

Computing the Saneblidze-Umble Diagonal on
Permutahedra

A Senior Thesis Submitted to the Department of
Mathematics In Partial Fulfillment of the
Requirements for the Departmental Honors
Baccalaureate

By Stephen J. Weaver

Millersville, Pennsylvania

May 2005

This Senior Thesis was completed in the Department of Mathematics, defended before and approved by the following members of the Thesis committee

Ronald Umble, Ph.D (Thesis Advisor)
Professor of Mathematics
Millersville University

David Hutchens, Ph.D
Associate Professor of Computer Science
Millersville University

Bruce Ikenaga, Ph.D
Assistant Professor of Mathematics
Millersville University

Abstract

This thesis establishes algorithms with C++ implementations for computing the Sanedidze-Umble diagonal approximation Δ_P on a permutahedron P_n , for computing the boundary of its faces and for solving large sparse systems of mod two linear equations. The latter is applied to find a chain homotopy from $(\Delta_P \otimes 1)\Delta_P$ to $(1 \otimes \Delta_P)\Delta_P$ in low dimensions giving an explicit formula for the homotopy coassociativity of Δ_P when $n \leq 4$.

Computing the Saneblidze-Umble Diagonal on Permutahedra

Contents

1 Permutahedra

Permutahedra are geometric shapes that are built from sets. Exactly one permutahedron corresponds to a set with a given number of elements. Each face of the permutahedron corresponds to a permutation in the order of subsets of the original set, where the subsets partition the original set. These faces fit together to form a permutahedron. For convenience, the sets in a partition are separated by a bar (|). For example, the partition $\{1, 2\} \cup \{3\}$ of $\{1, 2, 3\}$ is written as $12|3$. Since permutations are ordered but sets are not, $12|3$ is different from $3|12$, but $12|3$ is the same as $21|3$. Each permutation corresponds to some face of a permutahedron.

The set $\underline{n} = \{1, 2, \dots, n\}$ represents the $(n - 1)$ - dimensional permutahedron, P_n , whose boundary consists of $n - 2$ dimensional faces. Each $n - 2$ dimensional face $a \subset P_n$ is a partition of form $A|B$, where the bar is thought of geometrically as the Cartesian product. Thus a is a Cartesian product of lower dimensional permutahedra. For example, the partition $12|34$ is a face of P_4 and is geometrically a Cartesian product of P_2 's – a one-dimensional line segment (12) crossed with another line segment (34). This creates a two-dimensional rectangle.

Consider a face $u \subset P_n$ and its corresponding partition $u = U_1 | \dots | U_k$ of \underline{n} . The boundary of u , denoted $\partial(u)$, consists of all partitions that can be obtained from u by inserting a single bar. Hence $\partial(12)$ (boundary of a line segment) consists of vertices $1|2$ and $2|1$. A permutahedron is always “closed,” which means that the boundary of its boundary is empty. Let F_i be the set of i dimensional faces, the permutahedron $P_n = \bigcup_{0 \leq i \leq n-1} F_i$, where the sets F_i can be built inductively as follows:

$$\begin{aligned} \text{Set } F_{n-1} &= 123 \dots n. \text{ If } F_k \text{ has been constructed,} \\ &\text{where } 1 \leq k \leq n - 1, \\ \text{then } F_{k-1} &= \bigcup_{a \in F_k} \partial(a). \end{aligned}$$

These pieces fit together to make a geometric representation of P_n using the following adjacency rule: If the boundaries of two k -dimensional faces have a non-empty intersection, they intersect along a common subface and are considered to be adjacent through that

subface. For example, $1|23|4 = \partial(1|234) \cap \partial(123|4)$, so $1|234$ and $123|4$ (both hexagons) are adjacent through their common side $1|23|4$.

Since the only proper subset of $P_1 = \{1\}$ is the empty set, the geometric representation of P_1 is a point. The set $\underline{2} = \{1, 2\}$ has two non-empty proper subsets $\{1\}$ and $\{2\}$, whose permutations are $1|2$ and $2|1$. Thus the geometric representation of P_2 is an interval with endpoints $1|2$ and $2|1$. Pictures of two and three dimensional permutahedra appear below.

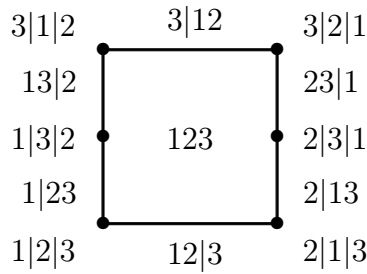


Figure 1: P_3 as a subdivision of $P_2 \times I$.

