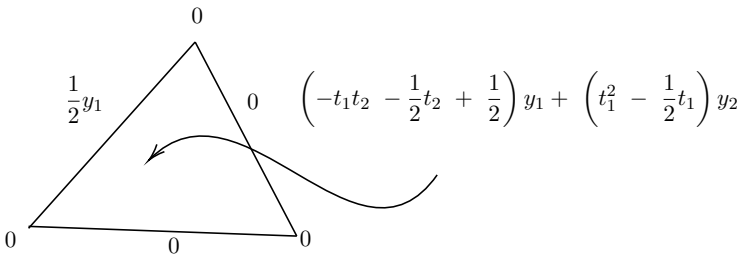
Example (III.3.3). Let $\mathcal{K} = \Delta^2$ as in Figure III.7 and consider the element $\omega \in \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ defined by:



Then, as the coefficient has degree 1, the algorithm gets a basis for all the elements of exterior degree 1 with coefficients of degree up to 2 in $\mathcal{A}(\mathcal{K})$. This elements are:

$$\begin{pmatrix} y_1 & y_2 & t_1y_1 & t_1y_2 & t_2y_1 & t_2y_2 \\ t_1^2y_1 & t_1^2y_2 & t_2^2y_1 & t_2^2y_2 & t_1t_2y_1 & t_1t_2y_2 \end{pmatrix}$$

Take into account that the algebra $\mathcal{A}(\mathcal{K})$ has the relations $\sum t_i - 1$ and $\sum y_j = 0$, and a basis does not contain all possible combinations. Once the primitive basis has been computed, the method gets a basis of the image by the differential and gets the coordinates of ω in such basis. Finally, the method solves the inverse problem, finding the primitive:



Concluding this section, With this approach, as we are making algebraic operations directly using the global structure of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$, instead of working locally at every simplex as in the previous method, we ensure that, if the system finds a solution, it will satisfy the patch conditions without a need to check it. This method improves significantly the computational performance of building the differential matrix, but the combinatorial explosion for large simplicial sets remains at the core of the method. In practice, this method limits us to work with small simplicial structures. This is the reason why the third and last method has been developed.

III.3.3. Method 3: Dupont's contraction

The third and last method we propose for the effective computation is actually the unique one we have found capable to work with large simplicial sets. The method just implements Dupont's contraction. This method finds a primitive of an exact differential form $\omega \in \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$

without the need of solving any kind of linear system. The method just applies Dupont's contraction locally, simplex by simplex, satisfying the patch conditions by definition. The theoretical definition can be found at section II.3, based on the description of Cheng et al. [**13**]. Although Dupont at [**19**] explained such contraction, personally, I would like to thank the authors of this article for the concrete and precise definition.

The computation of a primitive using Dupont's contraction is as follows: suppose we have an element $\Psi \in \mathcal{A}^{i+1}_{\mathcal{PL}}(\mathcal{K})$. By Dupont's contraction we know that:

$$\Psi - (\mathcal{W} \circ \oint)(\Psi) = (s \circ d)(\Psi) + (d \circ s)(\Psi).$$

Additionally, suppose that $\Psi = d\theta$. Then,

$$d\theta - (\mathcal{W} \circ \oint)(d\theta) = \cancel{(s \circ d)(d\theta)} + (d \circ s)(d\theta).$$

Moreover, we know that the differential commutes with integration and Whitney's map:

$$d(\theta) - (d \circ \mathcal{W} \circ \oint)(\theta) = (d \circ s)(d\theta).$$

Then,

$$d(\theta) = d((s \circ d)(\theta) + (\mathcal{W} \circ \oint)(\theta)).$$

So, **the element we are looking for is:**

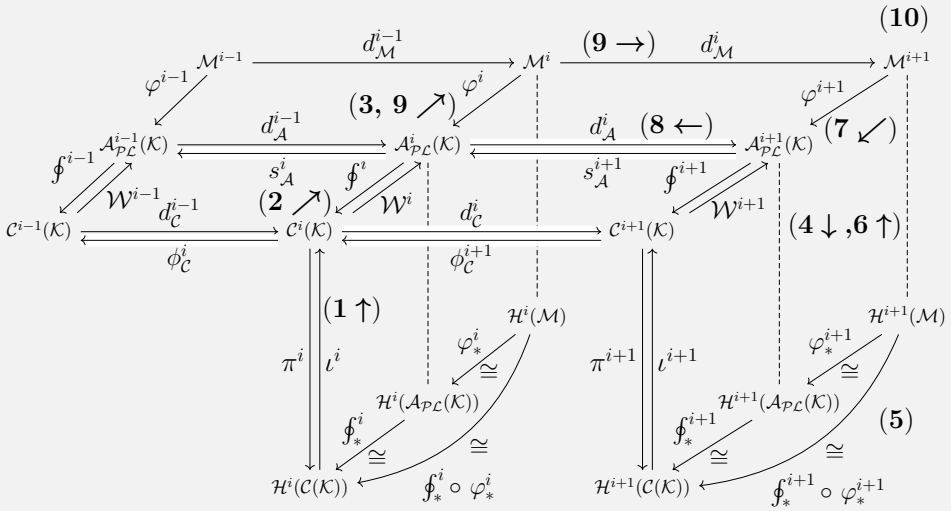
$$\theta = s(d\theta) + \mathcal{W}(\alpha),$$

where α is a solution of the linear system $d_{\mathcal{C}}(\alpha) = \oint(d\theta)$.

Analyzing our contribution in relation to Dupont's contraction, there is no new theoretical result. After noticing the existence of this contraction, we found a recent implementation [**14**], but as far as we know, this is the first computational tool that integrates that contraction in the context of Sullivan models of general finite simplicial sets and discrete hodge decompositions. At appendix C, it is shown an example where we apply that contraction for a 3-form over a simplicial complex of dimension 4 and 1704 simplices, and show the polynomial differential forms obtained by such contraction.

For the concrete implementation and documentation see the references in section III.5. A graphical summary of the new method can be visualized in the following diagram:

Key steps for the computation of $\varphi : \mathcal{M} \to \mathcal{A}_{\mathcal{PL}}(\mathcal{K})$



(1) Lifts generators $\iota^i : \mathcal{H}^i(\mathcal{C}(\mathcal{K})) \to \mathcal{C}^i(\mathcal{K})$.
(2) Lifts cochain representatives $\mathcal{W}^i : \mathcal{C}^i(\mathcal{K}) \to \mathcal{A}^i_{\mathcal{PL}}(\mathcal{K})$.
(3) Updates $(\mathcal{M}^i, \varphi^i)$ in order to make $\oint^i_* \circ \varphi^i_*$ surjective.
(4) Gets a basis of $\mathcal{H}^{i+1}(\mathcal{M})$.
(5) Computes the kernel of the induced morphism $\oint^{i+1}_* \circ \varphi^{i+1}_*$.
(6) Lifts the generators of $\ker(\oint^{i+1}_* \circ \varphi^{i+1}_*)$.
(7) For each $z \in \ker(\oint^{i+1}_* \circ \varphi^{i+1}_*)$, gets $\varphi(z) \in \mathcal{A}^{i+1}_{\mathcal{PL}}(\mathcal{K})$.
(8) For each $\varphi(z) \in \mathcal{A}^{i+1}_{\mathcal{PL}}(\mathcal{K})$, finds $\theta \in \mathcal{A}^i_{\mathcal{PL}}(\mathcal{K})$ such that $d_{\mathcal{A}}(\theta) = \varphi(z)$.
(9) Updates $(\mathcal{M}^i, \varphi^i)$ in order to make $\oint^{i+1}_* \circ \varphi^{i+1}_*$ injective.
(10) $(\mathcal{M}, \varphi)$ is now an i-minimal model of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$.

Arrow notation: having the needed elements → compute/modify/link to.

Section § III.4

Effective computation of Dupont's formula

In this last section, we show a concrete implementation of the product defined by Dupont in [**19**]. As we saw in section II.2, Dupont's product is defined by the formula

$$(\alpha \sqcup\!\sqcup \beta)(\sigma) = \frac{p!q!}{(p+q+1)!} \sum_{I,J} sgn(I,J) \cdot \alpha(\overline{\partial}_I \sigma) \cdot \beta(\overline{\partial}_J \sigma).$$

For cochains $\alpha \in \mathcal{C}^p(\mathcal{K})$ and $\beta \in \mathcal{C}^q(\mathcal{K})$, this product is defined locally at a simplex $\sigma_n \in \mathcal{K}_n$, where $n = p + q$. We found that this product has a very efficient effective way to be done using the face matrices of the simplicial structure. The effective computation is done as follows:

Let α, β be two cochains as described above. Let $\overline{\partial}_I \in I_p^n$ and $\overline{\partial}_J \in I_q^n$ be two q- and p- face paths from dimensions p and q respectively to dimension $n = p + q$. This paths can be seen as strictly increasing sequences of indices $\overline{\partial}_I = (i_1, \ldots, i_q)$ and $\overline{\partial}_J = (j_1, \ldots, j_p)$. In order to define a basis for the vector spaces $\mathcal{C}^*(\mathcal{K})$ we need to fix a total order in the sets $\mathcal{K}_n$. Let $\sigma_n \in \mathcal{K}_n$ be the n-simplex located at position k in that basis. Then, looking at the formula, the terms $\alpha(\overline{\partial}_I \sigma_n)$ and $\alpha(\overline{\partial}_I \sigma_n)$ are no other thing than the element at positon k of the vector resulting of multiplying the transpose face matrices in the sequence determined by the face paths I, J respectively.

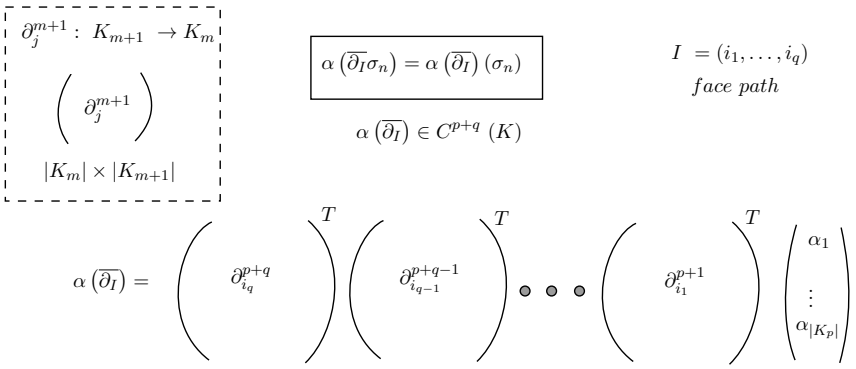


FIGURE III.8. The propagation of the cochain $\alpha \in \mathcal{C}^p(\mathcal{K})$ to dimension $n = p+q$.

Using the notation defined at Figure III.8), this means that, for a fixed pair of face paths I, J, we can compute at the same time the products $\alpha(\overline{\partial}_I \sigma_n) \cdot \alpha(\overline{\partial}_I \sigma_n)$ for all $\sigma_n \in \mathcal{K}_n$. This is done computing the pairwise product of $\alpha(\overline{\partial}_I)$ and $\beta(\overline{\partial}_J)$. Finally, computing the proper $sgn(I, J)$, summing all the products and multiplying by the multinomial coefficient, we obtain the corresponding cochain.

Although we do not use it for the computation of the minimal model, in the software we provide in this thesis, we give a concrete implementation of this product at the cochain level, without the need of creating any polynomial algebra of differential forms. We found that this product at the cochain level has the potential to explore more powerful computational techniques in order to improve substantially the performance of the method. We think that, if it is possible to generalize the formula we mentioned at the end of section II.2.1, this approach would allow to effectively apply rational homotopy theory to large datasets. In this direction, we found that simultaneously to the development of this thesis, Šimková carried out theoretical advances that support this idea [**54, 55**]. We conclude saying that, as future work, we will study her results in order to see if it is possible to interconnect both thesis in order to advance in the development of an actual effective definition of Persistent Rational Homotopy theory [**40, 65**].

SECTION § III.5
The SullivanMM package: source code and documentation

A concrete implementation of the effective method described in this chapter is available at `https://github.com/calquezar/sullivanmm`. The documentation is available onlie at `https://riemann.unizar.es/~calquezar/sullivan`. The package has been developed in SageMath [**17**].

IV

Examples

In this chapter are included some examples in order to illustrate the algorithm and its applications using our implementation of the method for SageMath. The software is available at GitHub (see section III.5).

The first example shows how it is used the program to compute the minimal model of the complement of a smooth projective curve. In the second example, we compute the 3-minimal model of the $\mathcal{K}3$ surface. It is a classical result that this space is formal [**16**], so we use it to verificate its formality as a non-trivial simplicial complex of dimension 4.

Section § IV.1
Complement of smooth projective curves

In a recent work [**7**], Artal et al. give an explicit construction of a 2-dimensional Δ-complex that is a strong deformation retract of the complement of a Fermat curve of degree d in the complex projective space.

Let's see how to compute the 3-minimal model of the complement of a smooth cubic curve with our implementation:

```
sage: from Sullivan_MM import *
sage: from sage.topology.simplicial_set import SimplicialSet
sage: d = 3
sage: Kd=DeltaComplex(cells_of_Kd(d))
sage: S=SimplicialSet(Kd)
sage: WS = APLK(S)
sage: MS = WS.minimal_model(i=3,max_iterations=3,verbose=False,
                            partial_result=True)
sage: MS.domain()
Commutative Differential Graded Algebra with generators ('x2_0',
'x2_1', 'y3_0', 'y3_1', 'y3_2') in degrees (2, 2, 3, 3, 3) over
Rational Field with differential:
   x2_0 --> 0
   x2_1 --> 0
   y3_0 --> x2_0^2
   y3_1 --> x2_0*x2_1
```

```
   y3_2 --> x2_1^2
```

We use the function cells_of_Kd, that outputs the simplicial data computed by the method described in [**7**]. We observed that the minimal model has 2 generators of degree 2, which means that, if the space would be a Sullivan space, its second rational homotopy group would have rank 2.
However, if we compute the minimal model of its universal cover:

```
sage: S=S.set_base_point(S.all_n_simplices(0)[0])
sage: U = S.universal_cover()
sage: WU = APLK(U)
sage: MU = WU.minimal_model(i=3,max_iterations=3,verbose=False,
                            partial_result=True)
sage: MU.domain()
Commutative Differential Graded Algebra with generators ('x2_0',
'x2_1', 'x2_2', 'x2_3', 'x2_4', 'x2_5', 'x2_6', 'x2_7', 'y3_0',
'y3_1', 'y3_2', 'y3_3', 'y3_4', 'y3_5', 'y3_6', 'y3_7', 'y3_8',
'y3_9', 'y3_10', 'y3_11', 'y3_12', 'y3_13', 'y3_14', 'y3_15',
'y3_16', 'y3_17', 'y3_18', 'y3_19', 'y3_20', 'y3_21', 'y3_22',
'y3_23', 'y3_24', 'y3_25', 'y3_26', 'y3_27', 'y3_28', 'y3_29',
'y3_30', 'y3_31', 'y3_32', 'y3_33', 'y3_34', 'y3_35') in
degrees (2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3) over Rational Field with differential:
   x2_0 --> 0
   x2_1 --> 0
   x2_2 --> 0
   x2_3 --> 0
   x2_4 --> 0
   x2_5 --> 0
   x2_6 --> 0
   x2_7 --> 0
   y3_0 --> x2_0^2
   y3_1 --> x2_0*x2_1
   y3_2 --> x2_1^2
   y3_3 --> x2_0*x2_2
   y3_4 --> x2_1*x2_2
   y3_5 --> x2_2^2
   y3_6 --> x2_0*x2_3
   y3_7 --> x2_1*x2_3
   y3_8 --> x2_2*x2_3
   y3_9 --> x2_3^2
   y3_10 --> x2_0*x2_4
   y3_11 --> x2_1*x2_4
   y3_12 --> x2_2*x2_4
   y3_13 --> x2_3*x2_4
   y3_14 --> x2_4^2
   y3_15 --> x2_0*x2_5
```

```
y3_16 --> x2_1*x2_5
y3_17 --> x2_2*x2_5
y3_18 --> x2_3*x2_5
y3_19 --> x2_4*x2_5
y3_20 --> x2_5^2
y3_21 --> x2_0*x2_6
y3_22 --> x2_1*x2_6
y3_23 --> x2_2*x2_6
y3_24 --> x2_3*x2_6
y3_25 --> x2_4*x2_6
y3_26 --> x2_5*x2_6
y3_27 --> x2_6^2
y3_28 --> x2_0*x2_7
y3_29 --> x2_1*x2_7
y3_30 --> x2_2*x2_7
y3_31 --> x2_3*x2_7
y3_32 --> x2_4*x2_7
y3_33 --> x2_5*x2_7
y3_34 --> x2_6*x2_7
y3_35 --> x2_7^2
```

We observe that the minimal model of the universal cover has 8 generators of degree 2. As it is simply-connected, it must be a Sullivan space, so the π_2 of the universal cover (which coincides with the π_2 of the space itself) has rank 8. As this doesn't match the rank 2 obtained before, we can conclude that the complement of the smooth cubic is not a Sullivan space. Furthermore, we can see that its rational π_3 has rank 36. We don't show it here because it is too large for a page, but the fourth rational homotopy group can also be computed in a few seconds, and it has rank 168. Similar computations can show that the rational homotopy groups of the complement of smooth quartics have ranks 27 and 378 respectively.

Section § IV.2
Formality of the $\mathcal{K}3$ Surface

In this second example, we compute the minimal model of a particular space known as the $\mathcal{K}3$ surface and we use it to verificate its formality [**16**]. This space is simply connected and it is available a concrete simplicial complex decomposition at the SageMath repository [**11**]. This simplicial structure has dimension 4 and it is build with 16 vertices and 288 facets (in this case, all facets are simplices of dimension 4). In Table 1 are listed the number of simplices for each dimension. It was constructed by Casella and Kühnel in [**10**]. The implementation in SageMath uses the labeling from Spreer and Kühnel [**56**].

	Dimension				
	0	**1**	**2**	**3**	**4**
Nº Simplices	16	120	560	720	288

Table 1. Number of simplices of the $\mathcal{K}3$ Surface in each dimension.

The Betti numbers of the space are listed in Table 2. Omitting degree 0, the cohomology has 22 generators of degree 2 and one generator of degree 4.

	Degree				
	0	**1**	**2**	**3**	**4**
Nº of Cohomology Generators	1	0	22	0	1

TABLE 2. Number of cohomology generators of the $\mathcal{K}3$ surface for each degree.

Due to Terzić [**59**], it is known that:

THEOREM (IV.2.1). *Let M be a closed oriented simply connected four-manifold and b_2 its second Betti number. Then:*

1) *If $b_2 = 0$ then $\operatorname{rk}\pi_4(M) = \operatorname{rk}\pi_7(M) = 1$ and $\pi_p(M)$ is finite for $p \neq 4, 7$,*
2) *If $b_2 = 1$ then $\operatorname{rk}\pi_2(M) = \operatorname{rk}\pi_5(M) = 1$ and $\pi_p(M)$ is finite for $p \neq 2, 5$,*
3) *If $b_2 = 2$ then $\operatorname{rk}\pi_2(M) = \operatorname{rk}\pi_3(M) = 2$ and $\pi_p(M)$ is finite for $p \neq 2, 3$,*
4) *If $b_2 > 2$ then $\dim \pi_*(M) \otimes \mathbb{Q} = \infty$ and*

$$\operatorname{rk}\pi_2(M) = b_2, \quad \operatorname{rk}\pi_3(M) = \frac{b_2(b_2+1)}{2} - 1, \quad \operatorname{rk}\pi_4(M) = \frac{b_2(b_2^2-4)}{3}.$$

In particular, as the $\mathcal{K}3$ surface has $b_2 = 22$, and following the theorem (IV.2.1), the rank of the first homotopy groups are (see Table 3):

	$\pi_n(\mathcal{K}3)$			
	1	**2**	**3**	**4**
rank	0	22	252	3520

TABLE 3. Rank of the first homotopy groups of K3.

Our implementation allows us to compute the i-minimal model of this space. As it is simply connected, the minimal model carries topological information. In particular, for $i = 3$:

```
sage: from Sullivan_MM import *
sage: K = simplicial_complexes.K3Surface()
sage: W = APLK(K)
sage: M = W.minimal_model(i=3)
sage: M.numerical_invariants()
{1: [0], 2: [22, 0], 3: [0, 252, 0]}
sage: len(M.basis(2))
22
sage: len(M.basis(3))
252
sage: len(M.basis(4))
253
sage: len(M.basis(5))
5544
```

In particular in this example, for degrees 2 and 3, the number of generators at each degree match the ranks of the rational homotopy groups indicated in table 3. Take into account that we are computing the i-minimal model, not the full minimal model. It means that our method guarantees that $\varphi_* : \mathcal{H}(\mathcal{M}) \to \mathcal{H}(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$ is an isomporphism up to degree i, and it is injective at degree $i+1$. As we computed the 3-minimal model, for degrees greater that 3, the ranks do not say anything about the rational homotopy groups. In other computations, we verified also that $\mathcal{M}^4$ and π_4 have exactly the same ranks.
Due to the extension of the output, the generators and their differentials are included in Appendix B. Also, as an ilustrative example of the complexity of the underlying differential forms, in Appendix C are included some computations and the internal represenation of the $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ elements.

In addition to the computation of the minimal model, our implementation allows us to ask if the space itself is i-formal (for the definition of formal spaces see I.3.6).

```
sage: W = APLK(K)
sage: W.is_formal(i=3)
True
```

For the case of the $\mathcal{K}3$ surface, we obtain that it is 3-formal. Applying the following theorem:

THEOREM (IV.2.2) ([**45**]). *If a nilpotent space $\mathcal{X}$ of dimension d is $(d-1)$-formal, then it is formal.*

A particular case of nilpotent spaces are the simply-connected spaces. Then, with all these results, we can deduce that:

FORMALITY'S PROOF OF THE $\mathcal{K}3$. As the $\mathcal{K}3$ is simply-connected, and we have verified that it is 3-formal, then it is formal. □

CONCLUSIONS

The main objective of this thesis has been to develop and implement, within a computer algebra system, an effective method for the computation of the Sullivan minimal model of the algebra of polynomial differential forms $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ over a finite type simplicial set $\mathcal{K}$. Although the **MMarco** method provided an effective solution for finitely generated commutative differential graded algebras (CDGA), its direct application to $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ was not feasible. The fundamental reason lies in the fact that the degree zero homogeneous part of this algebra is infinite-dimensional, which makes it impossible to construct the differential matrices necessary for the linear algebra computations on which the original algorithm is based.

To overcome this structural limitation, this work has introduced three key methodological contributions that allow bypassing these computational obstacles:

- **Effective representation:** A representation of the elements of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ based on the discrete Hodge decomposition of the cochain complex $\mathcal{C}(\mathcal{K})$ has been proposed. This allows working with a computationally manageable generating set, indexed by the non-degenerate simplices, avoiding the combinatorial explosion of a direct monomial representation.
- **Use of a cohomological proxy:** An approach has been designed that uses $\mathcal{C}(\mathcal{K})$ and its cohomology $\mathcal{H}^*(\mathcal{C}(\mathcal{K}))$ as an indirect way to work with $\mathcal{H}^*(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$. Through the combined use of Algebraic Topological models (AT-models) and the de Rham integration and Whitney map quasi-isomorphisms, the problem of computing $\mathcal{H}^*(\mathcal{A}_{\mathcal{PL}}(\mathcal{K}))$ is avoided, allowing generators to be lifted and injectivity/surjectivity to be verified effectively.
- **Implementation of Dupont's contraction:** As the most prominent contribution, the problem of finding primitives (the main bottleneck in step 7 of the original algorithm) has been solved through the concrete implementation of Dupont's simplicial contraction. Unlike alternative approaches based on matrix truncation or combinatorial models, Dupont's contraction operates locally, simplex by simplex. This guarantees the patch conditions by definition and reduces the computational complexity respect to other approaches. Also, the implementation of Dupon't formula at simplicial cochain complex level has been implemented. This opens the door to explore the algebraic transferred structured from $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ to $C(\mathcal{K})$ from a computational point of view.

