

Utah State University

DigitalCommons@USU

All Graduate Plan B and other Reports

Graduate Studies

12-2018

Calculating the Cohomology of a Lie Algebra Using Maple and the Serre Hochschild Spectral Sequence

Jacob Kullberg
Utah State University

Follow this and additional works at: <https://digitalcommons.usu.edu/gradreports>



Part of the [Algebraic Geometry Commons](#), [Numerical Analysis and Computation Commons](#), and the [Numerical Analysis and Scientific Computing Commons](#)

Recommended Citation

Kullberg, Jacob, "Calculating the Cohomology of a Lie Algebra Using Maple and the Serre Hochschild Spectral Sequence" (2018). *All Graduate Plan B and other Reports*. 1321.

<https://digitalcommons.usu.edu/gradreports/1321>

This Report is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Plan B and other Reports by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.



CALCULATING THE COHOMOLOGY OF A LIE ALGEBRA
USING MAPLE AND THE SERRE HOCHSCHILD SPECTRAL SEQUENCE

by

Jacob Kullberg

A report submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Mathematics

Approved:

Dr. Ian Anderson
Major Professor

Dr. John R. Stevens
Committee Member

Dr. Mark Fels
Committee Member

Dean of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2018

Copyright © Jacob Kullberg 2018

All Rights Reserved

ABSTRACT

Calculating the cohomology of a Lie algebra
using Maple and the Serre Hochschild spectral sequence

by

Jacob Kullberg, Master of Science

Utah State University, 2018

Major Professor: Dr. Ian Anderson
Department: Mathematics and Statistics

Lie algebra cohomology is an important tool in many branches of mathematics. It is used in the Topology of homogeneous spaces, Deformation theory, and Extension theory. There exists extensive theory for calculating the cohomology of semi simple Lie algebras, but more tools are needed for calculating the cohomology of general Lie algebras. To calculate the cohomology of general Lie algebras, I used the symbolic software program called Maple. I wrote software to calculate the cohomology in several different ways. I wrote several programs to calculate the cohomology directly. This proved to be computationally expensive as the number of differential forms used in each step of the direct calculation is equal to a binomial expansion. The second method used was the Serre–Hochschild spectral sequence. The Serre–Hochschild spectral sequence breaks the problem up into a grid of smaller problems that converge to the total cohomology of the Lie algebra.

(∞ pages)

ACKNOWLEDGMENTS

I would like to thank my wife, Demery, who motivated me to finish and encouraged me every step of the way. I would like to thank Dr. Fels for all his help. I would like to thank Dr. Brown for encouraging me to study math. I would like to thank Dr. Stevens and Gary Tanner for helping me navigate the graduation process. I would also like to thank Dr. Anderson for teaching me about Lie algebras and introducing me to the topic.

Jacob Kullberg

CONTENTS

<i>ABSTRACT</i>	iii
<i>ACKNOWLEDGMENTS</i>	iv
1. <i>Background</i>	1
1.1 Forms	1
1.1.1 Wedge Product	2
1.1.2 Forms revisited	3
1.2 Lie algebra	4
1.2.1 Lie algebra with coefficients	5
1.3 Exterior Derivative	5
1.3.1 Exterior Derivative in a Lie algebra	6
1.3.2 Closed Forms	7
1.3.3 Exact Forms	7
1.4 Quotient Vector Space	8
1.4.1 Definition of a Quotient Vector Space	8
2. <i>Cohomology</i>	9
2.1 Lie algebra cohomology	9
2.2 Calculating the Lie algebra cohomology	9
2.3 Example 1	13
3. <i>Serre-Hochschild spectral sequence</i>	17
3.1 Spectral Sequence	17
3.1.1 Page	18
3.1.2 Square	18
3.1.3 Page table	19
3.2 d_r	19
3.2.1 d_0	20
3.2.2 d_1	20
3.2.3 Chain	21
3.3 Calculating the Serre Hochschild spectral sequence	25
3.3.1 Setup	25
3.3.2 Filtration	25
3.3.3 Creating the squares from the filtration	27
3.4 Example 1	28
3.4.1 Lie Algebra	28
3.4.2 The Setup	28
3.4.3 E_0	32
3.4.4 E_1	33
3.4.5 E_2	37
3.4.6 E_3	38

3.4.7	Cohomology	39
3.5	Example 2	40
3.5.1	Lie algebra	40
3.5.2	E_0	40
3.5.3	E_1	42
3.5.4	E_2	43
3.5.5	E_3	45
3.6	Example 3	47
3.6.1	Lie algebra	47
3.6.2	E_0	47
3.6.3	E_1 Page	49
3.7	Example 4	50
3.7.1	Lie algebra	50
3.7.2	Calculating the exterior differential in a Lie algebra with coefficients in a representation	50
3.7.3	E_0	52
3.7.4	E_1	54
3.7.5	E_2	56
3.7.6	Checking The Answer	57
4.	Programs	58
4.1	Conversion Programs	58
4.1.1	Counting an ordered set of Basis Elements of Differential Forms	58
4.1.2	Program for Form Base Elements: FormToList	59
4.1.3	Tensor To list	65
4.1.4	Tensor valued form to list	70
4.1.5	Complementary Basis	73
4.2	Cohomology Programs	75
4.2.1	Closed Forms	75
4.2.2	Exact Forms	78
4.2.3	Calculating Cohomology	81
4.3	Serre Hofschild Spectral Sequence Programs	83
4.3.1	User Interface Function SHSS	83
4.3.2	Setting up the Spectral Sequence	87
4.3.3	Calculating the next page of the Spectral Sequence	93
4.4	Serre Hofschild Spectral Sequence Toolkit	97
4.4.1	Read Table	97
4.4.2	Read Exact	100
4.4.3	Read Complement	103
4.4.4	Read Cohomology	106
4.4.5	ZigZag	109
4.4.6	SSprojection	112
4.4.7	TCA	115
4.5	Auxiliary Functions	118
4.5.1	SSProjectionInternal	118
4.5.2	SSProjectionInternalLAWC	120
4.5.3	CohoDecompSHSS	122
4.5.4	CohoDecompSHSSLAWC	124
4.5.5	PQ	126