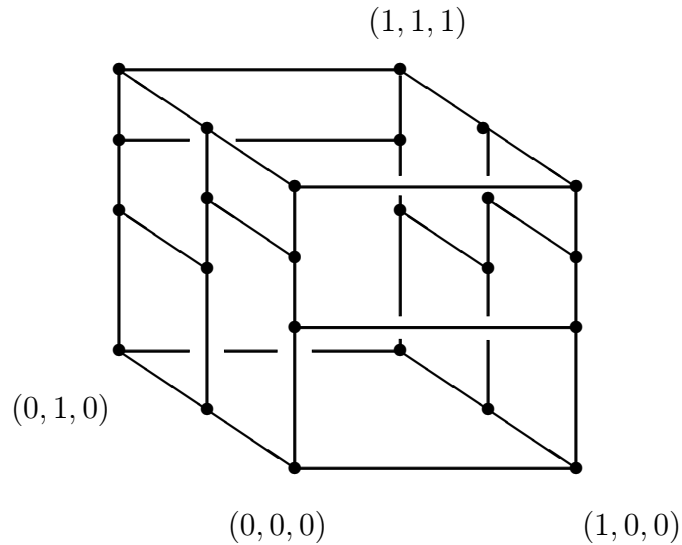


Figure 2a: P_4 as a subdivision of $P_3 \times I$.

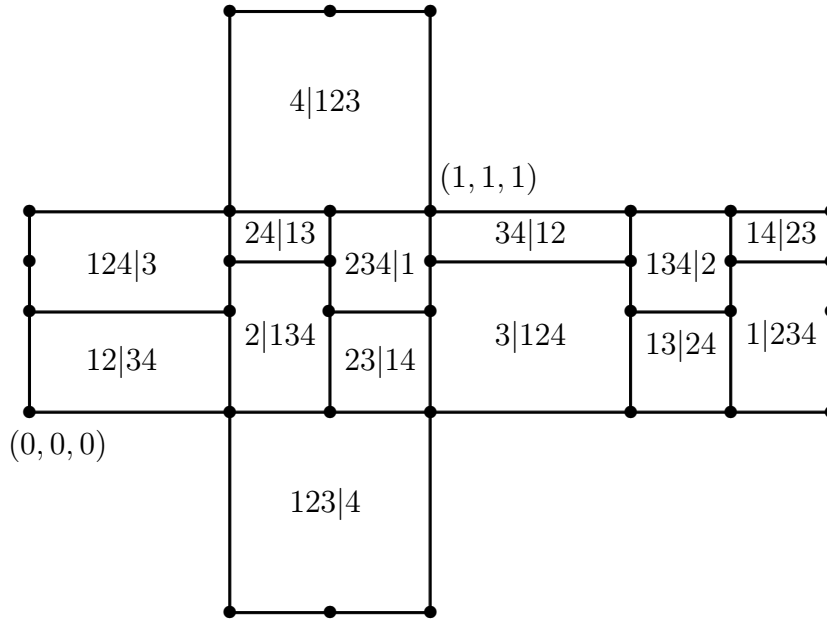


Figure 2b: The 2-faces of P_4 .

2 S-U Diagonal

The diagonal of a set S is defined to be $\Delta(S) = \{(x, x) | x \in S\} \subset S \times S$. When S is a permutahedron, $\Delta(P_n)$ is not a face of $P_n \times P_n$ for $n > 1$. However, $\Delta(P_n)$ can be approximated "up to homotopy" using the faces of $P_n \times P_n$. For example, for P_2

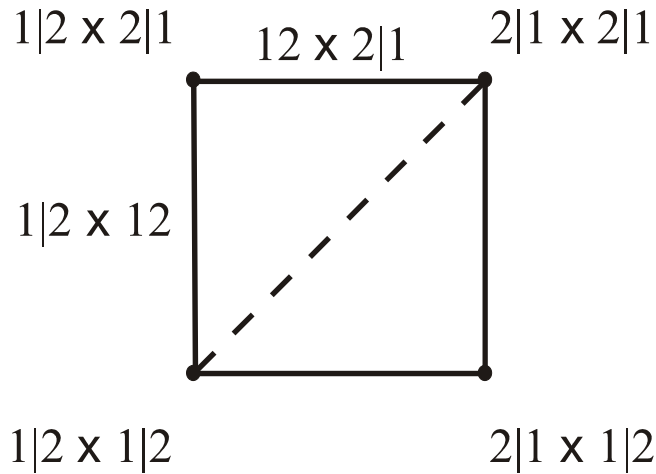


Figure 3: The diagonal on P_2 .

To think of this algebraically, consider the vector space of *cellular chains* of P_n , denoted $C_*(P_n) = \bigoplus_{i \geq 0} C_i(P_n)$, where $C_i(P_n)$ is the \mathbb{Z}_2 -vector space spanned by the i -dimensional faces of P_n . The tensor product $C_*(P_n) \otimes C_*(P_n)$ is the vector space spanned by all symbols $a \otimes b$, where $a, b \in C_*(P_n)$. Note that in general, $a \otimes b \neq b \otimes a$. Cartesian products of permutahedra translate into tensor products of vector spaces in the following way:

$$C_*(P_n \times P_m) = C_*(P_n) \otimes C_*(P_m).$$

When a and b are both faces of P_n , $a \times b$ is a face of $P_n \times P_n$ and $a \otimes b \in C_*(P_n) \otimes C_*(P_n)$. The geometric diagonal $\Delta(P_n)$ is approximated by a "path" of $(n-1)$ -dimensional faces of $P_n \times P_n$ thought of algebraically as a sum of elements $u_i \otimes v_i \in C_*(P_n) \otimes C_*(P_n)$, where $\dim u_i + \dim v_i = n-1$. Thus, a diagonal approximation of $\Delta(P_n)$ appears algebraically as

$$\Delta(\underline{n}) = \sum u_i \otimes v_i$$

for appropriate choices of u_i and v_i . Saneblidze and Umble [?] described a pairing of faces u and v that approximates Δ . This approximation of the diagonal is called the *S-U diagonal* and is denoted Δ_P .