Beyond the theoretical formulation, this work has materialized these ideas into a concrete software implementation within a computer algebra system. To the best of our knowledge, this constitutes the first effective implementation of Dupont's contraction applied to arbitrary simplicial sets in this context.

A collection of illustrative examples has been presented, highlighting the computation of the minimal model of the $\mathcal{K}3$ surface. It is a classical result from Deligne et al. that this space is formal, and we use it as a non-trivial example to verificate this property.

As lines of future work, this study opens up several interesting avenues. The first step involves thoroughly studying the Kenzo program to compare both approaches and determine if they could be combined. In addition to that, from a computational perspective, a benchmark will be designed in order to assess and compare the running times of the three methods presented in this work to obtain a primitive of an exact element of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$. Also, it would be valuable to further optimize the underlying data structures to handle simplicial complexes of very high dimensions or with a massive number of simplices, perhaps by leveraging parallel computing techniques. From a theoretical point of view, the connection established in this work with Finite Element Exterior Calculus (FEEC) suggests that the techniques developed here (in particular, the search for primitives without linear systems) could be exported to solve analogous problems in other contexts of computational geometry and numerical analysis. Furthermore, the question raised in Chapter II regarding the possible generalization of Dupont's product formula remains open, which would pave the way for computing Sullivan models and homotopy groups for large datasets.

Appendix



The Sullivan Models package

1 The cochain algebra $(\mathcal{A}_{\mathcal{PL}})_n$ 75
2 The simplicial cochain algebra $\mathcal{A}_{\mathcal{PL}}$ 87
3 Discrete Hodge
  3.1 The Algebraic Topological Model 93
  3.2 The Combinatorial Laplacian and the Discrete Hodge Decomposition 94
  3.3 Relation between the AT-model and the combinatorial laplacian 95
4 The CDG Algebra $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$ 103
5 Bibliography 117
6 Python Module Index 118

CHAPTER

ONE

THE COCHAIN ALGEBRA $(\mathcal{A}_{\mathcal{PL}})_n$

First, consider the free graded commutative algebra $\bigwedge(t_0, \ldots, t_n, y_0, \ldots, y_n)$, where the generators t_i have degree 0, and the generators y_j have degree 1. The differential d is given by

$$\begin{aligned} d(t_i) &= y_i, \\ d(y_j) &= 0. \end{aligned}$$

The cochain algebra $(A_{PL})_n$ is defined as the quotient:

$$(A_{PL})_n := \frac{\bigwedge(t_0, \ldots, t_n, y_0, \ldots, y_n)}{\langle \sum t_i - 1, \sum y_j \rangle}.$$

The implementation of this algebra has been made taking into account that this algebra is equivalent to the exterior algebra $Ext_y = \frac{\bigwedge(y_0,\ldots,y_n)}{\langle \sum y_j \rangle}$ with coefficients in the polynomial ring $R_t := \frac{\mathbb{Q}[t_0,\ldots,t_n]}{\langle \sum t_i - 1 \rangle}$:

$$(A_{PL})_n := R_t \otimes Ext_y.$$

For that reason, R_t is considered the base_ring of $(A_{PL})_n$.

Note: Notice that this algebra is actually isomorphic to $\bigwedge(t_1, \ldots, t_n, y_1, \ldots, y_n)$. That is, the relations allow us to express t_0 in terms of $t_1, \ldots, t_n$ and y_0 in terms of $y_1, \ldots, y_n$. After eliminating t_0, y_0, the expression of each element is unique. However, it is sometimes more convenient to use also t_0, y_0 to write certain formulas, so we will use one expression or the other depending on the context.

AUTHORS:

- Carlos Alquézar (20260601): Beta version released

class AplN(*args: Any**kwargs: Any)

Bases: `ExteriorAlgebra`, `UniqueRepresentation`, `CachedRepresentation`

The Algebra of polynomial differential forms.

INPUT:

- `n` – dimension
- `labels` – (optional, default: `t_i, y_i`); labels of the generators

tab-content EXAMPLES:

```
sage: A = AplN(5)
sage: A
The Commutative Differential Graded Algebra (A_{PL})_{5} defined as
```

(continues on next page)

(continued from previous page)

```
the  tensor product of the Polynomial Ring QQ<t_{0},...,t_{5}> with
the Exterior Algebra generated by QQ<y_{0},...,y_{5}>.

The degrees of the generators are:

    deg(t_i)=0
    deg(y_j)=1

The differential 'd' is defined as:

    d(t_i)=y_i
    d(y_j)=0

sage: A.inject_variables()
Defining y0, y1, y2, y3, y4, y5
sage: R = A.base_ring()
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4, t5
sage: p = t1*y2*y3 + t0**2*y4*y5
sage: p
t1*y2*y3 + t0^2*y4*y5
```

tab-content EXAMPLES:

```
A = AplN(5)
A.inject_variables()
R = A.base_ring()
R.inject_variables()
p = t1*y2*y3 + t0**2*y4*y5

labs = ('t2','t5','t6','y2','y5','y6')
A = AplN(n=2,labels=labs)
R = A.base_ring()
A.inject_variables()
R.inject_variables()
```

extended_basis(deg_ydeg_t=1cum=False)

Considering `self` as a graded module, this method returns the module basis of degree `deg_y` with polynomial coefficients truncated by degree `deg_t`.

INPUT:

- `deg_y` – integer ; degree of the graded module of `self`.
- `deg_t` – integer (default: *1*); coefficient's degree
  - `cum` – boolean (default: *False*); If *True*, the basis takes into account polynomial coefficients up to degree `deg_t`. Otherwise, the method returns the basis with homogeneous coefficients of degree `deg_t`.

OUTPUT: tuple of monomials of `self`; module generators of exterior degree `deg_y`.

tab-content EXAMPLES:

```
sage: A = AplN(2)
sage: A.extended_basis(deg_y=1,deg_t=1,cum=False)
[t0*y0, t0*y1, t0*y2, t1*y0, t1*y1, t1*y2, t2*y0, t2*y1, t2*y2]
sage: A.extended_basis(deg_y=1,deg_t=1,cum=True)
[t0*y0, t0*y1, t0*y2, t1*y0, t1*y1, t1*y2, t2*y0, t2*y1, t2*y2, y0, y1, y2]
```

tab-content EXAMPLES:

```
A = AplN(2)
A.extended_basis(deg_y=1,deg_t=1,cum=False)
A.extended_basis(deg_y=1,deg_t=1,cum=True)
```

extended_graded_module(deg_ydeg_t=1cum=False)

Return the extended graded module of `self` of degree `deg_y` with coefficients in the polynomial ring $\mathbb{Q}[t_0, \ldots, t_n]$.

INPUT:

- `deg_y` – integer; degree of the graded module of `self`.
- `deg_t` – integer (default: *1*); coefficient's degree.
- `cum` – boolean (default: *False*); If *True*, the basis takes into account

polynomial coefficients up to degree `deg_t`. Otherwise, the method returns the basis with homogeneous coefficients of degree `deg_t`.

OUTPUT: a CombinatorialFreeModule with coefficient in the base_ring of `self`.

tab-content EXAMPLES:

```
sage: A = AplN(2)
sage: A.extended_graded_module(deg_y=1,deg_t=1,cum=True)
Free module generated by {'t0y0', 't0y1', 't0y2', 't1y0', 't1y1', 't1y2',
↪'t2y0', 't2y1', 't2y2', 'y0',
'y1', 'y2'} over Multivariate Polynomial Ring in t0, t1, t2 over Rational␣
↪Field
```

graded_module_endomorphism(yideg)

Return the reduction matrix of the variable `yi` as a linear endomorphism of the graded vector space of degree `deg`.

INPUT:

- `yi` – integer;
- `deg` – integer;

OUTPUT: sparse matrix;

tab-content EXAMPLES:

```
sage: A = AplN(3)
sage: list(A.basis(2))
[y0*y1, y0*y2, y0*y3, y1*y2, y1*y3, y2*y3]
sage: list(A.reduced_exterior_basis(yi=0, deg=2))
[y1*y2 + y1*y3, -y1*y2 + y2*y3, -y1*y3 - y2*y3, y1*y2, y1*y3, y2*y3]
sage: A._graded_face_matrix(yi=0,deg=2)
```

(continues on next page)

(continued from previous page)

```
[ 1 -1  0  1  0  0]
[ 1  0 -1  0  1  0]
[ 0  1 -1  0  0  1]
```

reduced_exterior_basis(yideg)

Returns the exterior basis of the homogeneous graded module of degree `deg` after applying the relation $y_i = -\sum_{j \neq i} y_j$.

INPUT:

- `yi` – integer; the index of the exterior algebra generator
- `deg` – integer; the degree of the graded module

OUTPUT: tuple of nonnegative integers;

tab-content EXAMPLES:

```
sage: A = AplN(4)
sage: list(A.basis(1))
[y0, y1, y2, y3, y4]
sage: A.reduced_exterior_basis(yi=2, deg=1)
(y0, y1, -y0 - y1 - y3 - y4, y3, y4)
```

whitney_form()

Returns the *elementary volume form* (also called *constant volume form*) of `self`. The Whitney volume form on $(A_{PL})_n$ if a differential form of degree *n* defined as:

$$\omega_n = n! \cdot \sum_{k=0}^{n} (-1)^k t_0 \cdot y_0 \wedge \cdots \wedge \hat{y}_k \wedge \cdots \wedge y_n$$

OUTPUT: a differential form of degree *n*;

tab-content EXAMPLES:

```
sage: A = AplN(4)
sage: A.whitney_form()
24*t4*y0*y1*y2*y3 - 24*t3*y0*y1*y2*y4 + 24*t2*y0*y1*y3*y4 -␣
↪24*t1*y0*y2*y3*y4 + 24*t0*y1*y2*y3*y4
sage: _.reduce(0)
24*y1*y2*y3*y4
```

class Element(*args: Any**kwargs: Any)

Bases: `Element`

A polynomial differential form.

differential()

Return the differential of *self*.

For the q–form $\omega \in (A_{PL})^q_n$:

$$\omega = p(\bar{t})\ y_{j_0} \wedge \cdots \wedge y_{j_{q-1}}$$

the differential $d : (A_{PL})^q_n \to (A_{PL})^{q+1}_n$ is defined as

$$d(\omega) = \sum_{j=0}^{n} \frac{\partial p(\bar{t})}{\partial t_j} y_j \wedge y_{i_0} \wedge \cdots \wedge y_{j_{q-1}}$$

where $p(\bar{t}) \in \mathbb{Q}[t_0, \ldots, t_n]$.

Let $\omega, \theta \in (A_{PL})_n$. Then, the differential has the properties:

- $d \circ d = 0$,
- $d(\omega \wedge \theta) = d(\omega) \wedge \theta + (-1)^{|\omega|}\omega \wedge d(\theta)$.

OUTPUT: a differential $(q+1)$ –form.

tab-content EXAMPLES:

```
sage: A = AplN(5)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining y0, y1, y2, y3, y4, y5
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4, t5
sage: p = t0*t2*y2*y3
sage: p.differential()
t2*y0*y2*y3

sage: A = AplN(5)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining y0, y1, y2, y3, y4, y5
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4, t5
sage: w = (t1*t2+t3**2)*y2*y3
sage: theta = t0*y4
sage: w.degree()
2
sage: w.differential()*theta+((-1)**2)*w*theta.differential() ==␣
↪(w*theta).differential()
True
sage: w.differential()
t2*y1*y2*y3
sage: w.differential().differential()
0
```

tab-content EXAMPLES:

```
A = AplN(5)
R = A.base_ring()
A.inject_variables()
R.inject_variables()
p = t0*t2*y2*y3
p.differential()

A = AplN(5)
R = A.base_ring()
A.inject_variables()
R.inject_variables()
w = (t1*t2+t3**2)*y2*y3
theta = t0*y4
w.degree()
```

(continues on next page)

(continued from previous page)

```
w.differential()*theta+((-1)**2)*w*theta.differential() == (w*theta).
↪differential()
w.differential()
w.differential().differential()
```

dif()
Alias for :meth`self.differential`

dup(i)
Alias for *dupont_operator()*

dupont_operator(i)
This method implements the Dupont's operator $h_n^i : (A_{PL})_n^* \to (A_{PL})_n^{*-1}$ for the standard Δ^nsimplex as it is described in [CG08] (see also [Dup76]).

Let $\mathbb{K}$ be a field of characteristic 0 and let Δ^n be the standard $n$simplex. Also, let

$$A_{PL}(\Delta^n) := \{A_{PL}^p(\Delta^n)\}_{p\geq 0},$$

be the algebra of polynomial differential forms defined on Δ^n, where

$$A_{PL}^p(\Delta^n) := \{\sigma : \Delta^n \to (A_{PL})_n^p\}$$

Notice that we can consider Δ^n as a set with only one element so, in this case

$$A_{PL}(\Delta^n) \cong (A_{PL})_n.$$

In fact, the motivation for the construction of $(A_{PL})_n$ was precisely to model locally the algebra of polynomial differential forms on simplices and, at the same time, construct a global structure satisfying the geometrical, topological and algebraic constraints. For the same reason, being $C^*(\Delta^n)$ the simplicial cochain complex of Δ^n, we have:

$$C^p(\Delta^n) = \begin{cases} \mathbb{K}, & if \;\; p = n \\ 0, & otherwise \end{cases}$$

Let $\int_n$ be the de Rham map (see [Dup76, FelixHT01, Whi57]):

$$\int_n : (A_{PL})_n \to \mathbb{K}$$

For a subset $I = \{i_0, \ldots, i_p\} \subseteq \{0, \ldots, n\}$, with $0 \leq i_0 < \cdots < i_p \leq n$, the Whitney map (see [LK21, Whi57])

$$\mathcal{W}_n^I : \mathbb{K} \to (A_{PL})_n^p$$

is defined as:

$$\mathcal{W}_n^I(c) = c \cdot \sum_{j=0}^{p} (-1)^j t_{i_j} y_{i_0} \wedge \cdots \wedge \hat{y_{i_j}} \wedge \cdots \wedge y_{i_p}$$

where $\hat{}$ means the term is omitted. The elements in the image of $\mathcal{W}$ are called the *elementary differential forms* or just *Whitney forms*.

For a generic simplicial set S, and following the work of Hassler *Whitney* (see [Whi57]), *Dupont* (see [Dup76]) constructed an explicit simplicial contraction of the commutative differential graded

algebra of polynomial differential forms $A_{PL}(S)$ to $C^*(S)$ compatible with the intrinsic affine structure of the polynomial differential forms (affine relations) and also with the wedge product of forms (up to homotopy).

This method implements step by step the description of the contraction given by Cheng et al. (see [CG08]) for the case of Δ^n (for the case of a general simplicial set go to *dupont_contraction()*), with minor notation adaptations for the sake of maintain coherence with the rest of the code.

The description given by Cheng et al. is as follows:

Let $\omega \in (A_{PL})_n$. For each vertex e_i of the $n$simplex Δ^n, define the dilation map

$$\phi_i : [0,1] \times \Delta^n \to \Delta^n$$

by the formula

$$\phi_i(u, t_0, \ldots, t_n) = ((1-u)t_0, \ldots, (1-u)t_i + u, \ldots, (1-u)t_n).$$

Let

$$(\pi_i)_* : A^*_{PL}([0,1] \times \Delta^n) \to (A_{PL})^{*-1}_n$$

be integration over the first factor.

The operator h^i_n:

$$h^i_n : (A_{PL})^*_n \to (A_{PL})^{*-1}_n$$

is defined by the formula

$$h^i_n \omega = (\pi_i)_* \phi_i^* \omega.$$

where $\phi_i^*\omega$ is the pullback of ω respect to ϕ_i.

Let

$$\epsilon^i_n : A^*_{PL}(\Delta^n) \to \mathbb{K}$$

be evaluation at the vertex e_i. The Poincar'e Lemma states that

$$\mathbb{1}_A - \epsilon^i_n = d \circ h^i_n + h^i_n \circ d.$$

See also:

`_dupont_operator()`

integral()

INPUT:

- `p` – an element of `self`.
- `i` – integer; the vertex index to apply the operator

OUTPUT: an element of `self`;

tab-content EXAMPLES:

```
sage: A = AplN(4)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining y0, y1, y2, y3, y4
```

(continues on next page)

(continued from previous page)

```
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4
sage: w = t0*y1*y3
sage: w.dup(0).dif()
-1/3*t3*y0*y1 + 1/3*t1*y0*y3 + (2/3*t0 + 1/3)*y1*y3
sage: w.dif().dup(0)
1/3*t3*y0*y1 - 1/3*t1*y0*y3 + (1/3*t0 - 1/3)*y1*y3
sage: w.dup(0).dif()+w.dif().dup(0) == w
True
```

homogeneous_components()

Return the homogeneous components of `self`.

OUTPUT: a dictionary indexed by degree.

tab-content EXAMPLES:

```
sage: A = AplN(5)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining y0, y1, y2, y3, y4, y5
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4, t5
sage: p = t0*t2*y2*y3
sage: p = t0*t2*y2*y3 + y1
sage: p.homogeneous_components()
{1: y1, 2: t0*t2*y2*y3}
```

integral(yi=0)

Returns the integral of `self`, defined as follows:

Let $\int_n$ be the linear map:

$$\int_n : (A_{PL})^n_n \to \mathbb{Q}$$

by setting This map represents the integration $\int_{\Delta^n} \omega$ of an $n$form $\omega \in (A_{PL})^n_n$ over the standard $n$simplex Δ^n. For homogeneous elements $\theta \in A_{PL}(p_n)$ with $n \neq p$, the integral is set to zero. Then, the map is extended to all (A_{PL}) by linearity.

INPUT:

- `yi` – integer (optional, default: *0*); variable to reduce

`self` before integration

OUTPUT: a rational number;

tab-content EXAMPLES:

The integral of the Whitney form:

```
sage: A = AplN(4)
sage: w = A.whitney_form()
sage: w
24*t4*y0*y1*y2*y3 - 24*t3*y0*y1*y2*y4 + 24*t2*y0*y1*y3*y4 -␣
↪24*t1*y0*y2*y3*y4 + 24*t0*y1*y2*y3*y4
```

(continues on next page)

(continued from previous page)

```
sage: w.integral()
1
```

The integral of a generic form of maximum degree:

```
sage: A = AplN(4)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining y0, y1, y2, y3, y4
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4
sage: theta = (t1**2-1)*A.whitney_form()
sage: theta
(24*t1^2*t4 - 24*t4)*y0*y1*y2*y3 + (-24*t1^2*t3 + 24*t3)*y0*y1*y2*y4 +␣
↪(24*t1^2*t2 - 24*t2)*y0*y1*y3*y4 + (-24*t1^3 + 24*t1)*y0*y2*y3*y4 +␣
↪(24*t0*t1^2 - 24*t0)*y1*y2*y3*y4
sage: theta.reduce(0)
(24*t1^2 - 24)*y1*y2*y3*y4
sage: theta.integral()
-14/15
sage: (factorial(4)*2)/factorial(2+4)
1/15
sage: (factorial(4)*2)/factorial(2+4)-1
-14/15
```

The integral of an element of degree lower than the dimension of `self.parent()`:

```
sage: A = AplN(4)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining y0, y1, y2, y3, y4
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4
sage: theta = t1*y0
sage: theta.integral()
0
```

The integral of a nonhomogeneous element:

```
sage: A = AplN(4)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining y0, y1, y2, y3, y4
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4
sage: theta = t1*y0 + y1*y2
sage: theta.integral()
0
sage: theta = t1*y0 + y1*y2 + A.whitney_form()
sage: theta.integral()
1
```

koszul()

Return the koszul differential of *self*.

For the q–form $\omega \in (A_{PL})_n^q$:

$$\omega = p(\bar{t})\ y_{j_0} \wedge \cdots \wedge y_{j_{q-1}}$$

the koszul differential $\kappa : (A_{PL})_n^q \to (A_{PL})_n^{q-1}$ is defined as

$$\kappa(\omega) = p(\bar{t}) \sum_{j=0}^{q-1} (-1)^j t_{i_j} y_{i_0} \wedge \cdots \wedge \hat{y_{i_j}} \wedge \cdots \wedge y_{j_{q-1}}$$

where $p(\bar{t}) \in \mathbb{Q}[t_0, \ldots, t_n]$.

Let $\omega, \theta \in (A_{PL})_n$, and ω defined as before such that $q = |\omega|$, and $r = |p(\bar{t})|$. Then, the koszul differential has the properties:

- $\kappa \circ \kappa = 0$,
- $(\kappa \circ d)(\omega) + (d \circ \kappa)(\omega) = (r+q)\omega$,
- $\kappa(\omega \wedge \theta) = \kappa(\omega) \wedge \theta + (-1)^{|\omega|}\omega \wedge \kappa(\theta)$.

OUTPUT: a differential $(q-1)$ –form.

tab-content EXAMPLES:

This example illustrates the relation between the cochain differential and koszul differential:

```
sage: A = AplN(3)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining
y0, y1, y2, y3
sage: R.inject_variables()
Defining
t0, t1, t2, t3
sage: p = t0 * t2 * y2 * y3
sage: p
t0*t2*y2*y3
sage: p.koszul().differential()+p.differential().koszul()
4*t0*t2*y2*y3
sage: (p.differential().koszul()+p.koszul().differential()).reduce(0) ==␣
↪(2+2)*p.reduce(0)
True
```

This example illustrates the koszul differential of a wedge of forms:

```
sage: A = AplN(4)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining y0, y1, y2, y3, y4
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4
sage: p = t1*y2
sage: q = t2*y3*y4
sage: print(p.koszul()*q+((-1)**p.degree())*p*q.koszul().reduce(0) ==␣
↪(p*q).koszul().reduce(0))
True
```

(continues on next page)

(continued from previous page)

```
sage: print(p.koszul().differential()+ p.differential().koszul() ==␣
↪(1+1)*p)
True
sage: print(q.koszul().differential()+ q.differential().koszul() ==␣
↪(1+2)*q)
True
```

reduce(var_idxlabel_ref=False)

Return the reduction of `self` by the relations:

- $t_i = 1 - \sum_{j \neq i} t_j$,
- $y_i = -\sum_{j \neq i} y_j$.

INPUT:

- `var_idx` – (integer or str). The index or the label of the variable to be reduced.
- `label_ref` – boolean (default: False); If `True`, `var_idx` is considered as a string.