5. <i>Conclusion</i>	129
5.1 Calculating the Cohomology Directly	129
5.2 Calculating the Cohomology using the spectral sequence	133
5.3 Calculating Multiple Pages at once	148
5.4 Relative Chains	167
5.5 Testing Form Range	169
5.6 Toolkit testing	173
5.7 Conclusion	177
<i>Appendix</i>	179
A. <i>Source Code</i>	180
A.1 Conversion Programs	180
A.1.1 FormToList	180
A.1.2 TensorToList	181
A.1.3 TensorValuedFormToList	183
A.1.4 Complementary Basis	186
A.2 Cohomology Programs	191
A.2.1 Closed Forms	191
A.2.2 Exact Forms	193
A.2.3 Calculating Cohomology	194
A.3 Serre Hofschild Spectral Sequence Programs	196
A.3.1 SHSSLAWC2	196
A.3.2 SequenceTableLAWC2	199
A.3.3 NextPageTableLAWC	202
A.4 Serre Hofschild Spectral Sequence Toolkit	212
A.4.1 Read Table	212
A.4.2 Read Exact	213
A.4.3 Read Complement	214
A.4.4 Read Cohomology	215
A.4.5 ZigZag	216
A.4.6 SSprojection	217
A.4.7 TCA	218
A.4.8 SSExtDer	219
A.5 Auxiliary Functions	221
A.5.1 SSProjectionInternal	221
A.5.2 SSProjectionInternalLAWC	223
A.5.3 CohoDecompSHSS	225
A.5.4 CohoDecompSHSSLAWC	227
A.5.5 PQ	230
<i>APPENDICES</i>	232

1. BACKGROUND

Before we start talking about Cohomology there are a few things that I would like to review. To understand Lie algebra cohomology, it is necessary to know about Lie algebras, dual spaces, wedge products, and exterior derivatives. This chapter will give a brief background on these topics.

1.1 Forms

The first topic we shall go over is differential forms and the dual space. The objects that we will be manipulating throughout these calculations will be differential forms.

Definition 1.1.1 (Dual Space): Let V be a vector space with a field F . The dual space V^* is the space of linear functions $\alpha : V \rightarrow F$

The dual space of a vector space is also a vector space. If V is a vector space with dimension n then its dual space V^* also has dimension n . Dual vectors are also called 1-forms. The basis for the dual space we shall be using throughout this paper will be the dual basis.

Definition 1.1.2 (Dual Basis for V^*): Let $\{e_i\}$ be a basis for V then the dual basis for V^* , $\{\theta^i\}$, is defined by $\theta^i(e_k) = \delta_k^i$

In this thesis we will use a subscript to denote an element of the vector space and superscript to denote an element of the dual space. The last type of object we need to discuss is the k -form.

Definition 1.1.3 (Differential p -form): A differential p -form is a C^∞ skew-symmetric covariant tensor field of degree p . [1]

An example of a p-form is the Faraday tensor used in electromagnetism. The Faraday tensor, $F_{\mu\nu}$, is a second degree covariant tensor defined as :

$$F_{\mu\nu} = \frac{\partial A_\nu}{\partial x^\mu} - \frac{\partial A_\mu}{\partial x^\nu} = \begin{bmatrix} 0 & -E_1 & -E_2 & -E_3 \\ E_1 & 0 & B_3 & -B_2 \\ E_2 & -B_3 & 0 & B_1 \\ E_3 & B_2 & -B_1 & 0 \end{bmatrix}$$

Furthermore

$$F_{\nu\mu} = \begin{bmatrix} 0 & E_1 & E_2 & E_3 \\ -E_1 & 0 & -B_3 & B_2 \\ -E_2 & B_3 & 0 & -B_1 \\ -E_3 & -B_2 & B_1 & 0 \end{bmatrix}$$

Now $F_{\mu\nu} = -F_{\nu\mu}$, so this tensor is skew symmetric. So we can say that $F_{\mu\nu}$ are the components of a 2-form

$$F = F_{\mu\nu} dx^\mu \wedge dx^\nu$$

1.1.1 Wedge Product

To further talk about p-forms we must define the \wedge operation.

Definition 1.1.4 (Wedge Product): The symbol \wedge denotes the wedge or exterior product, which is an anti symmetric product between differential forms. Let ω^1 and ω^2 be 1-forms. Then the exterior or wedge product of ω^1 and ω^2 is the 2-form

$$(\omega^1 \wedge \omega^2)(X, Y) = \frac{1}{2}(\omega^1(X)\omega^2(Y) - \omega^1(Y)\omega^2(X))$$

The wedge product of p 1-forms $\omega^1 \dots \omega^p$ is given by

$$(\omega^1 \wedge \dots \wedge \omega^p)(X_1 \dots X_p) = \frac{1}{p!} \det|\omega^i(X_j)|$$

[7]

Let ω be a u-form and let π be a v-form then $\omega \wedge \pi$ is a (u+v)-form. Now a couple more properties of wedge products.

Definition 1.1.5 (Properties of Wedge Product):

- Anti Symmetric:

If α and β are differential k-forms with degrees p,q respectively then $\alpha \wedge \beta = (-1)^{pq} \beta \wedge \alpha$.

- Associative:

Let α, β, γ be in differential k-forms then

$$\alpha \wedge (\beta \wedge \gamma) = (\alpha \wedge \beta) \wedge \gamma$$

- Bilinear:

let $\alpha_1, \alpha_2, \beta_1, \beta_2$ be differential k-forms and $c_1, c_2 \in F$ where F is the field associated with the exterior algebra. then

$$(c_1\alpha_1 + c_2\alpha_2) \wedge \beta_1 = c_1(\alpha_1 \wedge \beta_1) + c_2(\alpha_2 \wedge \beta_1)$$

$$\alpha_1 \wedge (c_1\beta_1 + c_2\beta_2) = c_1(\alpha_1 \wedge \beta_1) + c_2(\alpha_1 \wedge \beta_2)$$

1.1.2 Forms revisited

Now that we have covered wedge products, it is time to go over forms again. Let M be a manifold. The set of all p-forms on a manifold M will be denoted $\Lambda^p(M)$. The collection of all forms on M will be denoted by $\Lambda(M)$. There are no p-forms when $p > d$, where d is the dimension of M .