3 Calculating the S-U Diagonal

The S-U diagonal can be calculated using "step matrices."

Definition 1 Let $M = (m_{i,j})$ be a $p \times q$ matrix. M is ordered if

1. $m_{i,j} \in \{0, 1, 2, \dots, p+q-1\}$.
2. Each row and column of M is nonzero.
3. Nonzero entries in M are distinct and increase in each row and column.

Furthermore,

Definition 2 An ordered matrix M is a step matrix if

1. The nonzero entries in each row M appear in consecutive columns.

2. The nonzero entries in each column of M appear in consecutive rows.
3. The sub, main, and super diagonals of M contain a single nonzero entry.

In descriptive terms, the nonzero entries of a step matrix form a staircase from the lower left to the upper right corners.

| | | | | |
|---|---|---|---|---|
| | | | 5 | 6 |
| | | 4 | 7 | |
| | | 8 | | |
| | 1 | 9 | | |
| 2 | 3 | | | |

Figure 4: A typical step matrix.

Each $p \times q$ step matrix such that $p + q - 1 = n$ corresponds to exactly one permutation of \underline{n} , and can be built from \underline{n} as follows:

Algorithm 1 Let $(\rho_1, \rho_2, \dots, \rho_n)$ be a permutation of \underline{n} ; let p be the number of strictly increasing substrings of maximal length.

| |
|---|
| <p>Let $m_{p,1} = \rho_1$.</p> <p>Inductively, for $k = 2, 3, \dots, n$, let (i, j) be the row and column position of ρ_{k-1}</p> <p style="padding-left: 20px;">If $\rho_k > \rho_{k-1}$, let $m_{i,j+1} = \rho_k$;</p> <p style="padding-left: 20px;">otherwise, let $m_{i-1,j} = \rho_k$.</p> |
|---|

A step matrix can also be transformed in two ways. Given a $p \times q$ step matrix M and $x \in \underline{n}$, there is some (i, j) such that $m_{i,j} = x$. Let C'_i be the set of elements in the i th column of M and let R'_j be the set of elements in the j th row; then $\{x\} = C'_i \cap R'_j$. Use the following two operations to transform M :

Operation 1a:

If $i < p$ and $x \in R'_i$,
 and $x > y, \forall y \in R'_{i+1}$,
 and $m_{i+1,j} = 0$,
 then x can be moved to $m_{i+1,j}$ and $m_{i,j}$ replaced with 0.

Operation 1b:

If $j < q$ and $x \in C'_j$,
 and $x > y, \forall y \in C'_{j+1}$,
 and $m_{i,j+1} = 0$,
 then x can be moved to $m_{i,j+1}$ and $m_{i,j}$ replaced with 0.

The transformed matrices are no longer step matrices, but they are still ordered matrices. These transformations can be performed on ordered matrices also, recursively transforming previously transformed matrices. This process will eventually terminate once the elements have moved downward or to the right as far as possible.

The pair $u_i \otimes v_i$ can be read from the transformed matrix M' in this fashion: $u_i = C'_1|C'_2|\dots|C'_q$ and $v_i = R'_p|\dots|R'_1$. The S-U diagonal $\Delta_P = \sum u_i \otimes v_i$, summed over all the step matrices and their transformations. Note that the $u_i \otimes v_i$ pairs are not always distinct. In this thesis we use mod 2 coefficients; thus a pair $u_i \otimes v_i$ that appears an even number of times is dropped. In general, the formula for Δ_P can be written using signed terms which encode the orientation of the faces, but this is not a concern here.

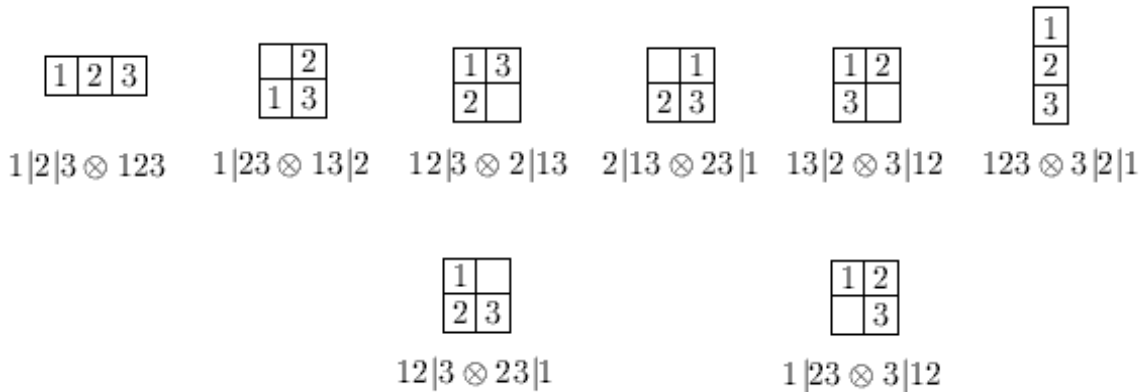


Figure 5: SU-diagonal on P_3 .