OUTPUT: an element of `self.parent()`

tab-content EXAMPLES:

```
sage: A = AplN(5)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining y0, y1, y2, y3, y4, y5
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4, t5
sage: theta = t0*y4
sage: theta.reduce(0)
(-t1 - t2 - t3 - t4 - t5 + 1)*y4
sage: theta = t1*y0
sage: theta.reduce(0)
-t1*y1 - t1*y2 - t1*y3 - t1*y4 - t1*y5
```

CHAPTER

TWO

THE SIMPLICIAL COCHAIN ALGEBRA $\mathcal{A}_{\mathcal{PL}}$

We define now a simplicial object in the category of cochain algebras, called the **simplicial cochain algebra** $A_{PL} = \{(A_{PL})_n\}_{n\geq 0}$ as follows:

- For each $n \geq 0$, the cochain algebra $(A_{PL})_n$ is the one defined above.
- The face and degeneration morphisms are the unique cochain algebra morphisms

$$\begin{array}{ll} \partial_i : & (A_{PL})_n \to (A_{PL})_{n-1}, \quad 0 \leq i \leq n, \\ s_j : & (A_{PL})_n \to (A_{PL})_{n+1}, \quad 0 \leq j \leq n, \end{array}$$

satisfying:

$$\partial_i(t_k) = \begin{cases} t_k, & k < i \\ 0, & k = i \\ t_{k-1}, & k > i \end{cases}, \qquad s_j(t_k) = \begin{cases} t_k, & k < j \\ t_k + t_{k+1}, & k = j \\ t_{k+1}, & k > j \end{cases}$$

The structure of the simplicial cochain algebra A_{PL} can be visualized in figure simplicial_apln, where horizontally is represented the simplicial structure for a given dimension n (subscript), and vertically the graded structure of the algebra (representing the degree p, with superscript). This object has a triangular structure due to the vanishing of the elements when $p > n$.

$$\begin{array}{ccccccccccc}
0 & \xleftarrow{\partial_i} & (A_{PL})^0_0 & \xleftarrow{\partial_i} & (A_{PL})^0_1 & \xleftarrow{\partial_i} & (A_{PL})^0_2 & \xleftarrow{\partial_i} & (A_{PL})^0_3 & \xleftarrow{\partial_i} & \cdots \\
 & & \downarrow{\scriptstyle d^0_A} & & \downarrow{\scriptstyle d^0_A} & & \downarrow{\scriptstyle d^0_A} & & \downarrow{\scriptstyle d^0_A} & & \\
 & & 0 & \xleftarrow{\partial_i} & (A_{PL})^1_1 & \xleftarrow{\partial_i} & (A_{PL})^1_2 & \xleftarrow{\partial_i} & (A_{PL})^1_3 & \xleftarrow{\partial_i} & \cdots \\
 & & & & \downarrow{\scriptstyle d^1_A} & & \downarrow{\scriptstyle d^1_A} & & \downarrow{\scriptstyle d^1_A} & & \\
 & & & & 0 & \xleftarrow{\partial_i} & (A_{PL})^2_2 & \xleftarrow{\partial_i} & (A_{PL})^2_3 & \xleftarrow{\partial_i} & \cdots \\
 & & & & & & \downarrow{\scriptstyle d^2_A} & & \downarrow{\scriptstyle d^2_A} & & \\
 & & & & & & 0 & \xleftarrow{\partial_i} & (A_{PL})^3_3 & \xleftarrow{\partial_i} & \cdots
\end{array}$$

Fig. 1: The structure of the simplicial cochain algebra $(A_{PL})_n$.

class SimplicialAPL(*args: Any**kwargs: Any)

Bases: UniqueRepresentation, CachedRepresentation

The simplicial cochain algebra $A_{PL} = \{(A_{PL})_n\}_{n\geq 0}$.

affine_map(facesdimcoef=1normalized=True)

Returns the differential form on $\omega \in (A_{PL})_{dim}$ of degree `n = dimlen(faces)` for which

$$\partial_J(\omega) = \frac{coef}{n!} * \mathcal{W}\langle\rangle\sqcup\backslash\rceil\dagger$$

with the facepath J specified in `faces`.

If `normalized=True`, then $\int \partial_J(\omega) = 1$.

Note: If `coef= factorial(n)`, this map is equivalent to *whitney_map()*.

INPUT:

- `faces` – tuple; the face path specified by an strictly increasing sequence of nonnegative integers
- `dim` – integer; codomain's dimension
- `coef` – an element of the domain's base_ring (default:1)
- `normalized` – boolean; if `True`, returns the normalized differential form

tab-content EXAMPLES:

```
sage: S = SimplicialAPL(5)
sage: A = S.n_th_object(4)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining y0, y1, y2, y3, y4
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4
sage: S.affine_map(faces=(1,),dim=5,coef=factorial(4)) == S.whitney_map((1,),
↪5)
True
sage: w = S.affine_map(faces=(1,),dim=5,coef=factorial(4))
sage: [S.face(w,i).reduce(0) for i in range(6)]
[0, 24*y1*y2*y3*y4, 0, 0, 0, 0]

sage: coef = t0*t1
sage: theta = S.affine_map((4,),5,coef)
sage: [S.face(theta,i).reduce(0) for i in range(6)]
[0, 0, 0, 0, (-t1^2 - t1*t2 - t1*t3 - t1*t4 + t1)*y1*y2*y3*y4, 0]
sage: S.face(theta,4).integral()
1/720

sage: eta = S.affine_map((4,),5,coef, normalized=True)
sage: S.face(eta,4).integral()
1
```

degeneracy(pi)

Returns the degeneracy `i` of the differential form `p`

tab-content EXAMPLES:

```
sage: from Sullivan_MM import *
sage: S = SimplicialAPL(10)
sage: A = AplN(4)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining y0, y1, y2, y3, y4
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4
sage: theta = t1*y0
sage: S.degeneracy(theta,0)
t2*y0 + t2*y1

sage: theta = A(t4)
sage: S.degeneracy(theta,4)
t4 + t5

sage: theta = y4
sage: S.degeneracy(theta,4)
y4 + y5

sage: theta = t0*y3*y4
sage: S.degeneracy(theta,4)
t0*y3*y4 + t0*y3*y5
sage: _ in AplN(5)
True
```

face(pi)

Returns the face `i` of the differential form `p`

tab-content EXAMPLES:

```
sage: S = SimplicialAPL(10)
sage: A = S.n_th_object(4)
sage: R = A.base_ring()
sage: A.inject_variables()
Defining y0, y1, y2, y3, y4
sage: R.inject_variables()
Defining t0, t1, t2, t3, t4
sage: w = t0*y1*y3
sage: S.face(w,0)
0
sage: S.face(w,2)
t0*y1*y2
sage: _ in S.n_th_object(3)
True
sage: S.face(w,4)
t0*y1*y3
sage: _ in S.n_th_object(3)
True
```

n_th_object(n)

Returns the nsimplex of `self`.

tab-content EXAMPLES:

```
sage: S = SimplicialAPL(10)
sage: S.n_th_object(5)
The Commutative Differential Graded Algebra (A_{PL})_{5} defined as
the  tensor product of the Polynomial Ring QQ<t_{0},...,t_{5}> with
the Exterior Algebra generated by QQ<y_{0},...,y_{5}>.

The degrees of the generators are:

    deg(t_i)=0
    deg(y_j)=1

The differential 'd' is defined as:

    d(t_i)=y_i
    d(y_j)=0
```

tab-content EXAMPLES:

```
S = SimplicialAPL(10)
S.n_th_object(5)
```

whitney_map(facesdim)

Let $\mathbb{K}$ be a field of characteristic 0 and let $S = \{S_n\}_{n\geq 0}$ be a simplicial set. Also, let $A_{PL}(S)$ be the algebra of polynomial differential forms defined on S, and $C^*(S)$, the simplicial cochain complex associated to S.

Let $I = \{i_0, \ldots, i_p\}$, with $0 \leq i_0 < \cdots < i_p \leq n$, and define

$$J := [n] \setminus I = \{j_0, \ldots, j_q\}$$

such that $0 \leq j_0 < \cdots < j_q \leq n$ and $p + q = n$.

Let $\sigma_p \in S_p$ and $\sigma_n \in S_n$ be two simplices of S such that, $\partial_J(\sigma_n) = \sigma_p$ where

$$\partial_J = \partial_{j_0} \circ \cdots \circ \partial_{j_q}.$$

Let $\beta \in C^p(S)$. The Whitney map (see [LK21, Whi57])

$$\mathcal{W}^p = \{\mathcal{W}^p_n : C^p(S_p) \to A^p_{PL}(S_n)\}$$

is defined as:

$$\mathcal{W}^p_n(\beta)(\sigma_n)|_{\sigma_p} = \beta(\sigma_p) \cdot n! \cdot \sum_{k=0}^{p} (-1)^j t_{i_k} y_{i_0} \wedge \cdots \wedge \hat{y}_{i_k} \wedge \cdots \wedge y_{i_p}$$

where the symbol $\hat{}$ means the term is omitted. The elements in the image of $\mathcal{W}$ are called the *elementary differential forms* or just *Whitney forms*. The motivation behind this notation is that the subset I comes from the inclusion

$$i : \Delta^p \hookrightarrow \Delta^n$$

that, by the A_{PL} functor translates to the inclusion of algebras:

$$\begin{array}{rccc} A_{PL}(i): & (A_{PL})_p & \hookrightarrow & (A_{PL})_n \\ & (t_0, \ldots, t_p) & \to & (t_{i_0}, \ldots, t_{i_p}) \end{array}$$

tab-content EXAMPLES:

This example ilustrates the locallity property of Whitney forms:

```
sage: S = SimplicialAPL(10)
sage: A3 = S.n_th_object(3)
sage: w1 = S.whitney_map((1,), 4)
sage: w1
-6*t4*y0*y2*y3 + 6*t3*y0*y2*y4 - 6*t2*y0*y3*y4 + 6*t0*y2*y3*y4
sage: S.face(w1,0)
0
sage: S.face(w1,1)
-6*t3*y0*y1*y2 + 6*t2*y0*y1*y3 - 6*t1*y0*y2*y3 + 6*t0*y1*y2*y3
sage: S.face(w1,2)
0
sage: S.face(w1,3)
0
sage: S.face(w1,4)
0
sage: S.face(w1,1) == A3.whitney_form()
True
```

This example ilustrates the Whitney map along a facepath of length 2:

```
sage: w23 = S.whitney_map((2,3),5)
sage: w23
-6*t5*y0*y1*y4 + 6*t4*y0*y1*y5 - 6*t1*y0*y4*y5 + 6*t0*y1*y4*y5
sage: S.face(S.face(w23,3),2) == A3.whitney_form()
True
```

CHAPTER

THREE

DISCRETE HODGE

The Combinatorial Laplacian and the Discrete Hodge Decomposition for simplicial complexes, simplicial sets, Delta complexes and Cubical complexes.

3.1 The Algebraic Topological Model

For such objects, this implementation uses the method `algebraic_topological_model` available at SAGEMATH. The specification of the method for the case of a finite simplicial complex (see SS, DC and CC for the implementation in other simplicial structures) is the following:

> Algebraic topological model for this simplicial complex with coefficients in `base_ring`.
>
> The term "algebraic topological model" is defined by Pilarczyk and Réal [PR15].
>
> INPUT:
>
> - `base_ring` – coefficient ring (default: `QQ`); must be a field
>
> Denote by C the chain complex associated to this simplicial complex. The algebraic topological model is a chain complex M with zero differential, with the same homology as C, along with chain maps $\pi : C \to M$ and $\iota : M \to C$ satisfying $\iota \circ \pi = 1_M$ and $\pi \circ \iota$ chain homotopic to 1_C. The chain homotopy ϕ must satisfy
>
> - $\phi \circ \phi = 0$,
> - $\pi \circ \phi = 0$,
> - $\phi \circ \iota = 0$.
>
> Such a chain homotopy is called a *chain contraction*.
>
> OUTPUT: a pair consisting of
>
> - chain contraction ϕ associated to C, M, π, and ι
> - the chain complex M
>
> Note that from the chain contraction ϕ, one can recover the chain maps π and ι via `phi.pi()` and `phi.iota()`. Then one can recover C and M from, for example, `phi.pi().domain()` and `phi.pi().codomain()`, respectively.

An alternative definition for cochain complexes is given in [CG08]:

> Given two cochain complexes C and M, a contraction (π, ι, ϕ) from C to M consists of chain maps:
>
> $$\begin{aligned} \pi : &\quad C \to M, \\ \iota : &\quad M \to C, \end{aligned}$$

and a chain homotopy

$$\phi : C^* \to C^{*-1},$$

such that

$$\pi \circ \iota = 1_M$$

and

$$\iota \circ \pi - 1_C = d_C \circ \phi + \phi \circ d_C.$$

Such a chain homotopy provides a strong relation between the chain complexes C and M; for example, their homology groups are isomorphic.

3.2 The Combinatorial Laplacian and the Discrete Hodge Decomposition

This definitions come from [Dod76a, Gol02].

Let K be a finite oriented simplicial complex, $d \geq 0$ be an integer, and $\partial_d : C_d(K) \to C_{d-1}(K)$, the *boundary operator*. Then,

Definition: The dth *combinatorial Laplacian* is the linear operator $\Delta_d : C_d \longrightarrow C_d$ given by

$$\Delta_d = \partial_{d+1} \circ \partial_{d+1}^* + \partial_d^* \circ \partial_d.$$

For convenience, we use the notations

$$\Delta_d^{\text{up}} = \partial_{d+1} \circ \partial_{d+1}^*$$

and

$$\Delta_d^{\text{down}} = \partial_d^* \circ \partial_d,$$

so that

$$\Delta_d = \Delta_d^{\text{up}} + \Delta_d^{\text{down}}.$$

Now, let B_d be the adjacenty matrix of dimension d defined by the simplicial structure of K.

The dth *Laplacian matrix* of K, denoted L_d, relative to some orderings of the standard bases for C_d and C_{d-1} of K, is the matrix representation of Δ_d.

Observe that

$$L_d = B_{d+1} B_{d+1}^T + B_d^T B_d.$$

As above, for convenience, we use the notations

$$L_d^{\text{up}} = B_{d+1} B_{d+1}^T$$

and

$$L_d^{\text{down}} = B_d^T B_d,$$

so that

$$L_d = L_d^{\text{up}} + L_d^{\text{down}}.$$

3.3 Relation between the *ATmodel* and the combinatorial laplacian

Let

$$\begin{array}{rcl} H_d & := & \ker\left(L_d^{up}\right) \cap \ker\left(L_d^{down}\right) \\ E_d & := & \ker\left(L_d^{up}\right) \cap \mathrm{Im}(L_d^{down}) \\ C_d & := & \mathrm{Im}(L_d^{up}) \cap \ker\left(L_d^{down}\right) \end{array}$$

It is easy to see that $H_d = \ker L_d$. The fundamental theo of discrete Hodge theory states:

> For any finite simplicial complex K, we have the orthogonal decomposition:
>
> $$C^k(K;\mathbb{K}) = H_d \oplus E_d \oplus C_d$$
>
> and
>
> $$\begin{array}{rcl} H_d & \perp & E_d \\ E_d & \perp & C_d \\ C_d & \perp & H_d \end{array}$$
>
> Furthermore, $\ker\left(L_k\right) \cong H^k(K;\mathbb{K})$ via the natural projection.

Taking into account the *ATmodel*:

> **Theorem:**
>
> $$\begin{array}{rcl} H_d & \cong & \mathrm{Im}(\iota \circ \pi) \\ E_d & \cong & \mathrm{Im}(d_C \circ \phi) \\ C_d & \cong & \mathrm{Im}(\phi \circ d_C) \end{array}$$
>
> and
>
> $$\begin{array}{rcl} \mathrm{Im}(\iota \circ \pi) & \perp & \mathrm{Im}(d_C \circ \phi) \\ \mathrm{Im}(d_C \circ \phi) & \perp & \mathrm{Im}(\phi \circ d_C) \\ \mathrm{Im}(\phi \circ d_C) & \perp & \mathrm{Im}(\iota \circ \pi) \end{array}$$
>
> and
>
> $$C^k(K) = \mathrm{Im}(\iota \circ \pi) \oplus \mathrm{Im}(d_C \circ \phi) \oplus \mathrm{Im}(\phi \circ d_C)$$

See also:

- Finite simplicial complexes: SC.
- Simplicial sets: SS.
- Finite Deltacomplexes: DC.
- Finite cubical complexes: CC.

Note: References:

- [Dod76a]
- [Dod76b]
- [DR02]
- [FIS97]
- [Gol02]
- [TB20]

---

class Hodge(*args: Any**kwargs: Any)

Bases: SageObject, UniqueRepresentation, CachedRepresentation

This class implements the Combinatorial Laplacian and the Discrete Hodge Decomposition of the cochain complex of a simplicial structure. For such decomposition, this class makes use of the AT_model implementation available at SAGEMATH.

base_ring()

Return the base ring of "self".

boundary_operator(dim)

The differentials which make up the chain complex.

INPUT:

- dim – element of the grading group (optional, default "None");

OUTPUT:

The differential starting in dimension dim as a sparse matrix.

tab-content EXAMPLES:

```
sage: K = simplicial_complexes.Sphere(2)
sage: H = Hodge(K)
sage: H.boundary_operator(1)
[-1 -1 -1  0  0  0]
[ 1  0  0 -1 -1  0]
[ 0  1  0  1  0 -1]
[ 0  0  1  0  1  1]
sage: H.boundary_operator(1) == K.chain_complex().differential(1)
True
```

coboundary_operator(dim)

The differentials which make up the cochain complex.

INPUT:

- dim – element of the grading group (optional, default "None");

OUTPUT:

The differential starting in dimension dim as a sparse matrix.

tab-content EXAMPLES:

```
sage: K = simplicial_complexes.Sphere(2)
sage: H = Hodge(K)
sage: H
The Discrete Hodge structure of Minimal triangulation of the 2-sphere over␣
↪the base ring Rational Field
sage: H.coboundary_operator(1)
[ 1 -1  0  1  0  0]
[ 1  0 -1  0  1  0]
[ 0  1 -1  0  0  1]
[ 0  0  0  1 -1  1]
sage: H.coboundary_operator(1) == K.chain_complex().dual().differential(1)
True
```

coexact_basis(d)

The basis of the coexact (nonclosed) cochain subspace of $C^d(K)$.

tab-content EXAMPLES:

```
sage: K = simplicial_complexes.Sphere(2)
sage: H = Hodge(K)
sage: H.coexact_basis(1)
[(1, 0, -1, 0, 1, 0), (0, 1, -1, 0, 0, 1), (0, 0, 0, 1, -1, 1)]
```

coexact_matrix(d)

The coexact (nonclosed) cochain subspace of $C^d(K)$ as a sparse matrix with rows the basis of the vector subspace.

tab-content EXAMPLES:

```
sage: K = simplicial_complexes.Sphere(2)
sage: H = Hodge(K)
sage: H.coexact_basis(1)
[(1, 0, -1, 0, 1, 0), (0, 1, -1, 0, 0, 1), (0, 0, 0, 1, -1, 1)]
sage: H.coexact_matrix(1)
[ 1  0 -1  0  1  0]
[ 0  1 -1  0  0  1]
[ 0  0  0  1 -1  1]
```

exact_basis(d)

The basis of the exact cochain subspace of $C^d(K)$.

tab-content EXAMPLES:

```
sage: K = simplicial_sets.Torus()
sage: H = Hodge(K)
sage: H.harmonic_basis(1)
[
(1, 0, 1),
(0, 1, 1)
]
sage: H.exact_basis(1)
[]
sage: H.coexact_basis(1)
[(1, 1, -1)]
```

exact_matrix(d)

The exact cochain subspace of $C^d(K)$ as a sparse matrix with rows the basis of the vector subspace.

tab-content EXAMPLES:

```
sage: K = delta_complexes.KleinBottle()
sage: H = Hodge(K)
sage: [H2,E2,C2] = H.hodge_decomposition(2)
sage: H2
[0 0]
sage: E2
[ 1  1]
[ 1 -1]
```

(continues on next page)

(continued from previous page)

```
sage: C2
[]
sage: K.dimension()
2
sage: [H1,E1,C1] = H.hodge_decomposition(1)
sage: H1
[0 0 1]
sage: E1
[]
sage: C1
[1 0 0]
[0 1 0]
sage: d1 = H.coboundary_operator(1)
sage: (d1*C1.T).T == E2
True
```

tab-content EXAMPLES:

```
K = delta_complexes.KleinBottle()
H = Hodge(K)
[H2,E2,C2] = H.hodge_decomposition(2)
print("K.dimension()")
[H1,E1,C1] = H.hodge_decomposition(1)
d1 = H.coboundary_operator(1)
print((d1*C1.T).T == E2)
```

harmonic_basis(d)

The basis of the harmonic cochain subspace of $C^d(K)$.

tab-content EXAMPLES:

```
sage: K = simplicial_sets.Torus()
sage: H = Hodge(K)
sage: H.harmonic_basis(1)
[
(1, 0, 1),
(0, 1, 1)
]
sage: H.exact_basis(1)
[]
sage: H.coexact_basis(1)
[(1, 1, -1)]
```

harmonic_matrix(d)

The harmonic cochain subspace of $C^d(K)$ as a sparse matrix with rows the basis of the vector subspace.

tab-content EXAMPLES:

```
sage: K = simplicial_sets.Torus()
sage: H = Hodge(K)
sage: H.harmonic_matrix(1)
[1 0 1]
```

(continues on next page)

(continued from previous page)

```
[0 1 1]
sage: H.exact_matrix(1)
[0 0 0]
sage: H.coexact_matrix(1)
[ 1  1 -1]
```

hodge_cochain_decomposition(cochaindegree)

Returns the hodge decomposition of the cochain c as a sum of three components

$$cochain = h + e + c$$

where h is the harmonic component, e is the exact component and c is the coexact component.

INPUT:

- `cochain` – a cochain of $C^d(K)$ represented as a vector over the base ring.
- `degree` – the degree of the cochain.

OUTPUT: a tuple *(h,e,c)* of elements of $C^d(K)$

tab-content EXAMPLES:

```
sage: K = cubical_complexes.Sphere(2)
sage: H = Hodge(K)
sage: v = random_vector(QQ, len(K.n_cells(2)))
sage: v
(-1/15, 1, 1/449, 1/3, -2/5, 0)
sage: (h,e,c) = H.hodge_cochain_decomposition(v,2)
sage: v == h+c+e
True
sage: h
(0, 0, 0, 0, 0, 4036/2245)
sage: e
(-1/15, 1, 1/449, 1/3, -2/5, -4036/2245)
sage: c
(0, 0, 0, 0, 0, 0)
```

hodge_decomposition(dim)

Returns the hodge decomposition of $C^dim(K)$.

INPUT:

- `dim` – the degree of the cochain module.

OUTPUT: a tuple of matrices *(H,E,C)* corresponding with the harmonic, exact and coexact matrices.

See also:

harmonic_matrix()

exact_matrix()

coexact_matrix()

tab-content EXAMPLES:

The hodge decomposition of the simplicial cochain complex of the 2Sphere at degree 2:

sage: K = cubical_complexes.Sphere(2) sage: H = Hodge(K) sage: (H,E,C) = H.hodge_decomposition(2) sage: H [0 0 0 0 0 1] sage: E [1 0 0 0 0 1] [0 1 0 0 0 1] [0 0 1 0 0 1] [0 0 0 1 0 1] [0 0 0 0 1 1] sage: C [] sage: len(K.n_cells(2)) 6

(TO UPDATE) Example that ilustrates the orthogonality of the decomposition:

sage: S2 = simplicial_sets.Sphere(2) sage: T = simplicial_sets.Torus() sage: K = S2.product(S2).product(T) sage: K S^2 x S^2 x Torus sage: W = APLK(K) sage: W Algebra of Whitney Forms on S^2 x S^2 x Torus sage: hodge = W.hodge()

tab-content S2 = simplicial_sets.Sphere(2) T = simplicial_sets.Torus() K = S2.product(S2).product(T) ch = K.chain_complex(base_ring=QQ).dual() (phi,M) = K.algebraic_topological_model() d1 = ch.differential(1) d2 = ch.differential(2) C = phi.dual().in_degree(3)*d2 E = d1*phi.dual().in_degree(2) H = phi.dual().iota().in_degree(2)*phi.dual().pi().in_degree(2) H*E == 0 H*C == 0 C*E == 0 E*C == 0

laplacian(dim)

Returns the discrete laplacian of dimension `dim`.

INPUT:

- `dim` – the dimension of the cochain module.