Definition 1.1.6: Basis for p-forms Let M be a manifold with dimension n. A basis for the p-forms on M in local coordinates is the set

$$\{dx^{i_1} \wedge \cdots \wedge dx^{i_p} \mid 1 \leq i_1 < i_2 < \cdots < i_p \leq n\}$$

The dimension of Λ^p is $\binom{n}{p}$

[7]

Basis Example

Let M be a 3-dimensional manifold where $\omega^1, \omega^2, \omega^3$ form the basis for Λ^1 . The basis for Λ^2 , the 2-forms, would have $\binom{3}{2}$ linear independent forms in it. So Λ^3 has three forms and would be the

following:

$$\Lambda^2 := \{\omega^1 \wedge \omega^2, \omega^1 \wedge \omega^3, \omega^2 \wedge \omega^3\}$$

The basis for Λ^3 would be:

$$\Lambda^3 := \{\omega^1 \wedge \omega^2 \wedge \omega^3\}$$

There is no 4-forms in this example. That is because 4 is larger than the dimension of M .

1.2 Lie algebra

Definition 1.2.1 (Lie algebra): A Lie algebra is a vector space g over some field F together with a binary operation $[-, -] : g \times g \rightarrow g$ called a Lie bracket.

A Lie bracket is an operation with the following axioms:

- Linear

$$[ax + by, z] = a[x, z] + b[y, z]$$

- Anti Commutativity

$$[x, y] = -[y, x]$$

- Jacobi Identity

$$[x, [y, z]] + [z, [x, y]] + [y, [z, x]] = 0$$

Definition 1.2.2 (Structure Equations): Let g be a finite dimensional Lie algebra. It has a basis $\{e_1, e_2, \dots, e_n\}$. The Lie bracket of g defines the structure equations as $[e_j, e_k] = \sum_i^n C_{jk}^i e_i$

Example 1

The Lie algebra $\mathfrak{so}(3)$ consists of all skew symmetric 3x3 matrices.

A basis for $\mathfrak{so}(3)$:

$$e_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}, e_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, e_3 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The Lie bracket between two elements in $\mathfrak{so}(3)$ is calculated using the matrix commutator: $[A, B] =$

$AB - BA$.

Calculating the Lie bracket:

$$\begin{aligned}
 [e_1, e_2] &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} = \\
 & \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = e_3
 \end{aligned}$$

Structure Equations: $[e_1, e_2] = e_3$ $[e_3, e_1] = e_2$ $[e_2, e_3] = e_1$

Calculating the Lie bracket of two elements of g using the properties of a Lie bracket.

Let

$$x = 2e_1 - e_3, \quad y = 8e_2$$

$$\begin{aligned}
 [x, y] &= [2e_1 - e_3, 8e_2] = 2[e_1, 8e_2] - [e_3, 8e_2] \\
 &= 16[e_1, e_2] - 8[e_3, e_2] = 16e_3 + 8e_1
 \end{aligned}$$

1.2.1 Lie algebra with coefficients

Definition 1.2.3 (Representation): Let g be a Lie algebra. A representation ρ is a homomorphism

$$\rho : g \rightarrow gl(V)$$

such that the Lie bracket on $gl(V)$ is the commutator.

Definition 1.2.4 (Lie algebra with coefficients in a representation): Let g be an n dimensional lie algebra with a basis $\{e_1, e_2, \dots, e_n\}$ and g^* be the dual space with a dual basis $\{\theta^1, \theta^2, \dots, \theta^n\}$. Now V is a vector space of dimension m with a basis $\{x_1, x_2, \dots, x_m\}$. Then a form of g with coefficients in the representation V , ω , is an element of $\Lambda^k(g^*, V)$, $\omega : g \times g \times \dots \times g \rightarrow V$ where

$$\omega = Cx_a \theta^i \wedge \theta^j \wedge \dots \theta^k$$

1.3 Exterior Derivative

An important operation we must study before we can define cohomology is the exterior derivative.

The exterior derivative of p -form is a $p+1$ -form.

1.3.1 Exterior Derivative in a Lie algebra

Definition 1.3.1 (Exterior Derivative in a Lie algebra): Let g be a finite Lie algebra with a basis $\{e_1, e_2, \dots, e_n\}$. Let the structure equations be $\sum_i^n C_{jk}^i e_i = [e_j, e_k]$. Then if g has a dual basis $\{\theta^1, \theta^2, \dots, \theta^n\}$, $d\theta^i = -\frac{1}{2} \sum_j^n \sum_k^n C_{jk}^i \theta^j \wedge \theta^k$.

Definition 1.3.2 (Exterior Derivative expanded to higher k-forms): The Exterior Derivative can be expanded to two forms and beyond by using Leibniz rule:

$$d(\theta^i \wedge \theta^j) = d\theta^i \wedge \theta^j - \theta^i \wedge d\theta^j$$

and

$$d(d\theta) = 0$$

One important thing to note is that $d : \Lambda^p \rightarrow \Lambda^{p+1}$. The second property $d(d\theta) = 0$ is very important. It means that exterior derivative of the exterior derivative of any k-form is always zero. This fact will be very important in the next section.

Example

Example Lie algebra Winternitz[5,6]

	e_1	e_2	e_3	e_4	e_5
e_1	0	0	0	0	0
e_2	0	0	0	e_1	e_3
e_3	0	0	0	0	e_1
e_4	0	$-e_1$	0	0	e_2
e_5	0	$-e_3$	$-e_1$	$-e_2$	0

Let's calculate $d(\theta^2)$

$$d(\theta^2) = -\frac{1}{2}(-\theta^5 \wedge \theta^4 + \theta^4 \wedge \theta^5) = -\theta^4 \wedge \theta^5$$

$d(\theta^1) = -\theta^2 \wedge \theta^4 - \theta^3 \wedge \theta^5$	$d(\theta^3) = -\theta^2 \wedge \theta^5$
$d(\theta^2) = -\theta^4 \wedge \theta^5$	$d(\theta^4) = 0$
	$d(\theta^5) = 0$

1.3.2 Closed Forms

There are two very important types of forms that need to be understood before we can define the cohomology. These two types of forms are defined by their relationship to the exterior derivative. The first one is the closed form.

Definition 1.3.3 (Closed Form): A closed form ω is a form such that $d\omega = 0$. The closed forms are the kernel of the exterior derivative.