Andy Tonks [?] showed that the number of terms in $\Delta_P(\underline{n})$ is $2(n+1)^{n-2}$, $n \geq 2$. Thus the number of terms in Δ_P becomes large quickly and it is necessary to perform these calculations on a computer for $n > 3$.

4 Coassociativity

The S-U diagonal is a linear map $\Delta_P : C_*(P_n) \rightarrow C_*(P_n) \otimes C_*(P_n)$ that preserves the Cartesian product (\otimes) operation, i.e., $\Delta_P(a|b) = \Delta_P(a) \otimes \Delta_P(b)$. Since Δ_P maps in the opposite direction of a normal multiplication ($m : A \otimes A \rightarrow A$), Δ_P is called a *comultiplication*. We are interested in the coassociativity of Δ_P . Coassociativity is the comultiplicative analog of associativity in which $(ab)c = a(bc)$. When a multiplication m is associative, we may multiply the left two factors or the right two factors first, i.e., $m(m(a \otimes b) \otimes c) = m(a \otimes m(b \otimes c))$ or equivalently, $m(m \otimes 1) = m(1 \otimes m)$. Coassociativity goes the other way: $(\Delta \otimes 1)\Delta = (1 \otimes \Delta)\Delta$. Thus after splitting the input, we are free to split either the left or right factor of the output.

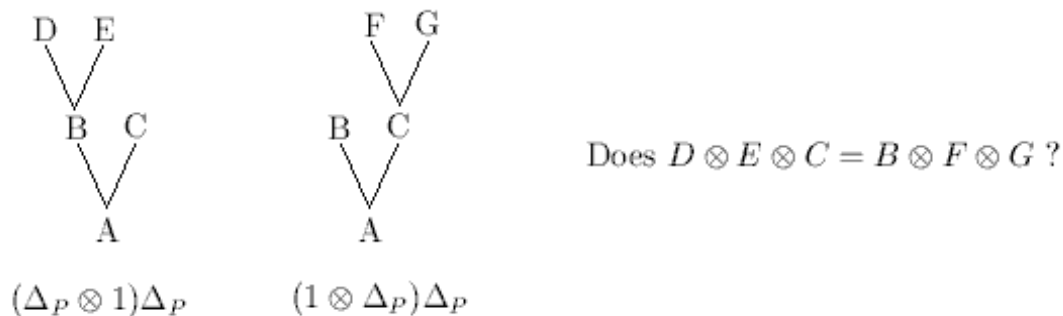


Figure 6: Coassociativity.

Unfortunately the S-U diagonal Δ_P is not coassociative as the following example on P_3 demonstrates:

$$\begin{aligned}
& [(\Delta_P \otimes Id)\Delta_P - (Id \otimes \Delta_P)\Delta_P](123) = \\
& \begin{array}{rcl}
2|1|3 \otimes 2|13 \otimes 23|1 & - & 1|2|3 \otimes 2|13 \otimes 23|1 \\
+1|3|2 \otimes 13|2 \otimes 3|12 & - & 1|2|3 \otimes 13|2 \otimes 3|12 \\
+12|3 \otimes 2|13 \otimes 3|2|1 & - & 12|3 \otimes 2|13 \otimes 2|3|1 \\
+12|3 \otimes 2|1|3 \otimes 23|1 & - & 12|3 \otimes 2|3|1 \otimes 23|1 \\
+1|23 \otimes 13|2 \otimes 3|2|1 & - & 1|23 \otimes 13|2 \otimes 3|1|2 \\
+1|23 \otimes 1|3|2 \otimes 3|12 & - & 1|23 \otimes 3|1|2 \otimes 3|12 \neq 0.
\end{array} \tag{1}
\end{aligned}$$

Our calculations will show that although Δ_P fails to be coassociative, the two coassociations relate in a meaningful way. In order to understand this more, let X be a polyhedron and consider the vector space of cellular chains $C_*(X)$ introduced earlier. The boundary of a face has dimension one less than the dimension of the face, so the boundary map ∂ lowers dimension by 1. With $C_i(X)$ denoting the \mathbb{Z}_2 -vector space spanned by the i -dimensional faces of X , we have

$$\partial = \sum_i \partial_i : C_i(X) \rightarrow C_{i-1}(X)$$

and as mentioned earlier, $\partial \circ \partial = 0$. The pair $(C_*(X), \partial)$ is the *cellular chain complex* on X (see Figure 7).

$$\cdots \xrightarrow{\partial} C_i(X) \xrightarrow{\partial} \cdots \xrightarrow{\partial} C_1(X) \xrightarrow{\partial} C_0(X)$$

Figure 7: The chain complex $C_*(X)$.

Cellular chains on any polyhedron X respect Cartesian products, i.e.,

$$C_* \underbrace{(X \times \cdots \times X)}_{n\text{-factors}} = \underbrace{C_*(X) \otimes \cdots \otimes C_*(X)}_{n\text{-factors}}.$$

The boundary map acts as a derivation (by the Leibniz rule) on a tensor products. So on 3-fold tensor products we have

$$\partial^{\otimes 3} = \partial \otimes 1 \otimes 1 + 1 \otimes \partial \otimes 1 + 1 \otimes 1 \otimes \partial.$$

Proposition 3 $(C_*(X)^{\otimes 3}, \partial^{\otimes 3})$ is a chain complex.