OUTPUT: a tuple of sparse matrices *(L, Ldown, Lup)* corresponding with the laplacian, the down laplacian and the up laplacian matrices.

tab-content EXAMPLES:

```
sage: K = delta_complexes.KleinBottle()
sage: K
Delta complex with 1 vertex and 7 simplices
sage: H = Hodge(K)
sage: H
The Discrete Hodge structure of Delta complex with 1 vertex and 7 simplices␣
↪over the base ring Rational Field
sage: (L, Ldown, Lup) = H.laplacian(1)
sage: L
[ 2  0  0]
[ 0  2 -2]
[ 0 -2  2]
sage: Ldown
[0 0 0]
[0 0 0]
[0 0 0]
sage: Lup
[ 2  0  0]
[ 0  2 -2]
[ 0 -2  2]
sage: L.right_kernel().dimension() == len(H.harmonic_basis(1))
True
```

preimage(cochaindegree)

Returns the coexact cochain of `degree1` whose differential correspond with the exact component of the given cochain.

INPUT:

- `cochain` – a cochain vector

- degree – the degree of the cochain.

OUTPUT: a cochain vector of degree degree1 whose differential corresponds with the exact component of the given cochain.

tab-content EXAMPLES:

```
sage: S2 = simplicial_sets.Sphere(2)
sage: K = S2.product(S2)
sage: H = Hodge(K)
sage: H
The Discrete Hodge structure of S^2 x S^2 over the base ring Rational Field
sage: v = random_vector(QQ, len(K.n_cells(3)))
sage: (h,e,c) = H.hodge_cochain_decomposition(v,3)
sage: h
(0, 0, 0, 0, 0, 0)
sage: e
(1/12, 0, 1/12, -1/12, 0, -1/12)
sage: c
(-1/12, -2, 1/4, 1/12, -1/6, 1/12)
sage: y = H.preimage(v,3)
sage: dif = K.chain_complex(base_ring=QQ).dual().differential(2)
sage: dif*y == e
True
```

CHAPTER

FOUR

THE CDG ALGEBRA $\mathcal{A}_{\mathcal{PL}}(\mathcal{K})$

The cochain algebra $A_{PL}(K)$. Let K be a simplicial set, and let $A_{PL} = \{(A_{PL})_n\}_{n\geq 0}$ be the simplicial cochain algebra defined at simplicialapl

$$A_{PL}(K) = \bigoplus_{p\geq 0} A^p_{PL}(K)$$

is the cochain algebra defined as follows:

- The homogenous part $A^p_{PL}(K)$ is the set of simplicial set morphisms from K to A^p_{PL}. That is, an element $\Phi \in A^p_{PL}(K)$ is a mapping that assigns to each $n$simplex $\sigma \in K_n$, an element $\Phi(\sigma) \in (A^p_{PL})_n$, satisfying

$$\Phi(\partial_i(\sigma)) = \partial_i(\Phi(\sigma))$$
$$\Phi(s_j(\sigma)) = s_j(\Phi(\sigma))$$

- Addition, scalar multiplication, product and the differential are induced by the corresponding operations in the algebras $(A_{PL})_n$.

The structure of the cochain algebra $A_{PL}(K)$, at some general dimension n and degree p, can be visualized in figure fig:simplicial_aplK. The horizontal direction corresponds to the simplicial structure for a given dimension n (subscript), and the vertical direction corresponds to the graded structure of the algebra (representing the degree p with superscript). As the simplicial cochain algebra $(A_{PL})_n$ is at the ground of this structure, all the elements of $A^p_{PL}(K)_n$ will vanish when $p > n$. This is because it is not possible to assign nonzero polynomial forms of degree greater than the dimension of the simplices.

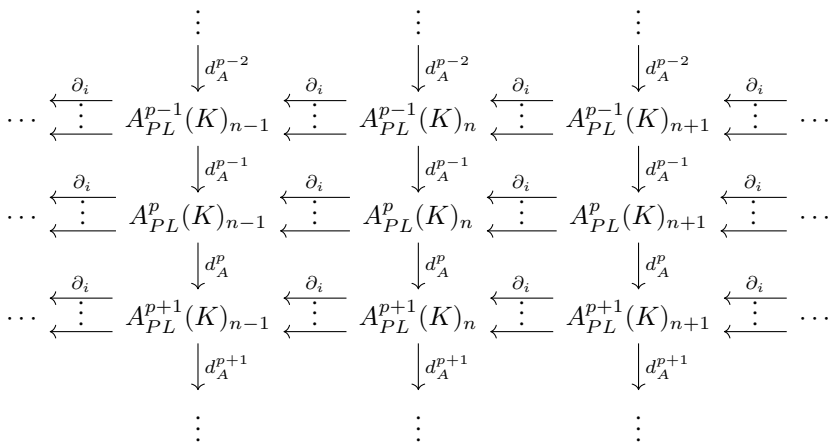


Fig. 1: The structure of the cochain algebra $A_{PL}(K)$.

class APLK(*args: Any**kwargs: Any)

Bases: `Parent`, `UniqueRepresentation`, `CachedRepresentation`

cohomology_algebra(max_degree=1verbose=False)

Compute a CDGA with trivial differential, that is isomorphic to the cohomology of `self` up to `max_degree` (default: dimension of the space)

INPUT:

- `max_degree` – integer (default: *dimension of the space*); degree to which the result is required to be isomorphic to `self`'s cohomology

tab-content EXAMPLES:

```
sage: S2 = simplicial_sets.Sphere(2)
sage: K = S2.product(S2)
sage: K.cohomology()
{0: 0, 1: 0, 2: Z x Z, 3: 0, 4: Z}
sage: W = APLK(K)
sage: H = W.cohomology_algebra()
sage: H
Commutative Differential Graded Algebra with generators ('x0', 'x1', 'x2')␣
↪in degrees (2, 2, 4) with relations [x0^2 + 2*x2, x0*x1 - x2, x1^2] over␣
↪Rational Field with differential:
   x0 --> 0
   x1 --> 0
   x2 --> 0
sage: hgens = K.cohomology_ring(base_ring=QQ).gens()
sage: hgens
(h^{0,0}, h^{2,0}, h^{2,1}, h^{4,0})
sage: x0 = hgens[1]
sage: x1 = hgens[2]
sage: x2 = hgens[3]
sage: x0**2+2*x2
0
sage: x0*x1-x2
```

(continues on next page)

(continued from previous page)

```
0
sage: x1**2
0
```

tab-content EXAMPLES:

```
S2 = simplicial_sets.Sphere(2)
K = S2.product(S2)
print("Simplicial cohomology")
print(K.cohomology())
W = APLK(K)
H = W.cohomology_algebra()
print(H)
hgens = K.cohomology_ring(base_ring=QQ).gens()
print("testing relations of the generators")
x0 = hgens[1]
x1 = hgens[2]
x2 = hgens[3]
print(f"x0**2+2*x2 ==> {x0**2+2*x2}")
print(f"x0*x1-x2 ==> {x0*x1-x2}")
print(f"x1**2 ==> {x1**2}")
```

is_formal(imax_iterations=3) → bool

Check if the algebra is iformal.

That is, if it is iquasiisomorphic to its cohomology algebra.

INPUT:

- i – integer; the degree up to which the formality is checked
- max_iterations – integer (default: *3*); the maximum number of iterations used in the computation of the minimal model

Warning: The method is not granted to finish (it can't, since the minimal model could be infinitely generated in some degrees). The parameter max_iterations controls how many iterations of the method are attempted at each degree. In case they are not enough, an exception is raised. If you think that the result will be finitely generated, you can try to run it again with a higher value for max_iterations.

Moreover, the method uses criteria that are often enough to conclude that the algebra is either formal or nonformal. However, it could happen that the used criteria can not determine the formality. In that case, an error is raised.

EXAMPLES:

```
sage: A.<e1, e2, e3, e4, e5> = GradedCommutativeAlgebra(QQ)
sage: B = A.cdg_algebra({e5: e1*e2 + e3*e4})
sage: B.is_formal(1)
True
sage: B.is_formal(2)
False
```

ALGORITHM:

Apply the criteria in [MMBuzunariz20]. Both the *i*minimal model of the algebra and its cohomology algebra are computed. If the numerical invariants are different, the algebra is not *i*formal.

If the numerical invariants match, the *psi* condition is checked.

numerical_invariants(max_degree=3max_iterations=3)

Return the numerical invariants of the algebra, up to degree `d`. The numerical invariants reflect the number of generators added at each step of the construction of the minimal model.

The numerical invariants are the dimensions of the subsequent Hirsch extensions used at each degree to compute the minimal model.

INPUT:

- `max_degree` – integer (default: *3*); the degree up to which the numerical invariants are computed
- `max_iterations` – integer (default: *3*); the maximum number of iterations used to compute the minimal model, if it is not already cached

EXAMPLES:

```
sage: A.<e1, e2, e3> = GradedCommutativeAlgebra(QQ)
sage: B = A.cdg_algebra({e3 : e1*e2})
sage: B.minimal_model(4)
Commutative Differential Graded Algebra morphism:
From: Commutative Differential Graded Algebra with
      generators ('x1_0', 'x1_1', 'y1_0') in degrees (1, 1, 1)
      over Rational Field with differential:
        x1_0 --> 0
        x1_1 --> 0
        y1_0 --> x1_0*x1_1
To:   Commutative Differential Graded Algebra with
      generators ('e1', 'e2', 'e3') in degrees (1, 1, 1)
      over Rational Field with differential:
        e1 --> 0
        e2 --> 0
        e3 --> e1*e2
Defn: (x1_0, x1_1, y1_0) --> (e1, e2, e3)
sage: B.numerical_invariants(2)
{1: [2, 1, 0], 2: [0, 0]}
```

ALGORITHM:

The numerical invariants are stored as the minimal model is constructed.

Warning: The method is not granted to finish (it can't, since the minimal model could be infinitely generated in some degrees). The parameter `max_iterations` controls how many iterations of the method are attempted at each degree. In case they are not enough, an exception is raised. If you think that the result will be finitely generated, you can try to run it again with a higher value for `max_iterations`.

REFERENCES:

For a precise definition and properties, see [MMBuzunariz20].

minimal_model(i=3max_iterations=3partial_result=Falseverbose=FalseparallelComp=False)

tab-content EXAMPLES:

```
sage: load("Sullivan_MM/tests/zariski.py")
sage: [S6_1, S6_2] = pair6()
sage: A1 = APLK(S6_1)
sage: M1 = A1.minimal_model(i=2, max_iterations=2,parallelComp=False,
↪verbose=True,partial_result=True)
Starting the computation at degree 1
Checking injectivity at degree 2 ==> iteration 1 (out of 2)
(1/3) Trying to get a Whitney form primitive...
(1/3) Trying to get a Whitney form primitive...
(1/3) Trying to get a Whitney form primitive...
Extending the model...
....creating GCA...
....building HOM B --> A...
....updating differential...
....Building new Minimal Model...
Checking injectivity at degree 2 ==> iteration 2 (out of 2)
sage: B1 = M1.domain()
sage: B1
Commutative Differential Graded Algebra with generators ('x1_0', 'x1_1', 'x1_
↪2', 'x1_3', 'x1_4', 'y1_0',
 'y1_1', 'y1_2', 'y1_3', 'y1_4', 'y1_5') in degrees (1, 1, 1, 1, 1, 1, 1, 1,␣
↪1, 1, 1) over Rational
 Field with differential:
   x1_0 --> 0
   x1_1 --> 0
   x1_2 --> 0
   x1_3 --> 0
   x1_4 --> 0
   y1_0 --> x1_1*x1_2
   y1_1 --> x1_1*x1_3
   y1_2 --> x1_2*x1_3
   y1_3 --> x1_1*x1_4
   y1_4 --> x1_2*x1_4
   y1_5 --> x1_3*x1_4
sage: B1.cohomology(2)
Free module generated by {[x1_0*x1_1], [x1_0*x1_2], [x1_0*x1_3], [x1_0*x1_4],
↪ [x1_1*y1_0], [x1_2*y1_0],
[x1_3*y1_0 - x1_1*y1_2], [x1_4*y1_0 - x1_1*y1_4], [x1_1*y1_1], [x1_2*y1_1 +␣
↪x1_1*y1_2], [x1_3*y1_1],
[x1_4*y1_1 - x1_1*y1_5], [x1_2*y1_2], [x1_3*y1_2], [x1_4*y1_2 - x1_2*y1_5],␣
↪[x1_1*y1_3],
[x1_2*y1_3 + x1_1*y1_4], [x1_3*y1_3 + x1_1*y1_5], [x1_4*y1_3], [x1_2*y1_4],␣
↪[x1_3*y1_4 + x1_2*y1_5],
[x1_4*y1_4], [x1_3*y1_5], [x1_4*y1_5]} over Rational Field
sage: x13 = M1._imgens['x1_3']
sage: y15 = M1._imgens['y1_5']
sage: p = x13*y15
sage: p
[1/12]*e2_28
```

(continues on next page)

(continued from previous page)

```
sage: p.dif()
0
sage: dp = x13*y15
sage: p = dp.solve_primitive()
sage: p.dif() == dp
True
sage: p.check()
True

sage: S2 = simplicial_sets.Sphere(2)
sage: S4 = simplicial_sets.Sphere(4)
sage: CP2 = simplicial_sets.ComplexProjectiveSpace(2)
sage: WS = APLK(S2.wedge(S4))
sage: WC = APLK(CP2)
sage: MS = WS.minimal_model(i=4,max_iterations=3,verbose=True)
Starting the computation at degree 2
Checking injectivity at degree 3 ==> iteration 1 (out of 3)
Checking surjectivity at degree 3...
Checking injectivity at degree 4 ==> iteration 1 (out of 3)
Extending the model...
....creating GCA...
....building HOM B --> A...
....updating differential...
....Building new Minimal Model...
Checking injectivity at degree 4 ==> iteration 2 (out of 3)
Checking surjectivity at degree 4...
Extending the model...
....creating GCA...
....building HOM B --> A...
....updating differential...
....Building new Minimal Model...
Checking injectivity at degree 5 ==> iteration 1 (out of 3)
sage: MC = WC.minimal_model(i=4,max_iterations=3,verbose=True)
Starting the computation at degree 2
Checking injectivity at degree 3 ==> iteration 1 (out of 3)
Checking surjectivity at degree 3...
Checking injectivity at degree 4 ==> iteration 1 (out of 3)
Checking surjectivity at degree 4...
Checking injectivity at degree 5 ==> iteration 1 (out of 3)
sage: MS.domain()
Commutative Differential Graded Algebra with generators ('x2_0', 'y3_0', 'x4_
↪0') in degrees (2, 3, 4) over Rational Field with differential:
   x2_0 --> 0
   y3_0 --> x2_0^2
   x4_0 --> 0
sage: MC.domain()
Commutative Differential Graded Algebra with generators ('x2_0',) in degrees␣
↪(2,) with relations [0] over Rational Field with differential:
   x2_0 --> 0
sage: S2.wedge(S4).cohomology()
{0: 0, 1: 0, 2: Z, 3: 0, 4: Z}
```

(continues on next page)

(continued from previous page)

```
sage: CP2.cohomology()
{0: 0, 1: 0, 2: Z, 3: 0, 4: Z}
sage: MS.cohomology(2)
Free module generated by {[x2_0]} over Rational Field
sage: MS.cohomology(3)
Free module generated by {} over Rational Field
sage: MS.cohomology(4)
Free module generated by {[x4_0]} over Rational Field
sage: MC.cohomology(2)
Free module generated by {[x2_0]} over Rational Field
sage: MC.cohomology(3)
Free module generated by {} over Rational Field
sage: MC.cohomology(4)
Free module generated by {[x2_0^2]} over Rational Field
```

tab-content EXAMPLES:

```
load("Sullivan_MM/tests/zariski.py")
[S6_1, S6_2] = pair6()
A1 = APLK(S6_1)
M1 = A1.minimal_model(i=2, max_iterations=2,parallelComp=False,verbose=True,
↪partial_result=True)
B1 = M1.domain()
B1.cohomology(2)
x13 = M1._imgens['x1_3']
y15 = M1._imgens['y1_5']
dp = x13*y15
p = dp.solve_primitive()
p.dif() == dp
p.check()

S2 = simplicial_sets.Sphere(2)
S4 = simplicial_sets.Sphere(4)
CP2 = simplicial_sets.ComplexProjectiveSpace(2)
WS = APLK(S2.wedge(S4))
WC = APLK(CP2)
MS = WS.minimal_model(i=4,max_iterations=3,verbose=True)
MC = WC.minimal_model(i=4,max_iterations=3,verbose=True)
MS.domain()
MC.domain()
S2.wedge(S4).cohomology()
CP2.cohomology()
MS.cohomology(2)
MS.cohomology(3)
MS.cohomology(4)
MC.cohomology(2)
MC.cohomology(3)
MC.cohomology(4)
```

class APLKElement(*args: Any**kwargs: Any)

Bases: Element

dif(check=False)
Alias for `differential()`

dup(check=False)
Alias for *dupont_contraction()*

dupont_contraction(check=False)
The Dupont's contraction of polynomial differential forms defined on a generic simplicial set S (see [Dup76]). This method implements the Dupont's contraction

$$s_n : A^*_{PL}(S_n) \to A^{*-1}_{PL}(S_n)$$

as it is described in [CG08] (see also [Dup76]).

Let $\mathbb{K}$ be a field of characteristic 0 and let $S = \{S_n\}_{n\geq 0}$ be a simplicial set. Also, let $A_{PL}(S)$ be the algebra of polynomial differential forms defined on S, and $C^*(S)$, the simplicial cochain complex associated to S.

Let $I = \{i_0, \dots, i_p\}$, with $0 \leq i_0 < \cdots < i_p \leq n$, and define

$$J := [n] \setminus I = \{j_0, \dots, j_q\}$$

such that $0 \leq j_0 < \cdots < j_q \leq n$ and $p + q = n$.

Let $\sigma_p \in S_p$ and $\sigma_n \in S_n$ be two simplices of S such that, $\partial_J(\sigma_n) = \sigma_p$ where

$$\partial_J = \partial_{j_0} \circ \cdots \circ \partial_{j_q}.$$

Let $\beta \in C^p(S)$. The Whitney map (see [LK21, Whi57])

$$\mathcal{W}^p = \{\mathcal{W}^p_n : C^p(S_p) \to A^p_{PL}(S_n)\}$$

is defined as:

$$\mathcal{W}^p_n(\beta)(\sigma_n)|_{\sigma_p} = \beta(\sigma_p) \cdot n! \cdot \sum_{k=0}^{p} (-1)^j t_{i_k} y_{i_0} \wedge \cdots \wedge \hat{y}_{i_k} \wedge \cdots \wedge y_{i_p}$$

where the symbol $\hat{}$ means the term is omitted. The elements in the image of $\mathcal{W}$ are called the *elementary differential forms* or just *Whitney forms*. The motivation behind this notation is that the subset I comes from the inclusion

$$i : \Delta^p \hookrightarrow \Delta^n$$

that, by the A_{PL} functor translates to the inclusion of algebras:

$$\begin{array}{rccc} A_{PL}(i) : & (A_{PL})_p & \hookrightarrow & (A_{PL})_n \\ & (t_0, \dots, t_p) & \to & (t_{i_0}, \dots, t_{i_p}) \end{array}$$

Finally, consider the de Rham map (see [Dup76, FelixHT01, Whi57]):

$$\int_n : A^n_{PL}(S) \to C^n(S)$$

Dupont constructed an explicit simplicial contraction of the commutative differential graded algebra of polynomial differential forms $A_{PL}(S)$ to $C^*(S)$ compatible with the intrinsic affine structure of the polynomial differential forms (affine relations) and also with the wedge product of forms (up to homotopy).

This method implements step by step the description of the contraction given by Cheng et al. (see [CG08]), with minor notation adaptations for the sake of maintain coherence with the rest of the code. The description given by Cheng et al. is as follows:

For each vertex e_i of the $n$simplex Δ^n, define the dilation map

$$\phi_i : [0,1] \times \Delta^n \to \Delta^n$$

by the formula

$$\phi_i(u, t_0, \ldots, t_n) = ((1-u)t_0, \ldots, (1-u)t_i + u, \ldots, (1-u)t_n).$$

Let

$$(\pi_i)_* : A^*_{PL}([0,1] \times \Delta^n) \to A^{*-1}_{PL}(\Delta^n)$$

be integration over the first factor.

Now, let $\omega \in (A_{PL})_n$ and define the operator

$$h^i_n : A^*_{PL}(\Delta^n) \to A^{*-1}_{PL}(\Delta^n)$$

by the formula

$$h^i_n \omega = (\pi_i)_* \phi_i^* \omega.$$

where $\phi_i^* \omega$ is the pullback of ω respect to ϕ_i.

Let

$$\epsilon^i_n : A^*_{PL}(\Delta^n) \to \mathbb{K}$$

be evaluation at the vertex e_i. The Poincar'e Lemma states that

$$\mathbb{1}_A - \epsilon^i_n = d \circ h^i_n + h^i_n \circ d.$$

Now, let $J^k := \{\{i_0, \ldots, i_{k-1}\} \mid 0 \leq i_0 < \cdots < i_{k-1} \leq n\}$, be the set of all subsets of $\{0, \ldots, n\}$ with cardinality k.

Theorem (Dupont [Dup76]):

The operator

$$s_n(\omega) = \sum_{k=0}^{n-1} \sum_{I \in J^k} \mathcal{W}^I_n(1) \wedge (h^{i_k} \circ \cdots \circ h^{i_0})(\omega)$$

defines a simplicial contraction from $A^*_{PL}(S)$ to $C^*(S)$:

$$\mathcal{W}_n \circ \int_n - \mathbb{1}_A = d \circ s_n + s_n \circ d.$$

Theorem (Getzler [Get09]):

The side conditions

- $\int_n \circ s_n = 0$
- $s_n \circ s_n = 0$

hold.

See also:

AplN.Element.integral()

AplN.whitney_form()

AplN.Element.dupont_operator()

SimplicialAPL.whitney_map()

integral()

INPUT:

- `p` – an element of `self`.
- `i` – integer; the vertex index to apply the operator

OUTPUT: an element of `self`;

EXAMPLES:

```
sage: K = simplicial_complexes.Sphere(2)
sage: W = WhitneyFormAlgebraComplex(K)
sage: W.inject_variables()
Getting generators of degree  0 ...
Getting generators of degree  1 ...
Getting generators of degree  2 ...
Defining t0, t1, t2, t3, e1_0, e1_1, e1_2, c1_0, c1_1, c1_2, h2_0, e2_0, e2_
↪1, e2_2
sage: p = t0**2+t3**5
sage: dp = p.differential()
sage: wi = sum(dp.reduce(0).hodge_whitney_decomposition())
sage: dp-wi == dp.dupont_contraction().differential()
True
```

integral()

Returns the integral of `self`.

George De Rham [DR31] in the 1930's proved that integration provides a natural direct quasiisomorphism of cochain complexes that connects the algebra of differential forms (in this context $(A_{PL}(K))$ with the cochain complex $C^*(K)$:

$$\oint : A_{PL}(K) \to C^*(K)$$

as follows. Let $\int_n$ be the linear map:

$$\int_n : (A_{PL})^n_n \to \mathbb{Q}$$

by setting This map represents the integration $\int_{\Delta^n} \omega$ of an *n*form $\omega \in (A_{PL})^n_n$ over the standard *n*simplex Δ^n. Using this map, we can integrate elements of $A_{PL}(K)$ as follows: consider a homogeneous element $\Phi^n_n \in A^n_{PL}(K_n)$. Its integral is the element of $C^n(K)$ determined by the integration map:

$$\left(\oint (\Phi^n_n)\right)(\sigma) = \int_n \Phi^n_n(\sigma),$$

for every $\sigma \in K_n$. For homogeneous elements $\Phi_m^n \in A_{PL}^n(K_m)$ with $n \neq m$, the integral is set to zero ($\left(\oint(\Phi_m^n)\right)(\sigma) = 0$). Then, the map is extended to all $A_{PL}(K)$ by linearity.