Example

Example Lie algebra Winternitz[5,6]

$d(\theta^1) = -\theta^2 \wedge \theta^4 - \theta^3 \wedge \theta^5$	$d(\theta^3) = -\theta^2 \wedge \theta^5$
$d(\theta^2) = -\theta^4 \wedge \theta^5$	$d(\theta^4) = 0$
	$d(\theta^5) = 0$

The Closed Forms

$d(\theta^5) = 0$	$d(\theta^4) = 0$
-------------------	-------------------

Example of a closed 2-form $(\theta^2 \wedge \theta^5)$

$$\begin{aligned} d(\theta^2 \wedge \theta^5) &= d(\theta^2) \wedge \theta^5 - \theta^2 \wedge d(\theta^5) \\ &= -\theta^4 \wedge \theta^5 \wedge \theta^5 - \theta^2 \wedge (0) = 0 \end{aligned}$$

1.3.3 Exact Forms

Definition 1.3.4 (Exact Form): A form, ω , is said to be exact if there exists an $\alpha \in \mathfrak{g}^*$ such that $d\alpha = \omega$. So $\omega \in \Lambda^k$ and $\alpha \in \Lambda^{k-1}$.

Example

Lie algebra Winternitz[5,6]

$d(\theta^1) = -\theta^2 \wedge \theta^4 - \theta^3 \wedge \theta^5$	$d(\theta^3) = -\theta^2 \wedge \theta^5$
$d(\theta^2) = -\theta^4 \wedge \theta^5$	$d(\theta^4) = 0$
	$d(\theta^5) = 0$

$d(\theta^1) = -\theta^2 \wedge \theta^4 - \theta^3 \wedge \theta^5$	$d(\theta^3) = -\theta^2 \wedge \theta^5$
$d(\theta^2) = -\theta^4 \wedge \theta^5$	

Note because $d(d\omega) = 0$, all exact forms are closed.

1.4 Quotient Vector Space

1.4.1 Definition of a Quotient Vector Space

The last thing we need to define before we can define the cohomology is a quotient vector space.

Definition 1.4.1 (Quotient Vector Space): Let V be a vector space. Let W be a subspace of V . The quotient vector space V/W is the set of equivalent classes $[v]$ where $v_1 \cong v_2$ if $v_1 - v_2 \in W$.

Two elements of V are congruent if their difference is in W .

Example

Let V be the vector space with basis $\{x_1, x_2, x_3\}$ Let W be a subspace with basis $\{x_2\}$. Then

$$V/W = \{[x_1], [x_3]\}$$

Furthermore

$$x_1 + 3x_2 \cong x_1 - 2x_2$$

because

$$x_1 + 3x_2 - (x_1 - 2x_2) = 5x_2 \in W$$

2. COHOMOLOGY

2.1 Lie algebra cohomology

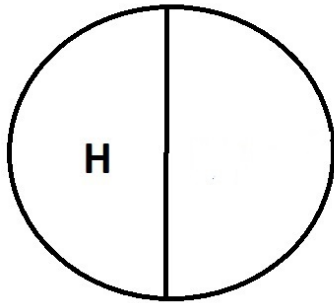
Now we are ready to define what the cohomology of a Lie algebra is. Essentially the cohomology of a Lie algebra is the closed forms modulo the exact forms.

Definition 2.1.1 (Cohomology of a Lie algebra): Let \mathfrak{g} be a Lie algebra. The Cohomology H^p is

$$H^p = \frac{\{\omega \in \Lambda^p(\mathfrak{g}^*) | d(\omega) = 0\}}{\{\omega \in \Lambda^p(\mathfrak{g}^*) | \omega = d(\nu)\}}$$

2.2 Calculating the Lie algebra cohomology

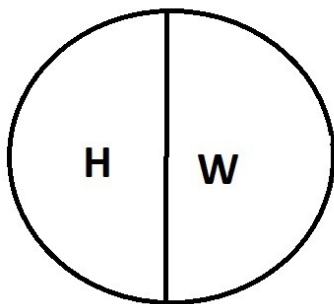
Now let \mathfrak{g} be a Lie algebra. To calculate the cohomology of a \mathfrak{g} , we will start with the first link in the cochain complex. With a general Lie algebra that would be the 1-forms. Because zero forms are constants, there are no exact one forms in a general Lie algebra. When working with a Lie algebra with coefficients in a representation there are nontrivial zero forms so the chain would start with the zero forms instead of 1-forms. In either case, in the first link of the chain there are no exact forms so we can then just solve for the basis of closed forms. With the absence of exact forms, the closed forms will form the cohomology of the 1-forms. The cohomology space of 1-forms will be denoted by H in the following diagram.

 ω_1

H is the cohomology

ω_1 is the space of 1-forms

Now that we have calculated the cohomology of the 1-forms, the next step would be to calculate the cohomology of the 2-forms. To do this we would find the complementary basis to H in ω_1 . The complementary basis to all the closed forms will be denoted by W.