Proof. We must show that $\partial^{\otimes 3} \circ \partial^{\otimes 3} = 0$.

$$\begin{aligned}
\partial^{\otimes 3} \circ \partial^{\otimes 3} &= (\partial \otimes 1 \otimes 1 + 1 \otimes \partial \otimes 1 + 1 \otimes 1 \otimes \partial)(\partial \otimes 1 \otimes 1 + 1 \otimes \partial \otimes 1 + 1 \otimes 1 \otimes \partial) \\
&= (\partial \circ \partial \otimes 1 \otimes 1 + \partial \otimes \partial \otimes 1 + \partial \otimes 1 \otimes \partial) + \\
&(\partial \otimes \partial \otimes 1 + 1 \otimes \partial \circ \partial \otimes 1 + 1 \otimes \partial \otimes \partial) + \\
&(\partial \otimes 1 \otimes \partial + 1 \otimes \partial \otimes \partial + 1 \otimes 1 \otimes \partial \circ \partial) \\
&= (\partial \otimes \partial \otimes 1 + \partial \otimes 1 \otimes \partial) + (\partial \otimes \partial \otimes 1 + 1 \otimes \partial \otimes \partial) + (\partial \otimes 1 \otimes \partial + 1 \otimes \partial \otimes \partial) \\
&= 2(\partial \otimes \partial \otimes 1) + 2(\partial \otimes 1 \otimes \partial) + 2(1 \otimes \partial \otimes \partial) \\
&\equiv 0 \quad \blacksquare
\end{aligned}$$

Definition 4 Let $(C_*(X), \partial_X)$ and $(C_*(Y), \partial_Y)$ be chain complexes. A chain map $f = \sum_i f_i : C_i(X) \rightarrow C_i(Y)$ is a linear map such that

$$f \circ \partial_X = \partial_Y \circ f.$$

Theorem 5 (Saneblidze and Umble) Δ_P is a chain map.

Note that $g = (\Delta_P \otimes 1) \circ \Delta_P$ and $h = (1 \otimes \Delta_P) \circ \Delta_P$ are the two coassociations of Δ_P . Furthermore

$$g, h : C_*(P_n) \rightarrow C_*(P_n)^{\otimes 3}$$

for all n .

Proposition 6 g and h are chain maps.

Proof. We must show that $g \circ \partial = \partial^{\otimes 3} \circ g$; a similar calculation holds for h .

$$\begin{aligned}
g \circ \partial &= (\Delta_P \otimes 1) \Delta_P \partial \\
&= (\Delta_P \otimes 1)(\partial \otimes 1 + 1 \otimes \partial) \Delta_P \\
&= (\Delta_P \partial \otimes 1 + \Delta_P \otimes \partial) \Delta_P \\
&= ((\partial \otimes 1 + 1 \otimes \partial) \Delta_P \otimes 1 + \Delta_P \otimes \partial) \Delta_P \\
&= ((\partial \otimes 1) \Delta_P \otimes 1 + (1 \otimes \partial) \Delta_P \otimes 1 + \Delta_P \otimes \partial) \Delta_P \\
&= [(\partial \otimes 1 \otimes 1)(\Delta_P \otimes 1) + (1 \otimes \partial \otimes 1)(\Delta_P \otimes 1) + (1 \otimes 1 \otimes \partial)(\Delta_P \otimes 1)] \Delta_P \\
&= (\partial \otimes 1 \otimes 1 + 1 \otimes \partial \otimes 1 + 1 \otimes 1 \otimes \partial)(\Delta_P \otimes 1) \Delta_P \\
&= \partial^{\otimes 3} \circ g. \quad \blacksquare
\end{aligned}$$

Definition 7 Let $(C_*(X), \partial_X)$ and $(C_*(Y), \partial_Y)$ be chain complexes. Two chain maps $f, f' : C_*(X) \rightarrow C_*(Y)$ are chain homotopic if and only if there exists a linear map $H = \sum_i H_i : C_i(X) \rightarrow C_{i+1}(Y)$ such that

$$f' - f = \partial_Y \circ H + H \circ \partial_X. \quad (2)$$

In our setting with \mathbb{Z}_2 coefficients this formula (??) becomes

$$f' + f = \partial_Y \circ H + H \circ \partial_X.$$

In particular, the sum $g + h = (\Delta_P \otimes 1)\Delta_P + (1 \otimes \Delta_P)\Delta_P$ measures how far Δ_P deviates from coassociativity. So if there is an H such that

$$g + h = \partial^{\otimes 3} \circ H + H \circ \partial \quad (3)$$

then g and h are chain homotopic and we say that Δ_P is *homotopy coassociative*. The diagram of chain complexes and chain maps with $f = g + h$ looks like this:

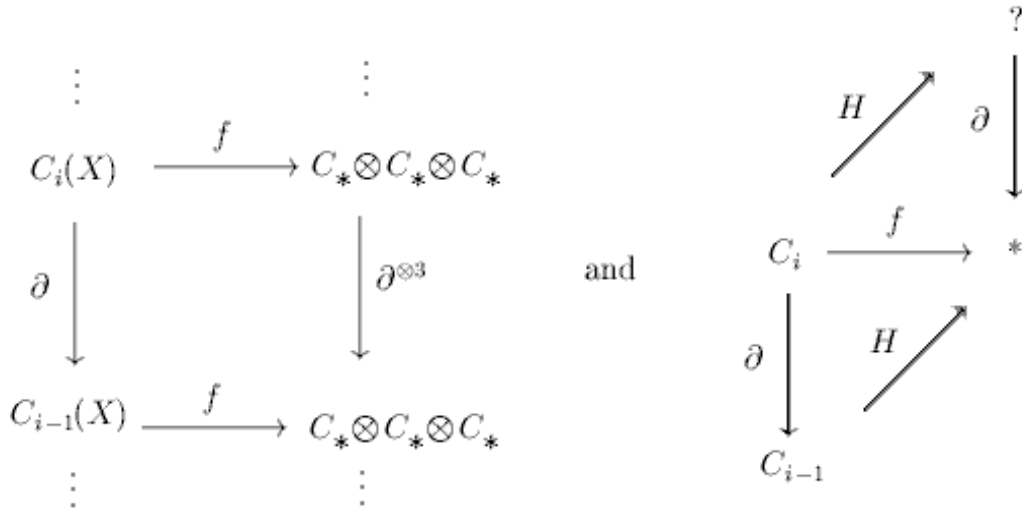


Figure 8: Chain map f and chain homotopy H .

>From the general theory of contractible polyhedra, which include the permutahedra, we know that Δ_P is automatically homotopy coassociative and therefore such an H exists. However, finding an explicit formula for H is computationally difficult because the number of faces in P_n , and hence the dimensions of the vector spaces $C_*(P_n)$ and $C_*(P_n)^{\otimes 3}$, grow

exponentially with n . The calculations on P_2 and P_3 can be done by hand. Referring to the calculations in line ??, we see that

$$[(\Delta \otimes Id)\Delta - (Id \otimes \Delta)\Delta](123) = \partial^{\otimes 3}(12|3 \otimes 2|13 \otimes 23|1 + 1|23 \otimes 13|2 \otimes 3|12).$$

So in this case, we define

$$H(123) = 12|3 \otimes 2|13 \otimes 23|1 + 1|23 \otimes 13|2 \otimes 3|12 \in C_1(P_3)^{\otimes 3}.$$

Then

$$(\partial^{\otimes 3}H + \partial^{\otimes 3}H)(123) = [(\Delta \otimes Id)\Delta - (Id \otimes \Delta)\Delta](123).$$

5 Algorithms

The main contribution of this thesis is a computer implementation that computes the chain homotopy H in low dimensions. The fact that ∂ lowers the dimension and that Δ_P is multiplicative on tensor products allows us to compute the action of H on P_n inductively, i.e., the action of $\partial^{\otimes 3} \circ H = f + H \circ \partial$ is determined by the (known) action of H on lower dimensional permutahedra. Since $\partial^{\otimes 3}(H(P_n)) = (f + H \circ \partial)(P_n)$, we define $H(P_n)$ to be a solution X of the linear system

$$\partial^{\otimes 3}(X) = (f + H \circ \partial)(P_n). \quad (4)$$

In general, there are many such solutions X , each of which determines a different chain homotopy.

The boundary of a face consists of the union of all faces obtained by inserting a bar in every possible way. The boundary operation acts according to the Leibniz rule on strings involving $|$ and \otimes operations. So for example

$$\partial(a_1|a_2|\cdots|a_k) = \partial(a_1)|a_2|\cdots|a_k + a_1|\partial(a_2)|\cdots|a_k + \cdots + a_1|a_2|\cdots|\partial(a_k) \quad (5)$$

and

$$\begin{aligned} \partial(a_1 \otimes a_2 \otimes \cdots \otimes a_k) &= \partial(a_1) \otimes a_2 \otimes \cdots \otimes a_k + a_1 \otimes \partial(a_2) \otimes \cdots \otimes a_k \\ &+ \cdots + a_1 \otimes a_2 \otimes \cdots \otimes \partial(a_k). \end{aligned}$$

This allows us to operate on each factor individually and decompose the boundary operation into a sum of ∂ applied to each set in the term. Since these sets contain n or fewer elements, their boundaries can be calculated generically just once instead of calculating ∂ on each term. A list is loaded at the start of the program that contains the boundary of generic strings of various lengths. The boundary of a particular face is computed by replacing the entries of a generic string with the data from the given face. This makes the boundary calculation a matter of string substitution. For example, if x, y, z are any 3 symbols, $\partial(xyz) = x|yz + y|xz + z|xy + xy|z + xz|y + yz|x$. The faces we obtain in this way accumulate in a set.

Step matrices are used to compute the diagonal Δ_P , however, since step matrices are in one-to-one correspondence with the permutations of a set, the contribution of a step matrix can be obtained directly from its permutation. The following algorithm reads a $u_i \otimes v_i$ pair directly from a permutation of \underline{n} .