OUTPUT: a vector represented the cochain element.

EXAMPLES:

```
sage: from Sullivan_MM import *
sage: S2 = simplicial_complexes.Sphere(2)
sage: K = S2.wedge(S2)
sage: K.set_immutable()
sage: W = WhitneyFormAlgebraComplex(K)
sage: W.inject_variables()
Getting generators of degree  0 ...
Getting generators of degree  1 ...
Getting generators of degree  2 ...
Defining t0, t1, t2, t3, t4, t5, t6, e1_0, e1_1, e1_2, e1_3, e1_4, e1_5, c1_
↪0, c1_1, c1_2, c1_3, c1_4,
c1_5, h2_0, h2_1, e2_0, e2_1, e2_2, e2_3, e2_4, e2_5
sage: e2_1.integral()
(0, 1, 0, 0, 0, 0, -1, 0)
sage: W.whitney_form_from_vector(v=e2_1.integral(),dim=2) == e2_1
True
```

class MorphismFromQQToAPLK(*args: Any**kwargs: Any)

Bases: Morphism

class CohomologyClass(*args: Any**kwargs: Any)

Bases: SageObject, CachedRepresentation

A class for representing cohomology classes.

This just has _repr_ and _latex_ methods which put brackets around the object's name.

EXAMPLES:

```
sage: from sage.algebras.commutative_dga import CohomologyClass
sage: CohomologyClass(3)
[3]
sage: A.<x,y,z,t> = GradedCommutativeAlgebra(QQ, degrees=(2,2,3,3))
sage: CohomologyClass(x^2 + 2*y*z, A)
[2*y*z + x^2]
```

TESTS:

In order for the cache to not confuse objects with the same representation, we can pass the parent of the representative as a parameter:

```
sage: A.<e1,e2,e3,e4,e5,e6> = GradedCommutativeAlgebra(QQ)
sage: B1 = A.cdg_algebra({e5:e1*e2,e6:e3*e4})
sage: B2 = A.cdg_algebra({e5:e1*e2,e6:e1*e2+e3*e4})
sage: B1.minimal_model()
Commutative Differential Graded Algebra morphism:
  From: Commutative Differential Graded Algebra with generators ('x1_0', 'x1_1',
↪'x1_2', 'x1_3', 'y1_0', 'y1_1') in degrees (1, 1, 1, 1, 1, 1) over Rational␣
↪Field with differential:
```

(continues on next page)

(continued from previous page)

```
   x1_0 --> 0
   x1_1 --> 0
   x1_2 --> 0
   x1_3 --> 0
   y1_0 --> x1_0*x1_1
   y1_1 --> x1_2*x1_3
  To:   Commutative Differential Graded Algebra with generators ('e1', 'e2', 'e3
↪', 'e4', 'e5', 'e6') in degrees (1, 1, 1, 1, 1, 1) over Rational Field with␣
↪differential:
   e1 --> 0
   e2 --> 0
   e3 --> 0
   e4 --> 0
   e5 --> e1*e2
   e6 --> e3*e4
  Defn: (x1_0, x1_1, x1_2, x1_3, y1_0, y1_1) --> (e1, e2, e3, e4, e5, e6)
sage: B2.minimal_model()
Commutative Differential Graded Algebra morphism:
  From: Commutative Differential Graded Algebra with generators ('x1_0', 'x1_1',
↪'x1_2', 'x1_3', 'y1_0', 'y1_1') in degrees (1, 1, 1, 1, 1, 1) over Rational␣
↪Field with differential:
   x1_0 --> 0
   x1_1 --> 0
   x1_2 --> 0
   x1_3 --> 0
   y1_0 --> x1_0*x1_1
   y1_1 --> x1_2*x1_3
  To:   Commutative Differential Graded Algebra with generators ('e1', 'e2', 'e3
↪', 'e4', 'e5', 'e6') in degrees (1, 1, 1, 1, 1, 1) over Rational Field with␣
↪differential:
   e1 --> 0
   e2 --> 0
   e3 --> 0
   e4 --> 0
   e5 --> e1*e2
   e6 --> e1*e2 + e3*e4
  Defn: (x1_0, x1_1, x1_2, x1_3, y1_0, y1_1) --> (e1, e2, e3, e4, e5, -e5 + e6)
```

representative()

Return the representative of `self`.

EXAMPLES:

```
sage: from sage.algebras.commutative_dga import CohomologyClass
sage: x = CohomologyClass(sin)
↪# needs sage.symbolic
sage: x.representative() == sin
↪# needs sage.symbolic
True
```

class MinimalModel(*args: Any**kwargs: Any)

Bases: `SageObject`, `UniqueRepresentation`, `CachedRepresentation`

im_gens()

Return the images of the generators of the domain.

OUTPUT:

- `dict` – a copy of the dictionary indexed by the generators of the minimal model, and its values are the corresponding images by the morphism :math:'vaphi'

tab-content

EXAMPLES:: sage: S2 = simplicial_sets.Sphere(2) sage: W = APLK(S2) sage: M = W.minimal_model(i=6) sage: M.im_gens() {'x2_0': h2_0, 'y3_0': 0}

laplacian(dim)

Returns the discrete laplacian of dimension `dim`.

INPUT:

- `dim` – the dimension of the cochain module.

OUTPUT: a tuple of sparse matrices *(L, Ldown, Lup)* corresponding with the laplacian, the down laplacian and the up laplacian matrices.

tab-content EXAMPLES:

BIBLIOGRAPHY

[CG08] Xue Zhi Cheng and Ezra Getzler. Transferring homotopy commutative algebraic structures. *Journal of Pure and Applied Algebra*, 212(11):2535–2542, 2008.

[DR31] Georges De Rham. Sur l'analysis situs des variétés à $ n $ dimensions. *Journal de mathématiques pures et appliquées*, 10:115–200, 1931.

[Dod76a] Jozef Dodziuk. Finitedifference approach to the hodge theory of harmonic forms. *American Journal of Mathematics*, 98(1):79–104, 1976.

[Dod76b] Józef Dodziuk. A remark on whitney's proof of de rham's theorem. *Proceedings of the American Mathematical Society*, 54(1):360–362, 1976.

[Dup76] Johan L Dupont. Simplicial de rham cohomology and characteristic classes of flat bundles. *Topology*, 15(3):233–245, 1976.

[DR02] Art Duval and Victor Reiner. Shifted simplicial complexes are laplacian integral. *Transactions of the American Mathematical Society*, 354(11):4313–4344, 2002.

[FIS97] Stephen H Friedberg, Arnold J Insel, and Lawrence E Spence. *Linear algebra*. Prentice Hall, 1997.

[FelixHT01] Yves Félix, Stephen Halperin, and JeanClaude Thomas. *Rational Homotopy Theory*. Number 205 in Graduate Texts in Mathematics. Springer, New York, 2001. ISBN 9780387950686.

[Get09] Ezra Getzler. Lie theory for nilpotent $l_\infty $ algebras. *Annals of Mathematics*, 170(1):271–301, 2009.

[Gol02] Timothy E Goldberg. *Combinatorial Laplacians of simplicial complexes*. PhD thesis, Bard College, 2002.

[LK21] Jonni Lohi and Lauri Kettunen. Whitney forms and their extensions. *Journal of Computational and Applied Mathematics*, 393:113520, 2021.

[MMBuzunariz20] Víctor Manero and Miguel Marco Buzunáriz. Effective computation of degree bounded minimal models of GCDAs. *Journal of Software for Algebra and Geometry*, 10(1):25–39, May 2020. URL: https://msp.org/jsag/2020/10-1/p04.xhtml (visited on 20230209), doi:10.2140/jsag.2020.10.25.

[PR15] Pawel Pilarczyk and Pedro Real. Computation of cubical homology, cohomology, and (co) homological operations via chain contraction. *Advances in Computational Mathematics*, 41(1):253–275, 2015.

[TB20] Joaquín J Torres and Ginestra Bianconi. Simplicial complexes: higherorder spectral dimension and dynamics. *Journal of Physics: Complexity*, 1(1):015002, may 2020.

[Whi57] Hassler Whitney. *Geometric Integration Theory*. Princeton University Press, Princeton, N.J., 1957.

INDEX

affine_map()SimplicialAPL method, 88
APLKclass in Sullivan_MM.AplK, 104
APLKElementclass in Sullivan_MM.AplK, 109
AplNclass in Sullivan_MM.AlgebraPL, 75
AplN.Elementclass in Sullivan_MM.AlgebraPL, 78

base_ring()Hodge method, 96
boundary_operator()Hodge method, 96

coboundary_operator()Hodge method, 96
coexact_basis()Hodge method, 96
coexact_matrix()Hodge method, 97
cohomology_algebra()APLK method, 104
CohomologyClassclass in Sullivan_MM.AplK, 113

degeneracy()SimplicialAPL method, 88
dif()APLKElement method, 109
dif()AplN.Element method, 80
differential()AplN.Element method, 78
dup()APLKElement method, 110
dup()AplN.Element method, 80
dupont_contraction()APLKElement method, 110
dupont_operator()AplN.Element method, 80

exact_basis()Hodge method, 97
exact_matrix()Hodge method, 97
extended_basis()AplN method, 76
extended_graded_module()AplN method, 77

face()SimplicialAPL method, 89

graded_module_endomorphism()AplN method, 77

harmonic_basis()Hodge method, 98
harmonic_matrix()Hodge method, 98
Hodgeclass in Sullivan_MM.hodge, 95
hodge_cochain_decomposition()Hodge method, 99
hodge_decomposition()Hodge method, 99
homogeneous_components()AplN.Element method, 82

im_gens()MinimalModel method, 114
integral()APLKElement method, 112
integral()AplN.Element method, 82

is_formal()APLK method, 105

koszul()AplN.Element method, 83

laplacian()Hodge method, 100
laplacian()MinimalModel method, 115

minimal_model()APLK method, 106
MinimalModelclass in Sullivan_MM.AplK, 114
module
    Sullivan_MM.AlgebraPL, 75
    Sullivan_MM.AplK, 103
    Sullivan_MM.hodge, 93
    Sullivan_MM.SimplicialAPL, 87
MorphismFromQQToAPLKclass in Sullivan_MM.AplK, 113

n_th_object()SimplicialAPL method, 89
numerical_invariants()APLK method, 106

preimage()Hodge method, 100

reduce()AplN.Element method, 85
reduced_exterior_basis()AplN method, 78
representative()CohomologyClass method, 114

SimplicialAPLclass in Sullivan_MM.SimplicialAPL, 87
Sullivan_MM.AlgebraPL
    module, 75
Sullivan_MM.AplK
    module, 103
Sullivan_MM.hodge
    module, 93
Sullivan_MM.SimplicialAPL
    module, 87

whitney_form()AplN method, 78
whitney_map()SimplicialAPL method, 90

B

The homogeneous part of $\mathcal{M}^3$ for the $\mathcal{K}3$ surface

Following example IV.2, the list of generators of degree 3 of the minimal model of the $\mathcal{K}3$ surface are listed within their differentials (i.e, homogeneous elements of degree 4 in $\mathcal{M}$).

```
sage: from Sullivan_MM import *
sage: K = simplicial_complexes.K3Surface()
sage: W = APLK(K)
sage: M = W.minimal_model(i=3)
sage: M.domain()
Commutative Differential Graded Algebra with generators ('x2_0',
'x2_1', 'x2_2', 'x2_3', 'x2_4', 'x2_5', 'x2_6', 'x2_7', 'x2_8',
'x2_9', 'x2_10', 'x2_11', 'x2_12', 'x2_13', 'x2_14', 'x2_15',
'x2_16', 'x2_17', 'x2_18', 'x2_19', 'x2_20', 'x2_21', 'y3_0',
'y3_1', 'y3_2', 'y3_3', 'y3_4', 'y3_5', 'y3_6', 'y3_7', 'y3_8',
'y3_9', 'y3_10', 'y3_11', 'y3_12', 'y3_13', 'y3_14', 'y3_15',
'y3_16', 'y3_17', 'y3_18', 'y3_19', 'y3_20', 'y3_21', 'y3_22',
'y3_23', 'y3_24', 'y3_25', 'y3_26', 'y3_27', 'y3_28', 'y3_29',
'y3_30', 'y3_31', 'y3_32', 'y3_33', 'y3_34', 'y3_35', 'y3_36',
'y3_37', 'y3_38', 'y3_39', 'y3_40', 'y3_41', 'y3_42', 'y3_43',
'y3_44', 'y3_45', 'y3_46', 'y3_47', 'y3_48', 'y3_49', 'y3_50',
'y3_51', 'y3_52', 'y3_53', 'y3_54', 'y3_55', 'y3_56', 'y3_57',
'y3_58', 'y3_59', 'y3_60', 'y3_61', 'y3_62', 'y3_63', 'y3_64',
'y3_65', 'y3_66', 'y3_67', 'y3_68', 'y3_69', 'y3_70', 'y3_71',
'y3_72', 'y3_73', 'y3_74', 'y3_75', 'y3_76', 'y3_77', 'y3_78',
'y3_79', 'y3_80', 'y3_81', 'y3_82', 'y3_83', 'y3_84', 'y3_85',
'y3_86', 'y3_87', 'y3_88', 'y3_89', 'y3_90', 'y3_91', 'y3_92',
'y3_93', 'y3_94', 'y3_95', 'y3_96', 'y3_97', 'y3_98', 'y3_99',
'y3_100', 'y3_101', 'y3_102', 'y3_103', 'y3_104', 'y3_105',
'y3_106', 'y3_107', 'y3_108', 'y3_109', 'y3_110', 'y3_111',
'y3_112', 'y3_113', 'y3_114', 'y3_115', 'y3_116', 'y3_117',
'y3_118', 'y3_119', 'y3_120', 'y3_121', 'y3_122', 'y3_123',
'y3_124', 'y3_125', 'y3_126', 'y3_127', 'y3_128', 'y3_129',
'y3_130', 'y3_131', 'y3_132', 'y3_133', 'y3_134', 'y3_135',
'y3_136', 'y3_137', 'y3_138', 'y3_139', 'y3_140', 'y3_141',
'y3_142', 'y3_143', 'y3_144', 'y3_145', 'y3_146', 'y3_147',
```

```
'y3_148', 'y3_149', 'y3_150', 'y3_151', 'y3_152', 'y3_153',
'y3_154', 'y3_155', 'y3_156', 'y3_157', 'y3_158', 'y3_159',
'y3_160', 'y3_161', 'y3_162', 'y3_163', 'y3_164', 'y3_165',
'y3_166', 'y3_167', 'y3_168', 'y3_169', 'y3_170', 'y3_171',
'y3_172', 'y3_173', 'y3_174', 'y3_175', 'y3_176', 'y3_177',
'y3_178', 'y3_179', 'y3_180', 'y3_181', 'y3_182', 'y3_183',
'y3_184', 'y3_185', 'y3_186', 'y3_187', 'y3_188', 'y3_189',
'y3_190', 'y3_191', 'y3_192', 'y3_193', 'y3_194', 'y3_195',
'y3_196', 'y3_197', 'y3_198', 'y3_199', 'y3_200', 'y3_201',
'y3_202', 'y3_203', 'y3_204', 'y3_205', 'y3_206', 'y3_207',
'y3_208', 'y3_209', 'y3_210', 'y3_211', 'y3_212', 'y3_213',
'y3_214', 'y3_215', 'y3_216', 'y3_217', 'y3_218', 'y3_219',
'y3_220', 'y3_221', 'y3_222', 'y3_223', 'y3_224', 'y3_225',
'y3_226', 'y3_227', 'y3_228', 'y3_229', 'y3_230', 'y3_231',
'y3_232', 'y3_233', 'y3_234', 'y3_235', 'y3_236', 'y3_237',
'y3_238', 'y3_239', 'y3_240', 'y3_241', 'y3_242', 'y3_243',
'y3_244', 'y3_245', 'y3_246', 'y3_247', 'y3_248', 'y3_249',
'y3_250', 'y3_251') in degrees (2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3) over Rational Field with differential:
   x2_0 --> 0
   x2_1 --> 0
   x2_2 --> 0
   x2_3 --> 0
   x2_4 --> 0
   x2_5 --> 0
   x2_6 --> 0
   x2_7 --> 0
   x2_8 --> 0
   x2_9 --> 0
   x2_10 --> 0
   x2_11 --> 0
   x2_12 --> 0
   x2_13 --> 0
   x2_14 --> 0
```

```
x2_15 --> 0
x2_16 --> 0
x2_17 --> 0
x2_18 --> 0
x2_19 --> 0
x2_20 --> 0
x2_21 --> 0
y3_0 --> x2_0^2 - x2_21^2
y3_1 --> x2_0*x2_1
y3_2 --> x2_1^2 - x2_21^2
y3_3 --> x2_0*x2_2
y3_4 --> x2_1*x2_2 + 1/2*x2_21^2
y3_5 --> x2_2^2 - x2_21^2
y3_6 --> x2_0*x2_3 - 1/2*x2_21^2
y3_7 --> x2_1*x2_3 - 1/2*x2_21^2
y3_8 --> x2_2*x2_3
y3_9 --> x2_3^2 - x2_21^2
y3_10 --> x2_0*x2_4 + 3/2*x2_21^2
y3_11 --> x2_1*x2_4
y3_12 --> x2_2*x2_4 + 1/2*x2_21^2
y3_13 --> x2_3*x2_4 + 3/2*x2_21^2
y3_14 --> x2_4^2 - 5*x2_21^2
y3_15 --> x2_0*x2_5 + 3/2*x2_21^2
y3_16 --> x2_1*x2_5 + x2_21^2
y3_17 --> x2_2*x2_5
y3_18 --> x2_3*x2_5 + x2_21^2
y3_19 --> x2_4*x2_5 - 4*x2_21^2
y3_20 --> x2_5^2 - 4*x2_21^2
y3_21 --> x2_0*x2_6
y3_22 --> x2_1*x2_6
y3_23 --> x2_2*x2_6
y3_24 --> x2_3*x2_6
y3_25 --> x2_4*x2_6 + x2_21^2
y3_26 --> x2_5*x2_6 + 1/2*x2_21^2
y3_27 --> x2_6^2 - x2_21^2
y3_28 --> x2_0*x2_7 - 1/2*x2_21^2
y3_29 --> x2_1*x2_7
y3_30 --> x2_2*x2_7
y3_31 --> x2_3*x2_7 - 1/2*x2_21^2
y3_32 --> x2_4*x2_7 + 5/2*x2_21^2
y3_33 --> x2_5*x2_7 + 2*x2_21^2
y3_34 --> x2_6*x2_7 - x2_21^2
y3_35 --> x2_7^2 - 2*x2_21^2
y3_36 --> x2_0*x2_8
y3_37 --> x2_1*x2_8
y3_38 --> x2_2*x2_8
y3_39 --> x2_3*x2_8 + 1/2*x2_21^2
```

```
y3_40 --> x2_4*x2_8 - x2_21^2
y3_41 --> x2_5*x2_8 - x2_21^2
y3_42 --> x2_6*x2_8 + 1/2*x2_21^2
y3_43 --> x2_7*x2_8 + 1/2*x2_21^2
y3_44 --> x2_8^2 - x2_21^2
y3_45 --> x2_0*x2_9 - x2_21^2
y3_46 --> x2_1*x2_9 - 1/2*x2_21^2
y3_47 --> x2_2*x2_9
y3_48 --> x2_3*x2_9 - x2_21^2
y3_49 --> x2_4*x2_9 + 2*x2_21^2
y3_50 --> x2_5*x2_9 + 5/2*x2_21^2
y3_51 --> x2_6*x2_9 - 1/2*x2_21^2
y3_52 --> x2_7*x2_9 - x2_21^2
y3_53 --> x2_8*x2_9 + 1/2*x2_21^2
y3_54 --> x2_9^2 - 2*x2_21^2
y3_55 --> x2_0*x2_10 - 3/2*x2_21^2
y3_56 --> x2_1*x2_10 - 1/2*x2_21^2
y3_57 --> x2_2*x2_10
y3_58 --> x2_3*x2_10 - x2_21^2
y3_59 --> x2_4*x2_10 + 4*x2_21^2
y3_60 --> x2_5*x2_10 + 7/2*x2_21^2
y3_61 --> x2_6*x2_10 - 1/2*x2_21^2
y3_62 --> x2_7*x2_10 - 2*x2_21^2
y3_63 --> x2_8*x2_10 + x2_21^2
y3_64 --> x2_9*x2_10 - 2*x2_21^2
y3_65 --> x2_10^2 - 4*x2_21^2
y3_66 --> x2_0*x2_11 - 1/2*x2_21^2
y3_67 --> x2_1*x2_11 - 1/2*x2_21^2
y3_68 --> x2_2*x2_11
y3_69 --> x2_3*x2_11
y3_70 --> x2_4*x2_11 + 1/2*x2_21^2
y3_71 --> x2_5*x2_11 + 1/2*x2_21^2
y3_72 --> x2_6*x2_11 + 1/2*x2_21^2
y3_73 --> x2_7*x2_11
y3_74 --> x2_8*x2_11
y3_75 --> x2_9*x2_11 - x2_21^2
y3_76 --> x2_10*x2_11 - 1/2*x2_21^2
y3_77 --> x2_11^2 - x2_21^2
y3_78 --> x2_0*x2_12 + 1/2*x2_21^2
y3_79 --> x2_1*x2_12 - 1/2*x2_21^2
y3_80 --> x2_2*x2_12 + 1/2*x2_21^2
y3_81 --> x2_3*x2_12
y3_82 --> x2_4*x2_12 - 3/2*x2_21^2
y3_83 --> x2_5*x2_12 - 1/2*x2_21^2
y3_84 --> x2_6*x2_12
y3_85 --> x2_7*x2_12 + 1/2*x2_21^2
y3_86 --> x2_8*x2_12 - 1/2*x2_21^2
```

```
y3_87 --> x2_9*x2_12 + 1/2*x2_21^2
y3_88 --> x2_10*x2_12 + x2_21^2
y3_89 --> x2_11*x2_12
y3_90 --> x2_12^2 - x2_21^2
y3_91 --> x2_0*x2_13 - 1/2*x2_21^2
y3_92 --> x2_1*x2_13
y3_93 --> x2_2*x2_13
y3_94 --> x2_3*x2_13 - 1/2*x2_21^2
y3_95 --> x2_4*x2_13 + 3/2*x2_21^2
y3_96 --> x2_5*x2_13 + x2_21^2
y3_97 --> x2_6*x2_13
y3_98 --> x2_7*x2_13 - 1/2*x2_21^2
y3_99 --> x2_8*x2_13 + 1/2*x2_21^2
y3_100 --> x2_9*x2_13 - 1/2*x2_21^2
y3_101 --> x2_10*x2_13 - x2_21^2
y3_102 --> x2_11*x2_13
y3_103 --> x2_12*x2_13 + 1/2*x2_21^2
y3_104 --> x2_13^2 - x2_21^2
y3_105 --> x2_0*x2_14 - 1/2*x2_21^2
y3_106 --> x2_1*x2_14 - 1/2*x2_21^2
y3_107 --> x2_2*x2_14
y3_108 --> x2_3*x2_14 - 1/2*x2_21^2
y3_109 --> x2_4*x2_14 + x2_21^2
y3_110 --> x2_5*x2_14 + x2_21^2
y3_111 --> x2_6*x2_14 + 1/2*x2_21^2
y3_112 --> x2_7*x2_14
y3_113 --> x2_8*x2_14 + 1/2*x2_21^2
y3_114 --> x2_9*x2_14 - x2_21^2
y3_115 --> x2_10*x2_14 - x2_21^2
y3_116 --> x2_11*x2_14 - 1/2*x2_21^2
y3_117 --> x2_12*x2_14
y3_118 --> x2_13*x2_14 - 1/2*x2_21^2
y3_119 --> x2_14^2 - x2_21^2
y3_120 --> x2_0*x2_15 + x2_21^2
y3_121 --> x2_1*x2_15 + 1/2*x2_21^2
y3_122 --> x2_2*x2_15
y3_123 --> x2_3*x2_15 + 1/2*x2_21^2
y3_124 --> x2_4*x2_15 - 2*x2_21^2
y3_125 --> x2_5*x2_15 - 2*x2_21^2
y3_126 --> x2_6*x2_15 - 1/2*x2_21^2
y3_127 --> x2_7*x2_15 + 1/2*x2_21^2
y3_128 --> x2_8*x2_15 - 1/2*x2_21^2
y3_129 --> x2_9*x2_15 + x2_21^2
y3_130 --> x2_10*x2_15 + 2*x2_21^2
y3_131 --> x2_11*x2_15 + 1/2*x2_21^2
y3_132 --> x2_12*x2_15 - 1/2*x2_21^2
y3_133 --> x2_13*x2_15 + x2_21^2
```

```
y3_134 --> x2_14*x2_15 + x2_21^2
y3_135 --> x2_15^2 - 2*x2_21^2
y3_136 --> x2_0*x2_16 + 1/2*x2_21^2
y3_137 --> x2_1*x2_16 + 1/2*x2_21^2
y3_138 --> x2_2*x2_16
y3_139 --> x2_3*x2_16 + 1/2*x2_21^2
y3_140 --> x2_4*x2_16 - x2_21^2
y3_141 --> x2_5*x2_16 - x2_21^2
y3_142 --> x2_6*x2_16
y3_143 --> x2_7*x2_16 + 1/2*x2_21^2
y3_144 --> x2_8*x2_16
y3_145 --> x2_9*x2_16 + x2_21^2
y3_146 --> x2_10*x2_16 + x2_21^2
y3_147 --> x2_11*x2_16 + 1/2*x2_21^2
y3_148 --> x2_12*x2_16
y3_149 --> x2_13*x2_16
y3_150 --> x2_14*x2_16 + 1/2*x2_21^2
y3_151 --> x2_15*x2_16 - 1/2*x2_21^2
y3_152 --> x2_16^2 - x2_21^2
y3_153 --> x2_0*x2_17 + 1/2*x2_21^2
y3_154 --> x2_1*x2_17
y3_155 --> x2_2*x2_17 + 1/2*x2_21^2
y3_156 --> x2_3*x2_17 + 1/2*x2_21^2
y3_157 --> x2_4*x2_17 - 2*x2_21^2
y3_158 --> x2_5*x2_17 - 3/2*x2_21^2
y3_159 --> x2_6*x2_17 + 1/2*x2_21^2
y3_160 --> x2_7*x2_17 + x2_21^2
y3_161 --> x2_8*x2_17 - 1/2*x2_21^2
y3_162 --> x2_9*x2_17 + x2_21^2
y3_163 --> x2_10*x2_17 + 3/2*x2_21^2
y3_164 --> x2_11*x2_17
y3_165 --> x2_12*x2_17 - 1/2*x2_21^2
y3_166 --> x2_13*x2_17 + 1/2*x2_21^2
y3_167 --> x2_14*x2_17
y3_168 --> x2_15*x2_17 - 1/2*x2_21^2
y3_169 --> x2_16*x2_17 - 1/2*x2_21^2
y3_170 --> x2_17^2 - x2_21^2
y3_171 --> x2_0*x2_18 - x2_21^2
y3_172 --> x2_1*x2_18 - 1/2*x2_21^2
y3_173 --> x2_2*x2_18
y3_174 --> x2_3*x2_18 - 1/2*x2_21^2
y3_175 --> x2_4*x2_18 + 5/2*x2_21^2
y3_176 --> x2_5*x2_18 + 5/2*x2_21^2
y3_177 --> x2_6*x2_18 - 1/2*x2_21^2
y3_178 --> x2_7*x2_18 - 3/2*x2_21^2
y3_179 --> x2_8*x2_18 + 1/2*x2_21^2
y3_180 --> x2_9*x2_18 - 3/2*x2_21^2
```

```
y3_181 --> x2_10*x2_18 - 5/2*x2_21^2
y3_182 --> x2_11*x2_18 - 1/2*x2_21^2
y3_183 --> x2_12*x2_18 + 1/2*x2_21^2
y3_184 --> x2_13*x2_18 - 1/2*x2_21^2
y3_185 --> x2_14*x2_18 - 1/2*x2_21^2
y3_186 --> x2_15*x2_18 + x2_21^2
y3_187 --> x2_16*x2_18 + x2_21^2
y3_188 --> x2_17*x2_18 + x2_21^2
y3_189 --> x2_18^2 - 2*x2_21^2
y3_190 --> x2_0*x2_19 - 1/2*x2_21^2
y3_191 --> x2_1*x2_19
y3_192 --> x2_2*x2_19 - 1/2*x2_21^2
y3_193 --> x2_3*x2_19 - 1/2*x2_21^2
y3_194 --> x2_4*x2_19 + 3/2*x2_21^2
y3_195 --> x2_5*x2_19 + x2_21^2
y3_196 --> x2_6*x2_19
y3_197 --> x2_7*x2_19 - 1/2*x2_21^2
y3_198 --> x2_8*x2_19 + 1/2*x2_21^2
y3_199 --> x2_9*x2_19 - x2_21^2
y3_200 --> x2_10*x2_19 - x2_21^2
y3_201 --> x2_11*x2_19 - 1/2*x2_21^2
y3_202 --> x2_12*x2_19 + 1/2*x2_21^2
y3_203 --> x2_13*x2_19 - 1/2*x2_21^2
y3_204 --> x2_14*x2_19 - 1/2*x2_21^2
y3_205 --> x2_15*x2_19 + 1/2*x2_21^2
y3_206 --> x2_16*x2_19 + 1/2*x2_21^2
y3_207 --> x2_17*x2_19 + 1/2*x2_21^2
y3_208 --> x2_18*x2_19 - 1/2*x2_21^2
y3_209 --> x2_19^2 - x2_21^2
y3_210 --> x2_0*x2_20 + 1/2*x2_21^2
y3_211 --> x2_1*x2_20 + 1/2*x2_21^2
y3_212 --> x2_2*x2_20
y3_213 --> x2_3*x2_20 + 1/2*x2_21^2
y3_214 --> x2_4*x2_20 - 2*x2_21^2
y3_215 --> x2_5*x2_20 - 3/2*x2_21^2
y3_216 --> x2_6*x2_20
y3_217 --> x2_7*x2_20 + x2_21^2
y3_218 --> x2_8*x2_20 - 1/2*x2_21^2
y3_219 --> x2_9*x2_20 + x2_21^2
y3_220 --> x2_10*x2_20 + 3/2*x2_21^2
y3_221 --> x2_11*x2_20
y3_222 --> x2_12*x2_20
y3_223 --> x2_13*x2_20 + 1/2*x2_21^2
y3_224 --> x2_14*x2_20 + 1/2*x2_21^2
y3_225 --> x2_15*x2_20 - x2_21^2
y3_226 --> x2_16*x2_20 - 1/2*x2_21^2
y3_227 --> x2_17*x2_20 - 1/2*x2_21^2
```

```
y3_228 --> x2_18*x2_20 + x2_21^2
y3_229 --> x2_19*x2_20 + 1/2*x2_21^2
y3_230 --> x2_20^2 - x2_21^2
y3_231 --> x2_0*x2_21 + 1/2*x2_21^2
y3_232 --> x2_1*x2_21 + 1/2*x2_21^2
y3_233 --> x2_2*x2_21
y3_234 --> x2_3*x2_21 + 1/2*x2_21^2
y3_235 --> x2_4*x2_21 - 3/2*x2_21^2
y3_236 --> x2_5*x2_21 - 3/2*x2_21^2
y3_237 --> x2_6*x2_21
y3_238 --> x2_7*x2_21 + 1/2*x2_21^2
y3_239 --> x2_8*x2_21 - 1/2*x2_21^2
y3_240 --> x2_9*x2_21 + x2_21^2
y3_241 --> x2_10*x2_21 + 3/2*x2_21^2
y3_242 --> x2_11*x2_21
y3_243 --> x2_12*x2_21
y3_244 --> x2_13*x2_21 + 1/2*x2_21^2
y3_245 --> x2_14*x2_21 + 1/2*x2_21^2
y3_246 --> x2_15*x2_21 - x2_21^2
y3_247 --> x2_16*x2_21 - 1/2*x2_21^2
y3_248 --> x2_17*x2_21 - 1/2*x2_21^2
y3_249 --> x2_18*x2_21 + x2_21^2
y3_250 --> x2_19*x2_21 + 1/2*x2_21^2
y3_251 --> x2_20*x2_21 - 1/2*x2_21^2
```