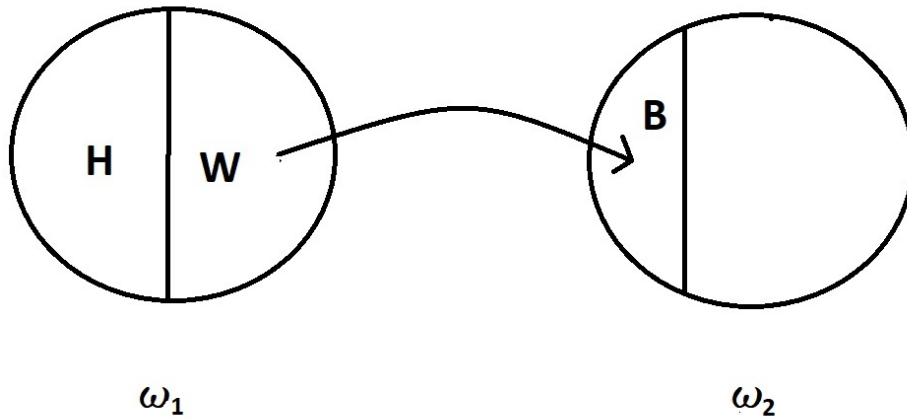
 ω_1

W is the complementary space

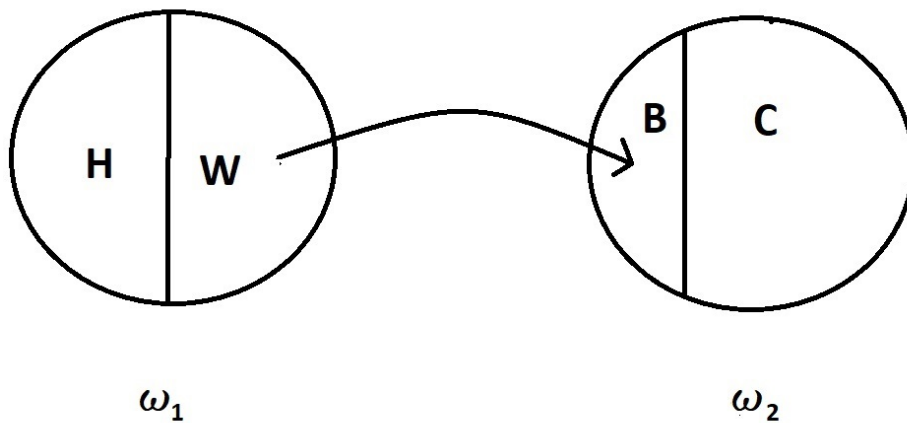
H is the cohomology

ω_1 is the space of 1-forms

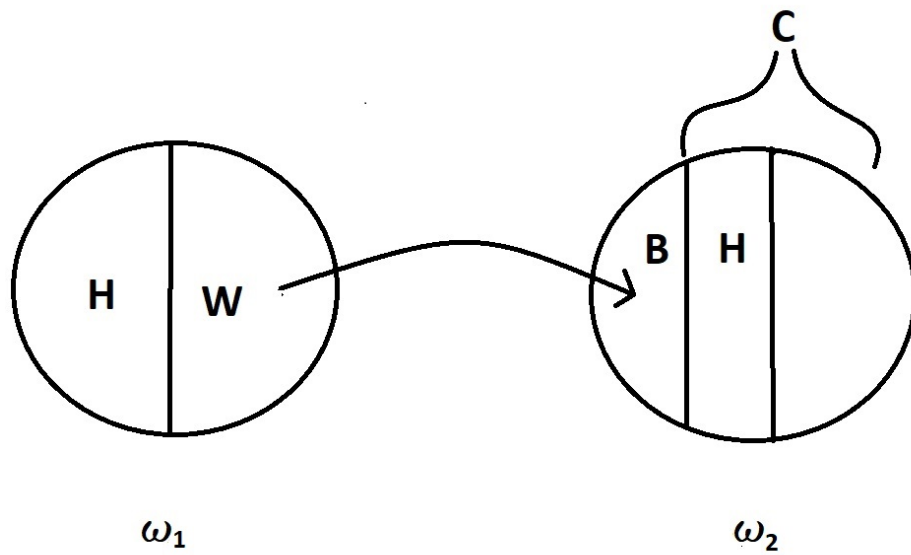
Now the exterior derivatives of the elements of W will form a basis for exact forms in ω_2 . This space of exact forms will be denoted by B .



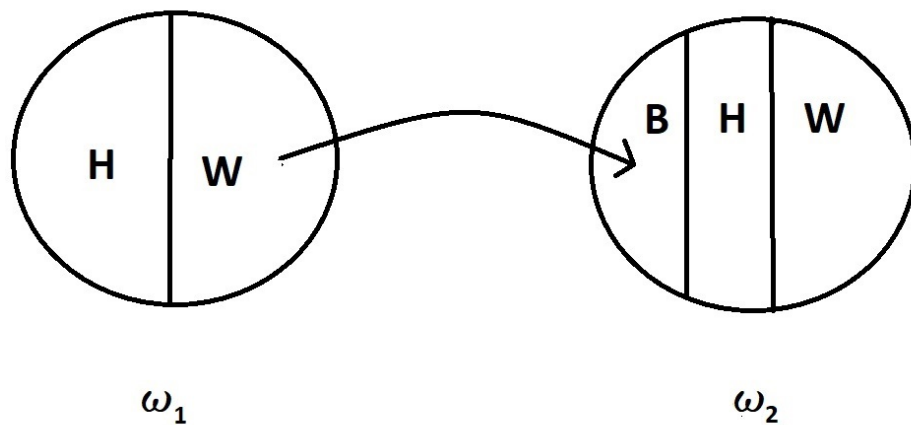
From there we can calculate the complement of B in ω_2 . This space will be called C .



Next we will calculate the closed forms in C . Because there were no exact forms in C , this will be H , the cohomology of ω_2 .



Now that we have calculated the cohomology of ω_2 , the next step would be to continue the process and calculate the cohomology of ω_3 . To do this, we will find complementary basis of the combined H and B in ω_2 . This will give us a basis for all the non-closed forms in ω_2 . Using this complementary space, W, we can then find a basis for the exact forms in ω_3 and the process starts over again.



2.3 Example 1

In this example we will use the Lie algebra defined by the following multiplication table.

	e_1	e_2	e_3	e_4	e_5
e_1	0	0	0	0	0
e_2	0	0	0	0	e_1
e_3	0	0	0	e_1	e_2
e_4	0	0	$-e_1$	0	0
e_5	0	$-e_1$	$-e_2$	0	0

The next table shows the exterior derivatives of the basis elements of the Lie algebra. This will be used to calculate the closed and exact forms.

$d(\theta^1) = -\theta^2 \wedge \theta^5 - \theta^3 \wedge \theta^4$	$d(\theta^2) = -\theta^3 \wedge \theta^5$
$d(\theta^3) = 0$	$d(\theta^4) = 0$
$d(\theta^5) = 0$	

The closed forms is the kernel of the exact forms. So using the above table we can write a transformation matrix. The rows signify 2-forms. The columns are the 1-forms. The null space of this matrix will be the basis for closed forms.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The null space of this matrix is:

$$\left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

This means that the basis for our closed form is $\{\theta^3, \theta^4, \theta^5\}$. Now because there are no exact 1-forms in this example, the set $\{\theta^3, \theta^4, \theta^5\}$ is H , the cohomology of the 1-forms. Now by taking the complementary basis for H in ω_1 we will get a basis for W , the space of nonclosed forms in ω_1 . The exterior derivatives the basis of W will give us a basis for the exact forms in ω_2 . The complementary basis through inspection is clearly $\{\theta^1, \theta^2\}$.