Algorithm 2: Reading a $u_i \otimes v_i$ pair from a permutation.

Let $(\rho_1, \rho_2, \dots, \rho_n)$ be a permutation of \underline{n} and let A and B be initially empty strings.

For $i = 1, 2, \dots, n - 1$

 Append ρ_i to A and B

 If $\rho_i < \rho_{i+1}$

 Append a bar to A

 otherwise

 Append a bar to B

Append ρ_n to A and B

$A \otimes B$ is the string representation of the term generated by the step matrix corresponding to $(\rho_1, \rho_2, \dots, \rho_n)$;

Put $A \otimes B$ into standard form by sorting the numbers in each partition from lowest to highest.

Applying algorithm 2 to all of the permutations of \underline{n} produces the terms of $\Delta_P(\underline{n})$ that come directly from step matrices. To complete the set of terms, the terms that result from

applying operations 1A and 1B must be added. These operations can be done with the terms in string form. Recall operation A only changes the row of a number. This affects the v part of a $u_i \otimes v_i$ pair and changes v_i by moving a number one partition to the left. i.e.



Figure 9: Transforming step matrices as strings.

Following the rules of operation 1A, a number can be moved one partition to the left if:

- 1) It is larger than every number in the partition to the left.
- 2) It is not the only element in its partition.

Similarly, analogous to operation 1B, a number can be moved one partition to the right in the u part of a $u_i \otimes v_i$ pair if:

- 1) It is larger than every number in the partition to the right.
- 2) It is not alone in its partition.

Note: If x is a nonzero entry directly below or to the right of another nonzero entry y , then $x > y$ because of the way step matrices are built. Also, lower numbers are blocked by higher numbers, but not vice versa. This makes the blocking condition of operations 1A and 1B extraneous since the higher number already causes the condition to fail.

Transformed terms can be further transformed using either operation, so a recursive transformation function similar to ones that create binary trees generates all the possible transformations. Applying the function to each term and canceling repeated pairs gives the full S-U diagonal.

Similar to the boundary operation, the S-U diagonal operation can be computed for generic faces of various dimensions, which generates a list of terms that gives specific solutions by substituting in specific data. Therefore, these calculations only need to be done once for each dimension.

Finding the S-U diagonal lets us compute f from equation (??). Further, we can compute ∂ , and since we are calculating H inductively, it is known how H acts on the $(n - 2)$ -

dimensional faces produced by ∂ . Therefore $\partial^{\otimes 3} \circ H$ is the only unknown term in $\partial^{\otimes 3} \circ H = f + H \circ \partial$. To find a chain homotopy between g and h , and thus to show Δ_P is homotopy coassociative, we use a coefficient matrix and Gauss-Jordan elimination. The function $H : C_*(P_n) \rightarrow C_*(P_n) \otimes C_*(P_n) \otimes C_*(P_n)$ is a linear map that raises dimension by one. Thus the image of $H(P_n)$ has dimension n , and X is a linear combination of n dimensional faces of $P_n \times P_n \times P_n$. The column space of the coefficient matrix is spanned by the n -dimensional faces of $P_n \times P_n \times P_n$, and the row space is spanned by the $n - 1$ dimensional faces of $P_n \times P_n \times P_n$ that result from applying $\partial^{\otimes 3}$ to the elements of the column space. The computable terms $f(P_n) + H \circ \partial(P_n)$, from equation (??), are loaded as the augmented column of the matrix. Row reducing this matrix provides a solution set for X .

The size of this matrix grows exponentially with respect to n . For $n = 4$, the matrix is already 120,960 by 73,729. However, this matrix is sparse with at most twenty nonzero entries per column. Also, since coefficients are in \mathbb{Z}_2 , all nonzero entries are ones. This means we only have to store the positions of the nonzero entries and not their values. I chose to represent this matrix as a vector of Standard Template Library sets, one set for each row. Each element of a set is the index of a nonzero element in that row. Storing it in this way has several advantages. With a sparse matrix, storing the positions of nonzero elements requires less memory than storing the entire matrix, which makes the space requirement linear with respect to the number of nonzero elements. It also condenses the information so that the matrix does not have to be scanned for the elements. Working modulo two makes adding rows together an exclusive union operation between sets, and since Standard Template Library sets remain sorted, finding the leading one in a row is just a matter of checking the first element in the set, which is constant time. However, when performing row operations, the matrix becomes much less sparse. Choosing the row with the least number of nonzero elements that has a leading one in the column in question slows down the rate at which the density grows. Grouping the nonzero elements by rows allows us to do a linear time search of the rows to find the smallest with a leading one in the column.