C

Example of an element $p \in \mathcal{A}^3_{\mathcal{PL}}(\mathcal{K}3)$

Following example IV.2, we compute a generating set of the Whitney cochain complex $\mathcal{W}(\mathcal{K}3)$ and show the basic operations we can do with such elements.

First, we inject to our sage's session the whitney forms from degree 0 to the dimension of the complex ($dim(\mathcal{K}3) = 4$):

```
sage: from Sullivan_MM import *
sage: K = simplicial_complexes.K3Surface()
sage: W = APLK(K)
sage: W.inject_variables()
Getting generators of degree  0 ...
Getting generators of degree  1 ...
Getting generators of degree  2 ...
Getting generators of degree  3 ...
Getting generators of degree  4 ...
Defining h0_0, c0_0, c0_1, c0_2, c0_3, c0_4, c0_5, c0_6, c0_7,
c0_8, c0_9, c0_10, c0_11, c0_12, c0_13, c0_14, e1_0, e1_1,
e1_2, e1_3, e1_4, e1_5, e1_6, e1_7, e1_8, e1_9, e1_10, e1_11,
e1_12, e1_13, e1_14, c1_0, c1_1, c1_2, c1_3, c1_4, c1_5, c1_6,
c1_7, c1_8, c1_9, c1_10, c1_11, c1_12, c1_13, c1_14, c1_15,
c1_16, c1_17, c1_18, c1_19, c1_20, c1_21, c1_22, c1_23, c1_24,
c1_25, c1_26, c1_27, c1_28, c1_29, c1_30, c1_31, c1_32, c1_33,
c1_34, c1_35, c1_36, c1_37, c1_38, c1_39, c1_40, c1_41, c1_42,
c1_43, c1_44, c1_45, c1_46, c1_47, c1_48, c1_49, c1_50, c1_51,
c1_52, c1_53, c1_54, c1_55, c1_56, c1_57, c1_58, c1_59, c1_60,
c1_61, c1_62, c1_63, c1_64, c1_65, c1_66, c1_67, c1_68, c1_69,
c1_70, c1_71, c1_72, c1_73, c1_74, c1_75, c1_76, c1_77, c1_78,
c1_79, c1_80, c1_81, c1_82, c1_83, c1_84, c1_85, c1_86, c1_87,
c1_88, c1_89, c1_90, c1_91, c1_92, c1_93, c1_94, c1_95, c1_96,
c1_97, c1_98, c1_99, c1_100, c1_101, c1_102, c1_103, c1_104,
h2_0, h2_1, h2_2, h2_3, h2_4, h2_5, h2_6, h2_7, h2_8, h2_9,
h2_10, h2_11, h2_12, h2_13, h2_14, h2_15, h2_16, h2_17, h2_18,
h2_19, h2_20, h2_21, e2_0, e2_1, e2_2, e2_3, e2_4, e2_5, e2_6,
e2_7, e2_8, e2_9, e2_10, e2_11, e2_12, e2_13, e2_14, e2_15,
e2_16, e2_17, e2_18, e2_19, e2_20, e2_21, e2_22, e2_23, e2_24,
```

```
e2_25, e2_26, e2_27, e2_28, e2_29, e2_30, e2_31, e2_32, e2_33,
e2_34, e2_35, e2_36, e2_37, e2_38, e2_39, e2_40, e2_41, e2_42,
e2_43, e2_44, e2_45, e2_46, e2_47, e2_48, e2_49, e2_50, e2_51,
e2_52, e2_53, e2_54, e2_55, e2_56, e2_57, e2_58, e2_59, e2_60,
e2_61, e2_62, e2_63, e2_64, e2_65, e2_66, e2_67, e2_68, e2_69,
e2_70, e2_71, e2_72, e2_73, e2_74, e2_75, e2_76, e2_77, e2_78,
e2_79, e2_80, e2_81, e2_82, e2_83, e2_84, e2_85, e2_86, e2_87,
e2_88, e2_89, e2_90, e2_91, e2_92, e2_93, e2_94, e2_95, e2_96,
e2_97, e2_98, e2_99, e2_100, e2_101, e2_102, e2_103, e2_104,
c2_0, c2_1, c2_2, c2_3, c2_4, c2_5, c2_6, c2_7, c2_8, c2_9,
c2_10, c2_11, c2_12, c2_13, c2_14, c2_15, c2_16, c2_17, c2_18,
c2_19, c2_20, c2_21, c2_22, c2_23, c2_24, c2_25, c2_26, c2_27,
c2_28, c2_29, c2_30, c2_31, c2_32, c2_33, c2_34, c2_35, c2_36,
c2_37, c2_38, c2_39, c2_40, c2_41, c2_42, c2_43, c2_44, c2_45,
c2_46, c2_47, c2_48, c2_49, c2_50, c2_51, c2_52, c2_53, c2_54,
c2_55, c2_56, c2_57, c2_58, c2_59, c2_60, c2_61, c2_62, c2_63,
c2_64, c2_65, c2_66, c2_67, c2_68, c2_69, c2_70, c2_71, c2_72,
c2_73, c2_74, c2_75, c2_76, c2_77, c2_78, c2_79, c2_80, c2_81,
c2_82, c2_83, c2_84, c2_85, c2_86, c2_87, c2_88, c2_89, c2_90,
c2_91, c2_92, c2_93, c2_94, c2_95, c2_96, c2_97, c2_98, c2_99,
c2_100, c2_101, c2_102, c2_103, c2_104, c2_105, c2_106, c2_107,
c2_108, c2_109, c2_110, c2_111, c2_112, c2_113, c2_114, c2_115,
c2_116, c2_117, c2_118, c2_119, c2_120, c2_121, c2_122, c2_123,
c2_124, c2_125, c2_126, c2_127, c2_128, c2_129, c2_130, c2_131,
c2_132, c2_133, c2_134, c2_135, c2_136, c2_137, c2_138, c2_139,
c2_140, c2_141, c2_142, c2_143, c2_144, c2_145, c2_146, c2_147,
c2_148, c2_149, c2_150, c2_151, c2_152, c2_153, c2_154, c2_155,
c2_156, c2_157, c2_158, c2_159, c2_160, c2_161, c2_162, c2_163,
c2_164, c2_165, c2_166, c2_167, c2_168, c2_169, c2_170, c2_171,
c2_172, c2_173, c2_174, c2_175, c2_176, c2_177, c2_178, c2_179,
c2_180, c2_181, c2_182, c2_183, c2_184, c2_185, c2_186, c2_187,
c2_188, c2_189, c2_190, c2_191, c2_192, c2_193, c2_194, c2_195,
c2_196, c2_197, c2_198, c2_199, c2_200, c2_201, c2_202, c2_203,
c2_204, c2_205, c2_206, c2_207, c2_208, c2_209, c2_210, c2_211,
c2_212, c2_213, c2_214, c2_215, c2_216, c2_217, c2_218, c2_219,
c2_220, c2_221, c2_222, c2_223, c2_224, c2_225, c2_226, c2_227,
c2_228, c2_229, c2_230, c2_231, c2_232, c2_233, c2_234, c2_235,
c2_236, c2_237, c2_238, c2_239, c2_240, c2_241, c2_242, c2_243,
c2_244, c2_245, c2_246, c2_247, c2_248, c2_249, c2_250, c2_251,
c2_252, c2_253, c2_254, c2_255, c2_256, c2_257, c2_258, c2_259,
c2_260, c2_261, c2_262, c2_263, c2_264, c2_265, c2_266, c2_267,
c2_268, c2_269, c2_270, c2_271, c2_272, c2_273, c2_274, c2_275,
c2_276, c2_277, c2_278, c2_279, c2_280, c2_281, c2_282, c2_283,
2_284, c2_285, c2_286, c2_287, c2_288, c2_289, c2_290, c2_291,
c2_292, c2_293, c2_294, c2_295, c2_296, c2_297, c2_298, c2_299,
c2_300, c2_301, c2_302, c2_303, c2_304, c2_305, c2_306, c2_307,
c2_308, c2_309, c2_310, c2_311, c2_312, c2_313, c2_314, c2_315,
```

c2_316, c2_317, c2_318, c2_319, c2_320, c2_321, c2_322, c2_323,
c2_324, c2_325, c2_326, c2_327, c2_328, c2_329, c2_330, c2_331,
c2_332, c2_333, c2_334, c2_335, c2_336, c2_337, c2_338, c2_339,
c2_340, c2_341, c2_342, c2_343, c2_344, c2_345, c2_346, c2_347,
c2_348, c2_349, c2_350, c2_351, c2_352, c2_353, c2_354, c2_355,
c2_356, c2_357, c2_358, c2_359, c2_360, c2_361, c2_362, c2_363,
c2_364, c2_365, c2_366, c2_367, c2_368, c2_369, c2_370, c2_371,
c2_372, c2_373, c2_374, c2_375, c2_376, c2_377, c2_378, c2_379,
c2_380, c2_381, c2_382, c2_383, c2_384, c2_385, c2_386, c2_387,
c2_388, c2_389, c2_390, c2_391, c2_392, c2_393, c2_394, c2_395,
c2_396, c2_397, c2_398, c2_399, c2_400, c2_401, c2_402, c2_403,
c2_404, c2_405, c2_406, c2_407, c2_408, c2_409, c2_410, c2_411,
c2_412, c2_413, c2_414, c2_415, c2_416, c2_417, c2_418, c2_419,
c2_420, c2_421, c2_422, c2_423, c2_424, c2_425, c2_426, c2_427,
c2_428, c2_429, c2_430, c2_431, c2_432, e3_0, e3_1, e3_2, e3_3,
e3_4, e3_5, e3_6, e3_7, e3_8, e3_9, e3_10, e3_11, e3_12, e3_13,
e3_14, e3_15, e3_16, e3_17, e3_18, e3_19, e3_20, e3_21, e3_22,
e3_23, e3_24, e3_25, e3_26, e3_27, e3_28, e3_29, e3_30, e3_31,
e3_32, e3_33, e3_34, e3_35, e3_36, e3_37, e3_38, e3_39, e3_40,
e3_41, e3_42, e3_43, e3_44, e3_45, e3_46, e3_47, e3_48, e3_49,
e3_50, e3_51, e3_52, e3_53, e3_54, e3_55, e3_56, e3_57, e3_58,
e3_59, e3_60, e3_61, e3_62, e3_63, e3_64, e3_65, e3_66, e3_67,
e3_68, e3_69, e3_70, e3_71, e3_72, e3_73, e3_74, e3_75, e3_76,
e3_77, e3_78, e3_79, e3_80, e3_81, e3_82, e3_83, e3_84, e3_85,
e3_86, e3_87, e3_88, e3_89, e3_90, e3_91, e3_92, e3_93, e3_94,
e3_95, e3_96, e3_97, e3_98, e3_99, e3_100, e3_101, e3_102,
e3_103, e3_104, e3_105, e3_106, e3_107, e3_108, e3_109, e3_110,
e3_111, e3_112, e3_113, e3_114, e3_115, e3_116, e3_117, e3_118,
e3_119, e3_120, e3_121, e3_122, e3_123, e3_124, e3_125, e3_126,
e3_127, e3_128, e3_129, e3_130, e3_131, e3_132, e3_133, e3_134,
e3_135, e3_136, e3_137, e3_138, e3_139, e3_140, e3_141, e3_142,
e3_143, e3_144, e3_145, e3_146, e3_147, e3_148, e3_149, e3_150,
e3_151, e3_152, e3_153, e3_154, e3_155, e3_156, e3_157, e3_158,
e3_159, e3_160, e3_161, e3_162, e3_163, e3_164, e3_165, e3_166,
e3_167, e3_168, e3_169, e3_170, e3_171, e3_172, e3_173, e3_174,
e3_175, e3_176, e3_177, e3_178, e3_179, e3_180, e3_181, e3_182,
e3_183, e3_184, e3_185, e3_186, e3_187, e3_188, e3_189, e3_190,
e3_191, e3_192, e3_193, e3_194, e3_195, e3_196, e3_197, e3_198,
e3_199, e3_200, e3_201, e3_202, e3_203, e3_204, e3_205, e3_206,
e3_207, e3_208, e3_209, e3_210, e3_211, e3_212, e3_213, e3_214,
e3_215, e3_216, e3_217, e3_218, e3_219, e3_220, e3_221, e3_222,
e3_223, e3_224, e3_225, e3_226, e3_227, e3_228, e3_229, e3_230,
e3_231, e3_232, e3_233, e3_234, e3_235, e3_236, e3_237, e3_238,
e3_239, e3_240, e3_241, e3_242, e3_243, e3_244, e3_245, e3_246,
e3_247, e3_248, e3_249, e3_250, e3_251, e3_252, e3_253, e3_254,
e3_255, e3_256, e3_257, e3_258, e3_259, e3_260, e3_261, e3_262,
e3_263, e3_264, e3_265, e3_266, e3_267, e3_268, e3_269, e3_270,