Now that we have W, we can find the basis for the B, the space of exact forms, in ω_2 . The exterior derivative of the basis elements of W will give us a basis for B. The exterior derivatives of $\{\theta^1, \theta^2\}$ is $\{-\theta^2 \wedge \theta^5 - \theta^3 \wedge \theta^4, -\theta^3 \wedge \theta^5\}$. Now that we have a basis for B, we will find the complementary basis of B in ω_2 . This is the space denoted by C.

$$C = \{\theta^1 \wedge \theta^2, \theta^1 \wedge \theta^3, \theta^1 \wedge \theta^4, \theta^1 \wedge \theta^5, \theta^2 \wedge \theta^3, \theta^2 \wedge \theta^4, \theta^2 \wedge \theta^5, \theta^4 \wedge \theta^5\}$$

Now there are no exact forms in C. Everything is either closed or part of complementary space W. To find the basis of closed forms in C we will take the exterior derivative of C and find the null space. The matrix associated with this is :

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The null space of this matrix is :

$$\left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$

Now these vectors are using C as the basis elements not ω_2 . So the closed forms in C are

$$\{\theta^4 \wedge \theta^5, \theta^2 \wedge \theta^5, \theta^2 \wedge \theta^4 + \theta^1 \wedge \theta^5, \theta^2 \wedge \theta^3\}$$

Because there are no exact forms in C, this list of closed forms is H, the cohomology. Next lets find the cohomology of ω_3 . To do this we want to find W in ω_2 , the complementary space. To do

this we would calculate the complementary basis of H in C.

$$W = \{\theta^1 \wedge \theta^2, \theta^1 \wedge \theta^3, \theta^1 \wedge \theta^4, \theta^1 \wedge \theta^5\}$$

The exterior derivative of this is :

$$d(W) = B = \{\theta^1 \wedge \theta^3 \wedge \theta^5 - \theta^2 \wedge \theta^3 \wedge \theta^4, \theta^2 \wedge \theta^3 \wedge \theta^5, \theta^2 \wedge \theta^4 \wedge \theta^5, -\theta^3 \wedge \theta^4 \wedge \theta^5\}$$

Now that we have B we can start the algorithm over again for ω_3 . We will calculate the complementary basis for B in ω_3 which will give us the closed forms and complementary space.

This space C is

$$C = \{\theta^1 \wedge \theta^2 \wedge \theta^3, \theta^1 \wedge \theta^2 \wedge \theta^4, \theta^1 \wedge \theta^2 \wedge \theta^5, \theta^1 \wedge \theta^3 \wedge \theta^4, \theta^1 \wedge \theta^3 \wedge \theta^5, \theta^1 \wedge \theta^4 \wedge \theta^5\}$$

Now taking the exterior derivative again and changing that into a matrix gives:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 \end{bmatrix}$$

The null space of this is :

$$\left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$

This means the cohomology of ω_3 is :

$$H = \{\theta^1 \wedge \theta^4 \wedge \theta^5, \theta^1 \wedge \theta^3 \wedge \theta^5, -\theta^1 \wedge \theta^2 \wedge \theta^5 + \theta^1 \wedge \theta^3 \wedge \theta^4, \theta^1 \wedge \theta^2 \wedge \theta^3\}$$

Now the last step would be to find the cohomology of ω_4 . To do this we will find the complementary space, W, in ω_3 . This will be the complementary basis of H in C.

$$W = \{\theta^1 \wedge \theta^2 \wedge \theta^4, \theta^1 \wedge \theta^2 \wedge \theta^5\}$$

Taking the exterior derivative of W will give us a basis for B in ω_4 .

$$d(W) = B = \{-\theta^1 \wedge \theta^3 \wedge \theta^4 \wedge \theta^5, -\theta^2 \wedge \theta^3 \wedge \theta^4 \wedge \theta^5\}$$

The complementary basis of B in ω_4 is C.

$$C = \{\theta^1 \wedge \theta^2 \wedge \theta^3 \wedge \theta^4, \theta^1 \wedge \theta^2 \wedge \theta^3 \wedge \theta^5, \theta^1 \wedge \theta^2 \wedge \theta^4 \wedge \theta^5\}$$

The exterior derivative of all these forms is zero. So all of them are closed. Thus the cohomology of ω_4 is

$$H = \{\theta^1 \wedge \theta^2 \wedge \theta^3 \wedge \theta^4, \theta^1 \wedge \theta^2 \wedge \theta^3 \wedge \theta^5, \theta^1 \wedge \theta^2 \wedge \theta^4 \wedge \theta^5\}$$

This ends the process. There is no closed or exact 5-forms in this example because the basis for the Lie algebra was 5. Putting all of the cohomology classes together gives the total cohomology of this Lie algebra which is:

$$H = \{\{\theta^3, \theta^4, \theta^5\}, \{\theta^4 \wedge \theta^5, \theta^2 \wedge \theta^5, \theta^2 \wedge \theta^4 + \theta^1 \wedge \theta^5, \theta^2 \wedge \theta^3\}, \\ \{\theta^1 \wedge \theta^4 \wedge \theta^5, \theta^1 \wedge \theta^3 \wedge \theta^5, -\theta^1 \wedge \theta^2 \wedge \theta^5 + \theta^1 \wedge \theta^3 \wedge \theta^4, \theta^1 \wedge \theta^2 \wedge \theta^3\}, \{\theta^1 \wedge \theta^2 \wedge \theta^3 \wedge \theta^4, \theta^1 \wedge \theta^2 \wedge \theta^3 \wedge \theta^5, \theta^1 \wedge \theta^2 \wedge \theta^4 \wedge \theta^5\}\}$$

3. SERRE-HOCHSCHILD SPECTRAL SEQUENCE

3.1 Spectral Sequence

Let \mathfrak{g} be a Lie algebra, \mathfrak{h} a sub-algebra, and A a module over \mathfrak{g} . Then there exists a spectral sequence

$$\{E_r^{p,q}, d_r^{p,q} : E_r^{p,q} \rightarrow E^{p+r, q-r+1}\}$$

with the following properties

i) $E_1^{p,q} = H^q(\mathfrak{h}; \text{Hom}(\Lambda^p(\mathfrak{g}/\mathfrak{h}), A))$

ii) $E_2^{p,0} = H^p(\mathfrak{g}, \mathfrak{h}; A)$

iii) if \mathfrak{g} is an ideal, then $E_2^{p,q} = H^p(\mathfrak{g}/\mathfrak{h}; H^q(\mathfrak{h}; A))$

iv) the term E_∞ is adjoint to $H^*(\mathfrak{g}; A)$; and the natural homomorphisms $H^q(\mathfrak{g}; A) \rightarrow H^q(\mathfrak{h}; A)$, $H^p(\mathfrak{g}, \mathfrak{h}; A) \rightarrow H^p(\mathfrak{g}; A)$ can be represented as the compositions:

$$H^q(\mathfrak{g}; A) \rightarrow E_\infty^{0,q} \rightarrow E_1^{0,q} = H^q(\mathfrak{h}; A)$$

$$H^p(\mathfrak{g}, \mathfrak{h}; A) = E_2^{p,0} \rightarrow E_\infty^{p,0} \rightarrow H^p(\mathfrak{g}; A)$$

[2]

This spectral sequence is called the Serre Hochschild spectral sequence. From property iv we can see that the sequence converges to the cohomology of the Lie algebra. We will use this sequence as an alternate method to calculate the cohomology.