6 Results

Applying the algorithms above to P_2 , P_3 and P_4 produces the following results:

$$H(P_2) = 0.$$

$$H(P_3) = 12|3 \otimes 2|13 \otimes 23|1 + 1|23 \otimes 13|2 \otimes 3|12.$$

One solution on P_4 : $H(P_4) =$

$$\begin{aligned} & 12|34 \otimes 24|13 \otimes 4|2|3|1 + 12|34 \otimes 24|13 \otimes 4|3|2|1 + 123|4 \otimes 3|24|1 \otimes 34|2|1 \\ & + 123|4 \otimes 3|2|14 \otimes 34|2|1 + 123|4 \otimes 3|2|14 \otimes 3|24|1 + 124|3 \otimes 4|2|13 \otimes 4|23|1 \\ & + 12|34 \otimes 24|1|3 \otimes 4|2|13 + 12|34 \otimes 24|3|1 \otimes 4|23|1 + 12|34 \otimes 2|14|3 \otimes 24|3|1 \\ & + 12|34 \otimes 2|14|3 \otimes 2|4|13 + 12|34 \otimes 2|14|3 \otimes 4|23|1 + 12|34 \otimes 2|4|13 \otimes 4|23|1 \\ & + 13|24 \otimes 34|1|2 \otimes 4|3|12 + 13|24 \otimes 3|14|2 \otimes 34|2|1 + 13|24 \otimes 3|14|2 \otimes 3|4|12 \\ & + 14|23 \otimes 4|13|2 \otimes 4|3|12 + 1|234 \otimes 14|3|2 \otimes 4|13|2 + 1|234 \otimes 14|3|2 \otimes 4|3|12 \\ & + 1|234 \otimes 4|13|2 \otimes 4|3|12 + 23|14 \otimes 3|24|1 \otimes 34|2|1 + 2|134 \otimes 24|3|1 \otimes 4|23|1 \\ & + 12|34 \otimes 2|1|4|3 \otimes 24|13 + 12|34 \otimes 2|4|1|3 \otimes 24|13 + 13|24 \otimes 3|1|4|2 \otimes 34|12 \\ & + 13|24 \otimes 3|4|1|2 \otimes 34|12 + 12|3|4 \otimes 23|14 \otimes 34|2|1 + 12|3|4 \otimes 23|14 \otimes 3|24|1 \\ & + 12|3|4 \otimes 2|134 \otimes 24|3|1 + 12|3|4 \otimes 2|134 \otimes 4|23|1 + 12|4|3 \otimes 24|13 \otimes 4|23|1 \\ & + 13|2|4 \otimes 3|124 \otimes 34|2|1 + 1|23|4 \otimes 134|2 \otimes 4|3|12 + 1|23|4 \otimes 13|24 \otimes 34|1|2 \\ & + 1|23|4 \otimes 13|24 \otimes 3|14|2 + 1|23|4 \otimes 13|24 \otimes 4|3|12 + 1|23|4 \otimes 3|124 \otimes 34|2|1 \\ & + 1|24|3 \otimes 14|23 \otimes 4|13|2 + 1|24|3 \otimes 14|23 \otimes 4|3|12 + 1|2|34 \otimes 124|3 \otimes 4|23|1 \\ & + 1|2|34 \otimes 124|3 \otimes 4|2|13 + 1|2|34 \otimes 14|23 \otimes 4|13|2 + 1|2|34 \otimes 14|23 \otimes 4|3|12 \\ & + 1|2|34 \otimes 24|13 \otimes 4|23|1 + 1|3|24 \otimes 134|2 \otimes 4|3|12 + 2|13|4 \otimes 23|14 \otimes 34|2|1 \\ & + 2|13|4 \otimes 23|14 \otimes 3|24|1 + 2|14|3 \otimes 24|13 \otimes 4|23|1 + 12|3|4 \otimes 2|13|4 \otimes 234|1 \\ & + 12|3|4 \otimes 2|13|4 \otimes 23|14 + 12|3|4 \otimes 2|1|34 \otimes 24|13 + 12|3|4 \otimes 2|3|14 \otimes 234|1 \\ & + 12|4|3 \otimes 2|14|3 \otimes 24|13 + 13|2|4 \otimes 3|1|24 \otimes 34|12 + 13|4|2 \otimes 3|14|2 \otimes 34|12 \\ & + 1|23|4 \otimes 13|2|4 \otimes 34|12 + 1|23|4 \otimes 13|2|4 \otimes 3|124 + 1|23|4 \otimes 1|3|24 \otimes 134|2 \\ & + 1|23|4 \otimes 3|14|2 \otimes 34|12 + 1|23|4 \otimes 3|1|24 \otimes 34|12 + 1|24|3 \otimes 14|2|3 \otimes 4|123 \\ & + 1|2|34 \otimes 14|2|3 \otimes 4|123 + 1|2|34 \otimes 1|24|3 \otimes 14|23 + 1|2|34 \otimes 1|24|3 \otimes 4|123 \\ & + 1|2|34 \otimes 2|14|3 \otimes 24|13 + 1|3|24 \otimes 3|14|2 \otimes 34|12 + 2|13|4 \otimes 2|3|14 \otimes 234|1. \end{aligned}$$

The choice for X on P_4 displayed here was checked using the boundary algorithm. It has 66 terms which are obtained by setting all independent parameters in the reduced matrix to zero. Thus, it is quite possible that this solution is not minimal in the sense that some other choice may have a fewer number of terms. A minimal solution is preferred if we wish to extend the calculations to P_5 . Finding a minimal solution by choosing some of the independent variables to be nonzero, and implementing a multiprocess algorithm for handling higher dimensional calculations are left as future work.