e3_271, e3_272, e3_273, e3_274, e3_275, e3_276, e3_277, e3_278,
e3_279, e3_280, e3_281, e3_282, e3_283, e3_284, e3_285, e3_286,
e3_287, e3_288, e3_289, e3_290, e3_291, e3_292, e3_293, e3_294,
e3_295, e3_296, e3_297, e3_298, e3_299, e3_300, e3_301, e3_302,
e3_303, e3_304, e3_305, e3_306, e3_307, e3_308, e3_309, e3_310,
e3_311, e3_312, e3_313, e3_314, e3_315, e3_316, e3_317, e3_318,
e3_319, e3_320, e3_321, e3_322, e3_323, e3_324, e3_325, e3_326,
e3_327, e3_328, e3_329, e3_330, e3_331, e3_332, e3_333, e3_334,
e3_335, e3_336, e3_337, e3_338, e3_339, e3_340, e3_341, e3_342,
e3_343, e3_344, e3_345, e3_346, e3_347, e3_348, e3_349, e3_350,
e3_351, e3_352, e3_353, e3_354, e3_355, e3_356, e3_357, e3_358,
e3_359, e3_360, e3_361, e3_362, e3_363, e3_364, e3_365, e3_366,
e3_367, e3_368, e3_369, e3_370, e3_371, e3_372, e3_373, e3_374,
e3_375, e3_376, e3_377, e3_378, e3_379, e3_380, e3_381, e3_382,
e3_383, e3_384, e3_385, e3_386, e3_387, e3_388, e3_389, e3_390,
e3_391, e3_392, e3_393, e3_394, e3_395, e3_396, e3_397, e3_398,
e3_399, e3_400, e3_401, e3_402, e3_403, e3_404, e3_405, e3_406,
e3_407, e3_408, e3_409, e3_410, e3_411, e3_412, e3_413, e3_414,
e3_415, e3_416, e3_417, e3_418, e3_419, e3_420, e3_421, e3_422,
e3_423, e3_424, e3_425, e3_426, e3_427, e3_428, e3_429, e3_430,
e3_431, e3_432, c3_0, c3_1, c3_2, c3_3, c3_4, c3_5, c3_6, c3_7,
c3_8, c3_9, c3_10, c3_11, c3_12, c3_13, c3_14, c3_15, c3_16,
c3_17, c3_18, c3_19, c3_20, c3_21, c3_22, c3_23, c3_24, c3_25,
c3_26, c3_27, c3_28, c3_29, c3_30, c3_31, c3_32, c3_33, c3_34,
c3_35, c3_36, c3_37, c3_38, c3_39, c3_40, c3_41, c3_42, c3_43,
c3_44, c3_45, c3_46, c3_47, c3_48, c3_49, c3_50, c3_51, c3_52,
c3_53, c3_54, c3_55, c3_56, c3_57, c3_58, c3_59, c3_60, c3_61,
c3_62, c3_63, c3_64, c3_65, c3_66, c3_67, c3_68, c3_69, c3_70,
c3_71, c3_72, c3_73, c3_74, c3_75, c3_76, c3_77, c3_78, c3_79,
c3_80, c3_81, c3_82, c3_83, c3_84, c3_85, c3_86, c3_87, c3_88,
c3_89, c3_90, c3_91, c3_92, c3_93, c3_94, c3_95, c3_96, c3_97,
c3_98, c3_99, c3_100, c3_101, c3_102, c3_103, c3_104, c3_105,
c3_106, c3_107, c3_108, c3_109, c3_110, c3_111, c3_112, c3_113,
c3_114, c3_115, c3_116, c3_117, c3_118, c3_119, c3_120, c3_121,
c3_122, c3_123, c3_124, c3_125, c3_126, c3_127, c3_128, c3_129,
c3_130, c3_131, c3_132, c3_133, c3_134, c3_135, c3_136, c3_137,
c3_138, c3_139, c3_140, c3_141, c3_142, c3_143, c3_144, c3_145,
c3_146, c3_147, c3_148, c3_149, c3_150, c3_151, c3_152, c3_153,
c3_154, c3_155, c3_156, c3_157, c3_158, c3_159, c3_160, c3_161,
c3_162, c3_163, c3_164, c3_165, c3_166, c3_167, c3_168, c3_169,
c3_170, c3_171, c3_172, c3_173, c3_174, c3_175, c3_176, c3_177,
c3_178, c3_179, c3_180, c3_181, c3_182, c3_183, c3_184, c3_185,
c3_186, c3_187, c3_188, c3_189, c3_190, c3_191, c3_192, c3_193,
c3_194, c3_195, c3_196, c3_197, c3_198, c3_199, c3_200, c3_201,
c3_202, c3_203, c3_204, c3_205, c3_206, c3_207, c3_208, c3_209,
c3_210, c3_211, c3_212, c3_213, c3_214, c3_215, c3_216, c3_217,
c3_218, c3_219, c3_220, c3_221, c3_222, c3_223, c3_224, c3_225,

c3_226, c3_227, c3_228, c3_229, c3_230, c3_231, c3_232, c3_233,
c3_234, c3_235, c3_236, c3_237, c3_238, c3_239, c3_240, c3_241,
c3_242, c3_243, c3_244, c3_245, c3_246, c3_247, c3_248, c3_249,
c3_250, c3_251, c3_252, c3_253, c3_254, c3_255, c3_256, c3_257,
c3_258, c3_259, c3_260, c3_261, c3_262, c3_263, c3_264, c3_265,
c3_266, c3_267, c3_268, c3_269, c3_270, c3_271, c3_272, c3_273,
c3_274, c3_275, c3_276, c3_277, c3_278, c3_279, c3_280, c3_281,
c3_282, c3_283, c3_284, c3_285, c3_286, h4_0, e4_0, e4_1, e4_2,
e4_3, e4_4, e4_5, e4_6, e4_7, e4_8, e4_9, e4_10, e4_11, e4_12,
e4_13, e4_14, e4_15, e4_16, e4_17, e4_18, e4_19, e4_20, e4_21,
e4_22, e4_23, e4_24, e4_25, e4_26, e4_27, e4_28, e4_29, e4_30,
e4_31, e4_32, e4_33, e4_34, e4_35, e4_36, e4_37, e4_38, e4_39,
e4_40, e4_41, e4_42, e4_43, e4_44, e4_45, e4_46, e4_47, e4_48,
e4_49, e4_50, e4_51, e4_52, e4_53, e4_54, e4_55, e4_56, e4_57,
e4_58, e4_59, e4_60, e4_61, e4_62, e4_63, e4_64, e4_65, e4_66,
e4_67, e4_68, e4_69, e4_70, e4_71, e4_72, e4_73, e4_74, e4_75,
e4_76, e4_77, e4_78, e4_79, e4_80, e4_81, e4_82, e4_83, e4_84,
e4_85, e4_86, e4_87, e4_88, e4_89, e4_90, e4_91, e4_92, e4_93,
e4_94, e4_95, e4_96, e4_97, e4_98, e4_99, e4_100, e4_101, e4_102,
e4_103, e4_104, e4_105, e4_106, e4_107, e4_108, e4_109, e4_110,
e4_111, e4_112, e4_113, e4_114, e4_115, e4_116, e4_117, e4_118,
e4_119, e4_120, e4_121, e4_122, e4_123, e4_124, e4_125, e4_126,
e4_127, e4_128, e4_129, e4_130, e4_131, e4_132, e4_133, e4_134,
e4_135, e4_136, e4_137, e4_138, e4_139, e4_140, e4_141, e4_142,
e4_143, e4_144, e4_145, e4_146, e4_147, e4_148, e4_149, e4_150,
e4_151, e4_152, e4_153, e4_154, e4_155, e4_156, e4_157, e4_158,
e4_159, e4_160, e4_161, e4_162, e4_163, e4_164, e4_165, e4_166,
e4_167, e4_168, e4_169, e4_170, e4_171, e4_172, e4_173, e4_174,
e4_175, e4_176, e4_177, e4_178, e4_179, e4_180, e4_181, e4_182,
e4_183, e4_184, e4_185, e4_186, e4_187, e4_188, e4_189, e4_190,
e4_191, e4_192, e4_193, e4_194, e4_195, e4_196, e4_197, e4_198,
e4_199, e4_200, e4_201, e4_202, e4_203, e4_204, e4_205, e4_206,
e4_207, e4_208, e4_209, e4_210, e4_211, e4_212, e4_213, e4_214,
e4_215, e4_216, e4_217, e4_218, e4_219, e4_220, e4_221, e4_222,
e4_223, e4_224, e4_225, e4_226, e4_227, e4_228, e4_229, e4_230,
e4_231, e4_232, e4_233, e4_234, e4_235, e4_236, e4_237, e4_238,
e4_239, e4_240, e4_241, e4_242, e4_243, e4_244, e4_245, e4_246,
e4_247, e4_248, e4_249, e4_250, e4_251, e4_252, e4_253, e4_254,
e4_255, e4_256, e4_257, e4_258, e4_259, e4_260, e4_261, e4_262,
e4_263, e4_264, e4_265, e4_266, e4_267, e4_268, e4_269, e4_270,
e4_271, e4_272, e4_273, e4_274, e4_275, e4_276, e4_277, e4_278,
e4_279, e4_280, e4_281, e4_282, e4_283, e4_284, e4_285, e4_286

For a variable ai_j, the notation means:

- a can be h (harmonic form), e (exact form) or c (coexact form)
- i refers to the degree of the form
- j is the index at the basis of Whitney forms of degree i.

So, for example, we can ask if a Whitney form is harmonic or not:

```
sage: h2_0.is_harmonic()
True
```

Remark **(C.0.1).** WARNING: the labeling is taking into account the projection onto the cochain complex, what means that multiple elements can share the same label although internaly are not equal. Use this representation carefully.

Here we show examples of cochain algebra operations:

```
sage: h2_0*h2_1
[1/6]*[(-1)*e4_30 + e4_81 + e4_83 + e4_86 + (-1)*e4_92 +
    (-2)*e4_133 + e4_135 + e4_144 + (-1)*e4_152 + (-2)*e4_153 +
    (-1)*e4_165 + (-1)*e4_175 + (-1)*e4_177 + e4_179 + (-1)*e4_191 +
    (-1)*e4_202 + e4_203 + (-1)*e4_204 + e4_208 + (-2)*e4_218 +
    (-1)*e4_220 + (-1)*e4_221 + e4_225 + e4_226 + (-1)*e4_227 +
    e4_228 + (2)*e4_230 + (-1)*e4_232 + e4_233 + e4_234 + e4_242 +
    e4_243 + (-1)*e4_245 + (-1)*e4_250 + e4_252 + (-1)*e4_258 +
    e4_260 + (-1)*e4_265 + (-1)*e4_284]
sage: p = c1_2*c2_9
sage: p
[1/12]*[e3_0 + e3_91 + e3_94 + e3_95 + e3_97 + e3_99 +
    e3_102 + (-1)*e3_104 + (-1)*e3_107 + (-1)*e3_110 +
    (-1)*e3_111 + (-1)*e3_112 + e3_114 + e3_115 + e3_123 +
    (-1)*e3_138 + (-1)*e3_139 + e3_180 + e3_188 + (-1)*e3_195 +
    (-1)*e3_198 + e3_235 + e3_236 + e3_243 + e3_292 + e3_294 +
    e3_300 + e3_301 + e3_307 + (-1)*e3_316 + (-1)*e3_317 +
    (-1)*e3_318 + (-1)*e3_319 + (-1)*e3_323 + (-1)*e3_326 +
    (-1)*e3_355 + (-1)*e3_358] + [1/12]*[(-1)*c3_31 + c3_40 +
    (-1)*c3_81 + (-1)*c3_83 + c3_133 + (-1)*c3_140 + c3_177 +
    c3_202 + (-1)*c3_228 + (-1)*c3_230 + c3_232]
sage: p.degree()
3
sage: p.dif() == 0
False
sage: dp = p.dif()
sage: dp
[1/12]*[(-1)*e4_31 + e4_40 + (-1)*e4_81 + (-1)*e4_83 +
        e4_133 + (-1)*e4_140 + e4_177 + e4_202 + (-1)*e4_228 +
(-1)*e4_230 + e4_232]
sage: r = dp.solve_primitive()
sage: r.dif() == dp
True
sage: p.to_AplK()
{(2, 3, 5, 13): 2*t0*t3*y0*y1*y2 - 2*t0*t2*y0*y1*y3 +
                2*t0*t1*y0*y2*y3 - 2*t0^2*y1*y2*y3,
 (2, 3, 4, 5, 13): 2*t0*t4*y0*y1*y3 - 2*t0*t3*y0*y1*y4 +
                2*t0*t1*y0*y3*y4 - 2*t0^2*y1*y3*y4,
 (2, 3, 5, 6, 13): 2*t0*t4*y0*y1*y2 - 2*t0*t2*y0*y1*y4 +
```

```
                2*t0*t1*y0*y2*y4 - 2*t0^2*y1*y2*y4}
sage: dp.to_AplK()
{(2, 3, 4, 5, 13): (10*t1 + 10*t2 + 10*t3 +
                    10*t4 - 10)*y1*y2*y3*y4,
 (2, 3, 5, 6, 13): (-10*t1 - 10*t2 - 10*t3 -
                    10*t4 + 10)*y1*y2*y3*y4}
```

And we can compute Dupont's contraction in an easy way:

```
sage: p
[1/12]*[e3_0 + e3_91 + e3_94 + e3_95 + e3_97 + e3_99 +
    e3_102 + (-1)*e3_104 + (-1)*e3_107 + (-1)*e3_110 +
    (-1)*e3_111 + (-1)*e3_112 + e3_114 + e3_115 + e3_123 +
    (-1)*e3_138 + (-1)*e3_139 + e3_180 + e3_188 +
    (-1)*e3_195 + (-1)*e3_198 + e3_235 + e3_236 + e3_243 +
    e3_292 + e3_294 + e3_300 + e3_301 + e3_307 + (-1)*e3_316 +
    (-1)*e3_317 + (-1)*e3_318 + (-1)*e3_319 +
    (-1)*e3_323 + (-1)*e3_326 + (-1)*e3_355 + (-1)*e3_358] +
    [1/12]*[(-1)*c3_31 + c3_40 + (-1)*c3_81 + (-1)*c3_83 +
    c3_133 + (-1)*c3_140 + c3_177 + c3_202 + (-1)*c3_228 +
    (-1)*c3_230 + c3_232]
sage: (h,e,c) = p.hodge_whitney_decomposition()
sage: h
0
sage: e
[1/12]*[e3_0 + e3_91 + e3_94 + e3_95 + e3_97 + e3_99 +
    e3_102 + (-1)*e3_104 + (-1)*e3_107 + (-1)*e3_110 +
    (-1)*e3_111 + (-1)*e3_112 + e3_114 + e3_115 +
    e3_123 + (-1)*e3_138 +  (-1)*e3_139 + e3_180 +
    e3_188 + (-1)*e3_195 + (-1)*e3_198 + e3_235 +
    e3_236 + e3_243 + e3_292 + e3_294 + e3_300 + e3_301 +
    e3_307 + (-1)*e3_316 + (-1)*e3_317 + (-1)*e3_318 +
    (-1)*e3_319 + (-1)*e3_323 + (-1)*e3_326 +
    (-1)*e3_355 + (-1)*e3_358]
sage: c
[1/12]*[(-1)*c3_31 + c3_40 + (-1)*c3_81 + (-1)*c3_83 +
    c3_133 + (-1)*c3_140 + c3_177 + c3_202 + (-1)*c3_228 +
    (-1)*c3_230 + c3_232]
sage: wi = h+e+c
sage: wi-p == p.dup().dif()+p.dif().dup()
True
sage: p.dup().to_AplK()
{(2, 3, 5, 13): (-1/2*t1*t3 - 1/2*t2*t3 - 1/2*t3^2 +
                 1/2*t3)*y1*y2 + (1/2*t1*t2 + 1/2*t2^2 +
                 1/2*t2*t3 - 1/2*t2)*y1*y3 + (-1/2*t1^2 -
                 1/2*t1*t2 - 1/2*t1*t3 + 1/2*t1)*y2*y3,
 (2, 3, 4, 5, 13): (-1/2*t1*t4 - 1/2*t2*t4 - 1/2*t3*t4 -
                 1/2*t4^2 + 1/2*t4)*y1*y3 + (1/2*t1*t3 +
                 1/2*t2*t3 + 1/2*t3^2 + 1/2*t3*t4 -
```

```
                1/2*t3)*y1*y4 + (-1/2*t1^2 -
                1/2*t1*t2 - 1/2*t1*t3 - 1/2*t1*t4 +
                1/2*t1)*y3*y4,
 (2, 3, 5, 6, 13): (-1/2*t1*t4 - 1/2*t2*t4 - 1/2*t3*t4 -
                 1/2*t4^2 + 1/2*t4)*y1*y2 + (1/2*t1*t2 +
                 1/2*t2^2 + 1/2*t2*t3 + 1/2*t2*t4 -
                 1/2*t2)*y1*y4 + (-1/2*t1^2 - 1/2*t1*t2 -
                 1/2*t1*t3 - 1/2*t1*t4 + 1/2*t1)*y2*y4}
```

To illustrate how complicated the output is, we show an example of two homogeneous elements of $\mathcal{A}_{\mathcal{PL}}(\mathcal{K}3)$ of degrees 3 and 4.

During the computation of the minimal model of the $K3$ surface, at the injectivity step (see section III.2), the method finds an element of degree 4 (in particular, the element $x_0^2x_4^2+\frac{3}{2}(x_{21}^2)^2 \in \mathcal{M}^4$) that should kill in order to make the quasi-isomorphism φ^* injective at degree 4. We can check if φ is a morphism of $CDGA$s as follows:

```
sage: from Sullivan_MM import *
sage: K = simplicial_complexes.K3Surface()
sage: W = APLK(K)
sage: M = W.minimal_model(i=3)
sage: A = M.domain()
sage: A.basis(3)[10]
y3_10
sage: y3_10 = A.basis(3)[10]
sage:  M.phi(y3_10).dif() ==h2_0*h2_4+(3/2)*(h2_21)**2
True
sage:  M.phi(y3_10) == (h2_0*h2_4+(3/2)*(h2_21)**2).solve_primitive()
True
```