3.1.1 Page

Each step in the Serre Hochschild spectral sequence is called a page. The r th page in the sequence is denoted by E_{r-1} . This is because the pages start at 0 and not 1. The collection of all $\{E_r^{p,q}, d_r^{p,q} : E_r^{p,q} \rightarrow E^{p+r, q-r+1}\}$ for a specific r is E_r . So each page is not just a collection of squares, but all the chains and pairings of forms that are created by the differential on that page.

3.1.2 Square

A page is made up of squares. Each square has a (p, q) coordinate. It is denoted by $E_r^{p,q}$. The type of forms that are in each form comes from the filtration process. The degree of the forms in the square is $p + q$. The only forms that are in each square are forms that were closed in the previous page and not exact.

3.1.3 Page table

Each page is represented by a grid. The (p,q) pair determines the placement of the square on the grid. The p is the x coordinate. The q is the y coordinate. The bottom left corner is always $(0,0)$.

Square placement

$\mathbf{E}^{(0,2)}$	$\mathbf{E}^{(1,2)}$	$\mathbf{E}^{(2,2)}$	$\mathbf{E}^{(3,2)}$
$\mathbf{E}^{(0,1)}$	$\mathbf{E}^{(1,1)}$	$\mathbf{E}^{(2,1)}$	$\mathbf{E}^{(3,1)}$
$\mathbf{E}^{(0,0)}$	$\mathbf{E}^{(0,1)}$	$\mathbf{E}^{(2,0)}$	$\mathbf{E}^{(3,0)}$

3.2 d_r

The differential d_r is

$$d_r = \Pi^{p+r, q-r+1} \circ d$$

d_r is just the part of the exterior derivative of a form that projects into a specific square depending on the page. Which square it is projecting into is determined by the following formula:

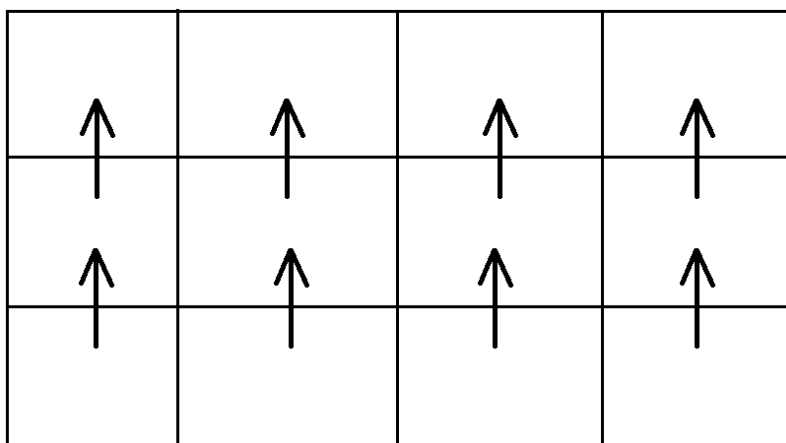
$$d_r : E_r^{p,q} \rightarrow E_r^{p+r, q-r+1}$$

3.2.1 d_0

The differential in the first page of the sequence is d_0 . The formula for d_0 is

$$d_0 : E_0^{p,q} \rightarrow E_0^{p,q+1}$$

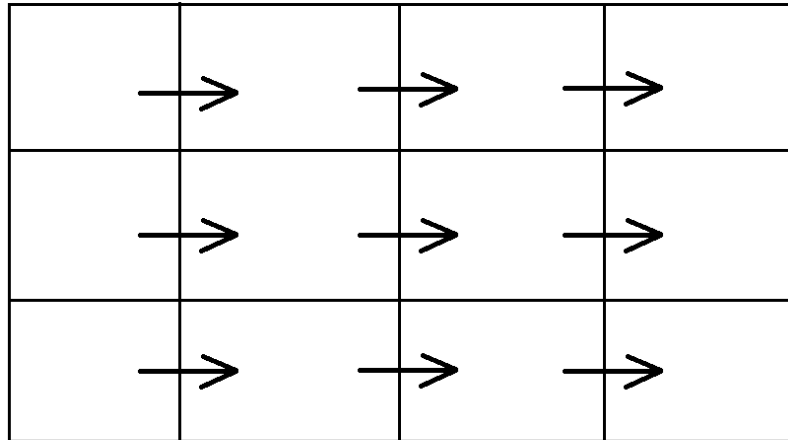
So d_0 is only the part of the exterior derivative that is projected into the square above.

3.2.2 d_1

The differential in the second page of the sequence is d_1 . The formula for d_1 is

$$d_1 : E_1^{p,q} \rightarrow E_1^{p+1,q}$$

So d_1 is the part of the exterior derivative that is projecting into the square to the right.



3.2.3 Chain

One important concept related to the sequence and the differential is the concept of chains. First we must define what it means for a form to be exact and closed on the page.

Definition 3.2.1 (Closed form on the page): Let $\alpha \in \Lambda^{p+q}$ and $\alpha \in E_r^{p,q}$. Now α is said to be closed on page r is $d_r(\alpha) = 0$

This definition is recursive. A form is only on page r if it was closed on page $r - 1$. Which means that $d_{r-1}(\alpha) = 0$ as well and so forth.

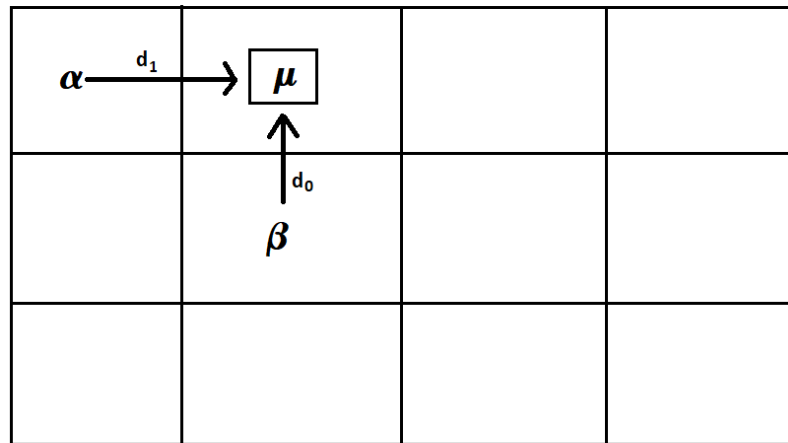
Definition 3.2.2 (Exact form on the page): Let $\alpha \in \Lambda^{p+q}$ and $\alpha \in E_r^{p,q}$. Now α is said to be exact if there exists a $\beta \in \Lambda^{p+q-1}$ and $\beta \in E^{p-r,q+r-1}$ such that $d_r(\beta) = \alpha$.