Finally, the image of this element by the morphism φ is represented internally as the dictionary:

```
sage: M.phi(y3_10)
[1/6]*[(-10)*e3_0 + (7)*e3_1 + e3_2 + (-1)*e3_3 + e3_5 + e3_6 +
(-7)*e3_9 + (6)*e3_10 + e3_11 + (4)*e3_13 + (-3)*e3_14 + e3_16 +
(12)*e3_17 + (4)*e3_18 + (-8)*e3_20 + (4)*e3_21 + (4)*e3_24 +
(-7)*e3_25 + (-7)*e3_28 + (-4)*e3_30 + (-15)*e3_31 + (7)*e3_39 +
(3)*e3_40 + (-8)*e3_41 + e3_46 + (4)*e3_48 + (-8)*e3_50 + (9)*e3_55 +
e3_56 + e3_58 + (-3)*e3_59 + (-8)*e3_63 + (-8)*e3_65 + (-8)*e3_66 +
(-8)*e3_69 + (-4)*e3_70 + (8)*e3_73 + (-10)*e3_76 + (8)*e3_82 +
(8)*e3_85 + (-13)*e3_91 + e3_92 + (7)*e3_93 + e3_94 + e3_97 +
e3_99 + e3_102 + (-1)*e3_103 + (15)*e3_104 + (-1)*e3_106 +
(4)*e3_107 + (14)*e3_110 + (8)*e3_111 + (8)*e3_112 + e3_114 +
e3_115 + (-1)*e3_117 + (-1)*e3_118 + (-4)*e3_119 + (8)*e3_120 +
e3_123 + (2)*e3_125 + (-7)*e3_128 + (-1)*e3_129 + (-7)*e3_130 +
(-7)*e3_131 + (-1)*e3_132 + (2)*e3_133 + (-2)*e3_134 + (3)*e3_136 +
(-1)*e3_138 + (-1)*e3_139 + e3_142 + e3_143 + e3_144 +
(-5)*e3_145 + e3_147 + (6)*e3_150 + (2)*e3_153 + (-6)*e3_156 +
(-1)*e3_157 + (-1)*e3_161 + (-6)*e3_163 + (-8)*e3_164 +
(6)*e3_167 + e3_168 + (-5)*e3_170 + (-13)*e3_171 + (-17)*e3_172 +
```

```
(-17)*e3_174 + (-5)*e3_175 + (-5)*e3_178 + (-5)*e3_180 + e3_181 +
(-3)*e3_185 + e3_186 + (-5)*e3_188 + e3_189 + (-2)*e3_191 +
(5)*e3_195 + (5)*e3_198 + (-2)*e3_199 + (-2)*e3_200 + (10)*e3_202 +
(-1)*e3_204 + (12)*e3_209 + (12)*e3_212 + (-2)*e3_220 + e3_235 +
(-7)*e3_241 + (-9)*e3_243 + e3_244 + e3_245 + (7)*e3_246 +
(4)*e3_247 + (4)*e3_248 + (-1)*e3_254 + e3_255 + (-7)*e3_258 +
(-9)*e3_260 + (4)*e3_262 + (-2)*e3_265 + (-4)*e3_271 + (-4)*e3_272 +
(-4)*e3_273 + (-4)*e3_276 + (-4)*e3_277 + (-4)*e3_278 + (-4)*e3_280 +
(-6)*e3_282 + (-2)*e3_285 + (-3)*e3_293 + (-2)*e3_294 + (-1)*e3_298 +
(-3)*e3_300 + (-2)*e3_301 + (-1)*e3_307 + (-2)*e3_308 + (-1)*e3_310 +
(-1)*e3_313 + (3)*e3_315 + e3_317 + (5)*e3_318 + e3_320 + (3)*e3_321 +
(2)*e3_322 + (5)*e3_323 + (4)*e3_324 + (2)*e3_326 + (-1)*e3_327 +
e3_328 + (-1)*e3_332 + (-2)*e3_335 + (10)*e3_337 + e3_339 + e3_341 +
(10)*e3_344 + (-1)*e3_345 + (10)*e3_347 + (7)*e3_348 + (7)*e3_350 +
(5)*e3_353 + (3)*e3_354 + (3)*e3_355 + (2)*e3_358 + (-2)*e3_359 +
(-2)*e3_368 + (4)*e3_370 + (-3)*e3_371 + (10)*e3_372 + (-1)*e3_373 +
(-2)*e3_374 + (4)*e3_376 + (-3)*e3_377 + (-1)*e3_378 + (-1)*e3_379 +
(4)*e3_383 + (3)*e3_385 + e3_387 + (-2)*e3_392 + (-2)*e3_397 + e3_403 +
(-4)*e3_405 + (-1)*e3_418 + (-2)*e3_421 + (-2)*e3_422 + (-2)*e3_425] +
[1/6]*[(-1)*c3_68 + (-1)*c3_70 + c3_84 + c3_85 + (-1)*c3_114 + c3_133 +
(-3)*c3_145 + c3_152 + (-1)*c3_153 + (-1)*c3_165 + (-2)*c3_175 +
(-2)*c3_177 + (-5)*c3_179 + (-3)*c3_181 + (3)*c3_182 + (5)*c3_191 +
(-5)*c3_197 + (5)*c3_199 + c3_203 + (5)*c3_204 + (-5)*c3_208 + (-5)*c3_218 +
(-1)*c3_220 + (3)*c3_221 + (-7)*c3_222 + (3)*c3_223 + (2)*c3_225 +
(-4)*c3_226 + (-3)*c3_227 + (3)*c3_228 + (3)*c3_229 + (5)*c3_230 +
(2)*c3_233 + (2)*c3_234 + c3_242 + (-1)*c3_243 + (-2)*c3_245 +
(-2)*c3_246 + c3_247 + (-2)*c3_248 + (-3)*c3_250 + (-1)*c3_251 +
(-1)*c3_252 + (2)*c3_253 + (-1)*c3_254 + (5)*c3_258 + c3_259 +
(-3)*c3_260 + c3_262 + (-1)*c3_264 + (-3)*c3_265 + c3_273 +
(-1)*c3_280 + (-2)*c3_284]
```

```
sage: from Sullivan_MM import *
sage: K = simplicial_complexes.K3Surface()
sage: W = APLK(K)
sage: M = W.minimal_model(i=3)
sage: A = M.domain()
sage: A.basis(3)[10]
y3_10
sage: M.phi(_).to_AplK()
{(1, 2, 8, 10): y1*y2*y3,
 (1, 2, 8, 12): y1*y2*y3,
 (1, 3, 4, 6): y1*y2*y3,
 (1, 3, 5, 6): -y1*y2*y3,
 (1, 3, 5, 9): -y1*y2*y3,
 (1, 3, 6, 10): -y1*y2*y3,
 (1, 3, 6, 11): -y1*y2*y3,
 (1, 3, 7, 9): 2*y1*y2*y3,
 (1, 3, 8, 9): 2*y1*y2*y3,
```

```
(1, 3, 8, 12): 3*y1*y2*y3,
(1, 3, 9, 12): y1*y2*y3,
(1, 5, 8, 10): y1*y2*y3,
(2, 3, 4, 9): -y1*y2*y3,
(2, 3, 7, 12): 2*y1*y2*y3,
(2, 3, 8, 12): 2*y1*y2*y3,
(2, 3, 9, 13): y1*y2*y3,
(2, 3, 10, 12): -y1*y2*y3,
(2, 3, 10, 13): -2*y1*y2*y3,
(2, 3, 10, 14): -3*y1*y2*y3,
(2, 3, 12, 14): -2*y1*y2*y3,
(2, 3, 13, 14): -y1*y2*y3,
(2, 4, 5, 9): -2*y1*y2*y3,
(2, 4, 5, 12): -2*y1*y2*y3,
(2, 6, 9, 11): y1*y2*y3,
(2, 10, 13, 15): -y1*y2*y3,
(3, 4, 5, 8): y1*y2*y3,
(3, 4, 5, 9): -3*y1*y2*y3,
(3, 4, 5, 11): -3*y1*y2*y3,
(3, 4, 6, 14): -y1*y2*y3,
(3, 4, 8, 14): -y1*y2*y3,
(3, 5, 6, 11): 2*y1*y2*y3,
(3, 6, 7, 10): -2*y1*y2*y3,
(3, 6, 8, 10): -3*y1*y2*y3,
(3, 6, 8, 11): -4*y1*y2*y3,
(3, 7, 9, 10): -2*y1*y2*y3,
(3, 9, 10, 12): -y1*y2*y3,
(3, 9, 12, 15): y1*y2*y3,
(4, 5, 6, 7): y1*y2*y3,
(4, 5, 6, 12): -2*y1*y2*y3,
(4, 5, 7, 9): -3*y1*y2*y3,
(4, 5, 8, 9): -3*y1*y2*y3,
(4, 5, 8, 11): -3*y1*y2*y3,
(5, 6, 7, 12): -3*y1*y2*y3,
(5, 6, 8, 11): -2*y1*y2*y3,
(5, 7, 9, 12): 3*y1*y2*y3,
(5, 8, 9, 10): -2*y1*y2*y3,
(5, 8, 9, 14): -2*y1*y2*y3,
(5, 8, 9, 15): y1*y2*y3,
(5, 8, 10, 12): -y1*y2*y3,
(5, 8, 11, 12): -y1*y2*y3,
(5, 8, 11, 15): -5*y1*y2*y3,
(5, 8, 12, 13): y1*y2*y3,
(5, 9, 10, 13): -2*y1*y2*y3,
(6, 7, 9, 10): y1*y2*y3,
(6, 7, 9, 11): -y1*y2*y3,
(6, 7, 10, 12): 2*y1*y2*y3,
```

```
(6, 7, 10, 14): -y1*y2*y3,
(6, 7, 11, 12): 3*y1*y2*y3,
(6, 7, 11, 14): y1*y2*y3,
(6, 8, 10, 11): -y1*y2*y3,
(6, 8, 10, 15): -3*y1*y2*y3,
(6, 9, 11, 12): -y1*y2*y3,
(6, 10, 13, 15): -y1*y2*y3,
(6, 11, 12, 14): -y1*y2*y3,
(7, 9, 10, 12): y1*y2*y3,
(7, 9, 10, 13): -y1*y2*y3,
(8, 10, 11, 15): 2*y1*y2*y3,
(8, 11, 12, 14): -2*y1*y2*y3,
(8, 11, 12, 15): -2*y1*y2*y3,
(1, 2, 3, 8, 12): y1*y3*y4 + 3*y2*y3*y4,
(1, 2, 5, 8, 10): y1*y3*y4 + y2*y3*y4,
(1, 2, 8, 10, 12): y1*y2*y3 + y1*y2*y4,
(1, 3, 4, 6, 10): y1*y2*y3 - y1*y3*y4,
(1, 3, 4, 6, 14): y1*y2*y3,
(1, 3, 5, 6, 9): -y1*y2*y3 - y1*y2*y4,
(1, 3, 5, 6, 11): -y1*y2*y3 - y1*y3*y4,
(1, 3, 5, 9, 12): -y1*y2*y3 + y1*y3*y4,
(1, 3, 6, 10, 11): -y1*y2*y3 - y1*y2*y4,
(1, 3, 7, 8, 9): 2*y1*y2*y4 + 2*y1*y3*y4,
(1, 3, 7, 9, 10): 2*y1*y2*y3,
(1, 3, 8, 9, 12): 2*y1*y2*y3 + 3*y1*y2*y4 + y1*y3*y4,
(1, 5, 8, 10, 12): y1*y2*y3,
(2, 3, 4, 5, 9): -y1*y2*y4 - 2*y2*y3*y4,
(2, 3, 4, 9, 13): -y1*y2*y3 + y1*y3*y4,
(2, 3, 7, 8, 12): 2*y1*y2*y4 + 2*y1*y3*y4,
(2, 3, 7, 12, 14): 2*y1*y2*y3 - 2*y1*y3*y4,
(2, 3, 9, 13, 14): y1*y2*y3 - y1*y3*y4,
(2, 3, 10, 12, 14): -y1*y2*y3 - 3*y1*y2*y4 - 2*y1*y3*y4,
(2, 3, 10, 12, 15): (t4 - 1)*y1*y2*y3 - t3*y1*y2*y4 +
                    t2*y1*y3*y4 - t1*y2*y3*y4,
(2, 3, 10, 13, 14): -2*y1*y2*y3 - 3*y1*y2*y4 - y1*y3*y4,
(2, 3, 10, 13, 15): (t4 - 2)*y1*y2*y3 - t3*y1*y2*y4 +
                    t2*y1*y3*y4 + (-t1 - 1)*y2*y3*y4,
(2, 4, 5, 6, 12): -2*y1*y2*y4,
(2, 4, 5, 9, 12): -2*y1*y2*y3 - 2*y1*y2*y4,
(2, 6, 7, 9, 11): y1*y3*y4,
(2, 6, 9, 11, 12): y1*y2*y3,
(2, 10, 13, 15, 16): (t4 - 1)*y1*y2*y3 - t3*y1*y2*y4 +
                     t2*y1*y3*y4 - t1*y2*y3*y4,
(3, 4, 5, 8, 11): (-t4 + 1)*y1*y2*y3 + (t3 - 3)*y1*y2*y4 -
                  t2*y1*y3*y4 + t1*y2*y3*y4,
(3, 4, 5, 8, 15): (-t4 + 1)*y1*y2*y3 + t3*y1*y2*y4 -
                  t2*y1*y3*y4 + t1*y2*y3*y4,
```

```
(3, 4, 5, 9, 11): -3*y1*y2*y3 - 3*y1*y2*y4,
(3, 4, 6, 8, 14): -y1*y2*y4 - y1*y3*y4,
(3, 4, 8, 14, 15): (t4 - 1)*y1*y2*y3 - t3*y1*y2*y4 +
                   t2*y1*y3*y4 - t1*y2*y3*y4,
(3, 5, 6, 8, 11): 2*y1*y2*y4 - 4*y2*y3*y4,
(3, 6, 7, 10, 12): -2*y1*y2*y3,
(3, 6, 7, 10, 16): (2*t4 - 2)*y1*y2*y3 - 2*t3*y1*y2*y4 +
                   2*t2*y1*y3*y4 - 2*t1*y2*y3*y4,
(3, 6, 8, 10, 11): -3*y1*y2*y3 - 4*y1*y2*y4,
(3, 6, 8, 10, 16): (3*t4 - 3)*y1*y2*y3 - 3*t3*y1*y2*y4 +
                   3*t2*y1*y3*y4 - 3*t1*y2*y3*y4,
(3, 7, 9, 10, 12): -2*y1*y2*y3 - y2*y3*y4,
(3, 9, 10, 12, 15): -y1*y2*y3 + y1*y3*y4,
(3, 9, 12, 15, 16): (-t4 + 1)*y1*y2*y3 + t3*y1*y2*y4 -
                    t2*y1*y3*y4 + t1*y2*y3*y4,
(4, 5, 6, 7, 12): y1*y2*y3 - 2*y1*y2*y4,
(4, 5, 6, 7, 16): (-t4 + 1)*y1*y2*y3 + t3*y1*y2*y4 -
                  t2*y1*y3*y4 + t1*y2*y3*y4,
(4, 5, 7, 9, 12): -3*y1*y2*y3,
(4, 5, 7, 9, 15): (3*t4 - 3)*y1*y2*y3 - 3*t3*y1*y2*y4 +
                  3*t2*y1*y3*y4 - 3*t1*y2*y3*y4,
(4, 5, 8, 9, 11): -3*y1*y2*y3 - 3*y1*y2*y4,
(4, 5, 8, 9, 15): (2*t4 - 3)*y1*y2*y3 - 2*t3*y1*y2*y4 +
                  2*t2*y1*y3*y4 - 2*t1*y2*y3*y4,
(5, 6, 7, 12, 16): (3*t4 - 3)*y1*y2*y3 - 3*t3*y1*y2*y4 +
                   3*t2*y1*y3*y4 - 3*t1*y2*y3*y4,
(5, 6, 8, 11, 15): (-3*t4 - 2)*y1*y2*y3 + 3*t3*y1*y2*y4 -
                   3*t2*y1*y3*y4 + (3*t1 - 5)*y2*y3*y4,
(5, 7, 9, 12, 16): (-3*t4 + 3)*y1*y2*y3 + 3*t3*y1*y2*y4 -
                   3*t2*y1*y3*y4 + 3*t1*y2*y3*y4,
(5, 8, 9, 10, 13): -2*y1*y2*y3 - 2*y2*y3*y4,
(5, 8, 9, 10, 14): -2*y1*y2*y3 - 2*y1*y2*y4,
(5, 8, 9, 13, 15): t4*y1*y2*y3 + (-t3 + 1)*y1*y2*y4 +
                   t2*y1*y3*y4 - t1*y2*y3*y4,
(5, 8, 9, 14, 16): (2*t4 - 2)*y1*y2*y3 - 2*t3*y1*y2*y4 +
                   2*t2*y1*y3*y4 - 2*t1*y2*y3*y4,
(5, 8, 10, 12, 13): -y1*y2*y3 + y1*y3*y4,
(5, 8, 11, 12, 15): (-2*t4 - 1)*y1*y2*y3 +
                    (2*t3 - 5)*y1*y2*y4 - 2*t2*y1*y3*y4 +
                    2*t1*y2*y3*y4,
(5, 8, 11, 12, 16): (t4 - 1)*y1*y2*y3 - t3*y1*y2*y4 +
                    t2*y1*y3*y4 - t1*y2*y3*y4,
(5, 8, 12, 13, 15): (-t4 + 1)*y1*y2*y3 + t3*y1*y2*y4 -
                    t2*y1*y3*y4 + t1*y2*y3*y4,
(5, 9, 10, 13, 15): (2*t4 - 2)*y1*y2*y3 - 2*t3*y1*y2*y4 +
                    2*t2*y1*y3*y4 - 2*t1*y2*y3*y4,
(6, 7, 9, 10, 13): y1*y2*y3,
```

```
(6, 7, 9, 10, 14): y1*y2*y3 - y1*y3*y4,
(6, 7, 9, 11, 14): -y1*y2*y3 + y1*y3*y4,
(6, 7, 10, 12, 15): (-2*t4 + 2)*y1*y2*y3 + 2*t3*y1*y2*y4 -
                    2*t2*y1*y3*y4 + 2*t1*y2*y3*y4,
(6, 7, 10, 13, 15): -t4*y1*y2*y3 + t3*y1*y2*y4 - t2*y1*y3*y4 +
                    (t1 - 1)*y2*y3*y4,
(6, 7, 10, 14, 16): (t4 - 1)*y1*y2*y3 - t3*y1*y2*y4 +
                    t2*y1*y3*y4 - t1*y2*y3*y4,
(6, 7, 11, 12, 15): (-3*t4 + 3)*y1*y2*y3 + 3*t3*y1*y2*y4 -
                    3*t2*y1*y3*y4 + 3*t1*y2*y3*y4,
(6, 7, 11, 12, 16): (-3*t4 + 3)*y1*y2*y3 + 3*t3*y1*y2*y4 -
                    3*t2*y1*y3*y4 + 3*t1*y2*y3*y4,
(6, 7, 11, 14, 16): (-t4 + 1)*y1*y2*y3 + t3*y1*y2*y4 -
                    t2*y1*y3*y4 + t1*y2*y3*y4,
(6, 8, 10, 11, 15): (-4*t4 - 1)*y1*y2*y3 + (4*t3 - 3)*y1*y2*y4 -
                    4*t2*y1*y3*y4 + 4*t1*y2*y3*y4,
(6, 8, 10, 15, 16): (3*t4 - 3)*y1*y2*y3 - 3*t3*y1*y2*y4 +
                    3*t2*y1*y3*y4 - 3*t1*y2*y3*y4,
(6, 9, 11, 12, 14): -y1*y2*y3 - y2*y3*y4,
(6, 10, 13, 15, 16): (t4 - 1)*y1*y2*y3 - t3*y1*y2*y4 +
                     t2*y1*y3*y4 - t1*y2*y3*y4,
(6, 11, 12, 14, 16): (t4 - 1)*y1*y2*y3 - t3*y1*y2*y4 +
                     t2*y1*y3*y4 - t1*y2*y3*y4,
(7, 9, 10, 12, 15): (-t4 + 1)*y1*y2*y3 + t3*y1*y2*y4 -
                    t2*y1*y3*y4 + t1*y2*y3*y4,
(7, 9, 10, 13, 15): (t4 - 1)*y1*y2*y3 - t3*y1*y2*y4 +
                    t2*y1*y3*y4 - t1*y2*y3*y4,
(8, 10, 11, 15, 16): (-2*t4 + 2)*y1*y2*y3 + 2*t3*y1*y2*y4 -
                     2*t2*y1*y3*y4 + 2*t1*y2*y3*y4,
(8, 11, 12, 14, 15): -2*y1*y2*y3 - 2*y1*y2*y4,
(8, 11, 12, 14, 16): (2*t4 - 2)*y1*y2*y3 - 2*t3*y1*y2*y4 +
                     2*t2*y1*y3*y4 - 2*t1*y2*y3*y4}
```

BIBLIOGRAPHY

[1] Benjamin I Albert. The dupont homotopy formula and stellar subdivision. *arXiv preprint arXiv:1902.00627*, 2019.

[2] Carlos Alquézar Baeta, Miguel Ángel Marco Buzunáriz, and Jorge Martín Morales. Cálculo efectivo del modelo de sullivan de un espacio topológico y su tipo de homotopía racional. Master's thesis, Universidad de Zaragoza, 2024.

[3] Carlos Alfonso Alquézar-Baeta, Miguel A Marco-Buzunariz, and Jorge Martín-Morales. The minimal model of a simplicial complex: algorithm and implementation. In *EACA 2022: XVII Encuentro de Álgebra Computacional y Aplicaciones*, pages 39–42. Servei de Comunicació i Publicacions, 2023.

[4] Douglas Arnold, Richard Falk, and Ragnar Winther. Finite element exterior calculus: from hodge theory to numerical stability. *Bulletin of the American mathematical society*, 47(2):281–354, 2010.

[5] Douglas N Arnold. *Finite element exterior calculus*. SIAM, 2018.

[6] Douglas N Arnold, Richard S Falk, and Ragnar Winther. Finite element exterior calculus, homological techniques, and applications. *Acta numerica*, 15:1–155, 2006.

[7] E. Artal, A. Larraya Sancho, and M. A. Marco Buzunariz. Retraction of the complement of smooth projective curves to a 2-dimensional δ-complex, 2026.

[8] Steve Awodey. *Category theory*, volume 52. OUP Oxford, 2010.

[9] Aldridge Knight Bousfield and Victor KAM Gugenheim. *On PL DeRham theory and rational homotopy type*, volume 179. American Mathematical Soc., 1976.

[10] Mario Casella and Wolfgang Kühnel. A triangulated K3 surface with the minimum number of vertices. *Topology*, 40(4):753–772, 2001.

[11] Frédéric Chapoton. The simplicial complex of the K3 surface (SageMath repository), 2011. `https://doc.sagemath.org/html/en/reference/topology/sage/topology/simplicial_complex_examples.html#sage.topology.simplicial_complex_examples.K3Surface`.

[12] Aohua Cheng, Yunhui Xu, Pei Sun, and Yang Tian. A simplex path integral and a simplex renormalization group for high-order interactions. *Reports on Progress in Physics*, 87(8):087601, 2024.

[13] Xue Zhi Cheng and Ezra Getzler. Transferring homotopy commutative algebraic structures. *Journal of Pure and Applied Algebra*, 212(11):2535–2542, 2008.

[14] DanielRobertNicoud. Danielrobertnicoud/dupont-contraction: v2.1.0, December 2020.

[15] Georges De Rham. *Sur l'analysis situs des variétés à n dimensions*. 1931.

[16] Pierre Deligne, Phillip Griffiths, John Morgan, and Dennis Sullivan. Real homotopy theory of kähler manifolds. *Inventiones mathematicae*, 29(3):245–274, 1975.

[17] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 10.2)*, 2023. `https://www.sagemath.org`.

[18] Jozef Dodziuk. Finite-difference approach to the hodge theory of harmonic forms. *American Journal of Mathematics*, 98(1):79–104, 1976.

[19] Johan L Dupont. Simplicial de rham cohomology and characteristic classes of flat bundles. *Topology*, 15(3):233–245, 1976.

[20] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[21] Yves Félix, Stephen Halperin, and Jean-Claude Thomas. *Rational Homotopy Theory*. Number 205 in Graduate Texts in Mathematics. Springer, New York, 2001.

[22] Yves Félix and Steve Halperin. Rational homotopy theory via Sullivan models: A survey. *Notices of the International Consortium of Chinese Mathematicians*, 5(2):14–36, 2017.

[23] Yves Félix and Steve Halperin. Rational homotopy via Sullivan models and enriched Lie algebras. *EMS Surveys in Mathematical Sciences*, 10(1):101–122, 2023.

[24] Yves Félix, Steve Halperin, and Jean-Claude Thomas. *Rational homotopy theory II*. World Scientific, 2015.

[25] Yves Felix, Barry Jessup, and Paul-Eugène Parent. The combinatorial model for the sullivan functor on simplicial sets. *Journal of Pure and Applied Algebra*, 213(2):231–240, 2009.

[26] Yves Félix, John Oprea, and Daniel Tanré. *Algebraic models in geometry*. OUP Oxford, 2008.

[27] Greg Friedman. Survey article: an elementary illustrated introduction to simplicial sets. *The Rocky Mountain Journal of Mathematics*, pages 353–423, 2012.

[28] Joel Friedman. Computing betti numbers via combinatorial laplacians. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, pages 386–391, 1996.

[29] Rudolf Fritsch and Renzo A Piccinini. *Cellular structures in topology*. Number 19. Cambridge University Press, 1990.

[30] Antonio Garvin, Rocío González-Díaz, Miguel Ángel Marco, and Belén Medrano. Making Sullivan Algebras Minimal Through Chain Contractions. *Mediterranean Journal of Mathematics*, 18(2):43, January 2021.

[31] Ezra Getzler. Lie theory for nilpotent l_∞ -algebras. *Annals of Mathematics*, 170(1):271–301, 2009.

[32] Timothy E Goldberg. *Combinatorial Laplacians of simplicial complexes*. PhD thesis, Bard College, 2002.

[33] Rocío González-Díaz, María José Jiménez, Belén Medrano, and Pedro Real. Chain homotopies for object topological representations. *Discrete Applied Mathematics*, 157(3):490–499, 2009.

[34] Rocío González-Díaz, María José Jiménez, Belén Medrano, and Pedro Real. A tool for integer homology computation: λ-at-model. *Image and Vision Computing*, 27(7):837–845, 2009.

[35] Rocio Gonzalez-Diaz and Pedro Real. Computation of cohomology operations of finite simplicial complexes. 2003.

[36] Rocío González-Díaz and Pedro Real. On the cohomology of 3d digital images. *Discrete Applied Mathematics*, 147(2-3):245–263, 2005.

[37] Phillip Griffiths and John W. Morgan. *Rational Homotopy Theory and Differential Forms*, volume 16. Springer, 1981.

[38] Richard Gustavson. *Some New Results on the Combinatorial Laplacian*. PhD thesis, CORNELL UNIVERSITY, 2011.

[39] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, 2002.

[40] Kathryn Hess, Samuel Lavenir, and Kelly Maggs. Cell decompositions of persistent minimal models. *arXiv preprint arXiv:2312.08326*, 2023.

[41] Oliver Knill. The energy of a simplicial complex. *Linear Algebra and its Applications*, 600:96–129, 2020.

[42] Jean-Louis Loday. *Cyclic homology*, volume 301 of *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1992. Appendix E by María O. Ronco.

[43] Jonni Lohi and Lauri Kettunen. Whitney forms and their extensions. *Journal of Computational and Applied Mathematics*, 393:113520, 2021.

[44] Luigi Lunardon. Some remarks on dupont contraction. *arXiv preprint arXiv:1807.02517*, 2018.

[45] Anca Daniela Măcinic. Cohomology rings and formality properties of nilpotent groups. *Journal of Pure and Applied Algebra*, 214(10):1818–1826, 2010.

[46] Michael A Mandell. Cochains and homotopy type. *Publications Mathématiques de l'Institut des Hautes Études Scientifiques*, 103(1):213–246, 2006.

[47] Víctor Manero and Miguel Marco Buzunáriz. Effective computation of degree bounded minimal models of GCDAs. *Journal of Software for Algebra and Geometry*, 10(1):25–39, May 2020.

[48] Elisabeth Louise Mansfield and ED Fackerell. *Differential gröbner bases*, volume 92. University of Sidney, 1992.

[49] Anibal M Medina-Mardones. The diagonal of cellular spaces and effective algebro-homotopical constructions. *EMS Surveys in Mathematical Sciences*, 10(2):223–241, 2023.

[50] Keirn Yves Ross Munro. *Simplicial Approximation of the Hodge Laplacian Using Cauchy Sequences of Hilbert Complexes*. PhD thesis, University of Toronto (Canada), 2024.

[51] John H. Palmieri. *Algebraic Topological Models for a cell complex.*, 2015. `https://doc.sagemath.org/pdf/en/reference/homology/homology.pdf`.

[52] John H. Palmieri. *Chain Homotopies and Chain Contractions.*, 2015. `https://doc.sagemath.org/pdf/en/reference/homology/homology.pdf`.

[53] Pawel Pilarczyk and Pedro Real. Computation of cubical homology, cohomology, and (co) homological operations via chain contraction. *Advances in Computational Mathematics*, 41(1):253–275, 2015.

[54] Mária Šimková. *Algorithmic homotopy theory*. PhD thesis, Masaryk University, 2024.

[55] Mária Šimková. Rational homotopy equivalence. *arXiv preprint arXiv:2512.21182*, 2025.

[56] Jonathan Spreer and Wolfgang Kühnel. Combinatorial properties of the $K3$ surface: Simplicial blowups and slicings. *Experimental Mathematics*, 20(2):201–216, 2011.

[57] Dennis Sullivan. Infinitesimal computations in topology. *Publications Mathématiques de l'IHÉS*, 47:269–331, 1977.

[58] Dennis Sullivan. Appendix to infinity structure of Poincaré duality spaces [**60**]. *Algebr. Geom. Topol*, 7:233–260, 2007.

[59] Svjetlana Terzić. On rational homotopy of four-manifolds. In *Contemporary Geometry and Related Topics*, pages 375–388. World Scientific, 2004.

[60] T Thomas. Infinity structure of poincaré duality spaces. *Appendix A by Dennis Sullivan, Algebr. Geom. Topol.*, 7:233, 2007.

[61] Joaquín J Torres and Ginestra Bianconi. Simplicial complexes: higher-order spectral dimension and dynamics. *Journal of Physics: Complexity*, 1(1):015002, may 2020.

[62] Hassler Whitney. Geometric integration theory. 1957.

[63] Scott O. Wilson. Geometric structures on the cochains of a manifold, 2022.

[64] Yan-Lin Yu. Combinatorial Gauss-Bonnet-Chern formula. *Topology*, 22(2):153–163, 1983.

[65] Ling Zhou. Persistent Sullivan minimal models of metric spaces. *arXiv preprint arXiv:2310.06263*, 2023.