There are times when the d_r of a form projects into a square but it does not equal zero, but a form that is not on the page. In this case we will create a chain. This chain will cause the form to be closed on the current page.

Chain Example

Let $\alpha \in E_1^{0,2}$. Let $\mu \in E_0^{1,2}$ and $\beta \in E_0^{1,1}$. Let $d_0(\beta) = \mu$. Now this means that β is not closed and μ is exact. Thus neither will be on page 1. Now let $d_1(\alpha) = \mu$. Now μ is not on the 1st page because it was exact on page 0. So

$$d_1(\alpha) = \mu = d_0(\beta)$$



So we will create a chain using beta.

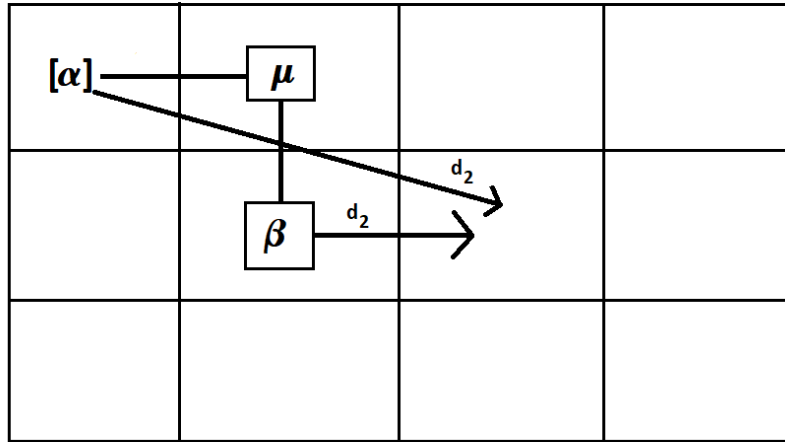
$$d_1(\alpha) - d_0(\beta) = 0$$

So the chain will be

$$[\alpha] = \alpha - \beta$$

Now $[\alpha]$ is a closed form on page 1. To find the differential of a chain first you take the exterior derivative of the entire chain. Then you project it into the appropriate square.

Chain



$$[\alpha] = \alpha - \beta$$

Chain creation

Let $d_r(\beta) = n\mu$ and $d_{r+1}(\alpha) = m\mu$ where $n, m \in R$. Then α will be turned into a chain.

$$[\alpha] = \alpha - \frac{n}{m}\beta$$

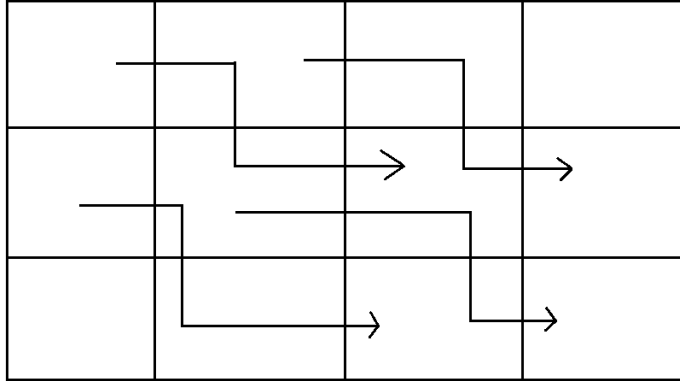
now

$$d_{r+1}([\alpha]) = 0$$

d_2

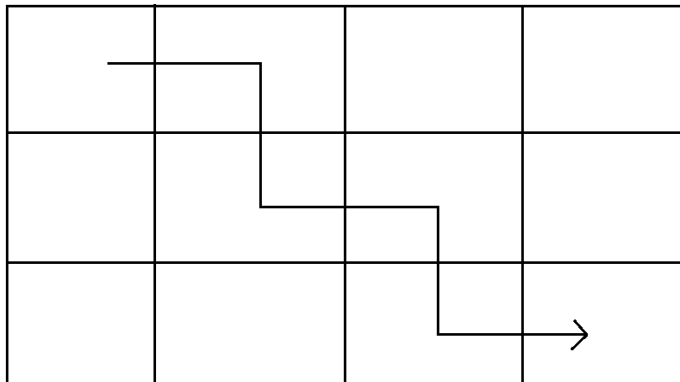
The differential for page 2 is d_2 The formula for d_2 is

$$d_2 : E_2^{p,q} \rightarrow E_2^{p+2,q-1}$$


 d_3

The differential for page 3 is d_3 The formula for d_3 is

$$d_3 : E_3^{p,q} \rightarrow E_3^{p+3,q-2}$$



3.3 Calculating the Serre Hochschild spectral sequence

3.3.1 Setup

Let \mathfrak{g} be an algebra with \mathfrak{h} a sub-algebra. Then \mathfrak{g} can be decomposed into

$$\mathfrak{g} = \mathfrak{m} \oplus \mathfrak{h}$$

where \mathfrak{m} is a complement. This sum is not a lie algebra direct sum but a vector space direct sum. Now let x_i be the basis of \mathfrak{h} and let y_a be the basis of \mathfrak{m} . And let α^i be the dual to x_i and β^a be the dual to y_a . Now because \mathfrak{h} is a sub algebra the lie bracket $[x_n, x_m] \in \mathfrak{h}$. now $d\beta$ can be written as a $\alpha \wedge \alpha$ term, a $\alpha \wedge \beta$ term, and $\beta \wedge \beta$ term.

$$d\beta = a\alpha \wedge \alpha + b\alpha \wedge \beta + c\beta \wedge \beta$$

Now if you were to hook $d\beta$ with two x 's terms you would get

$$d\beta(x, x) = a$$

Now $\beta[x, x] = 0$ so $a=0$. Which means

$$d\beta = b\alpha \wedge \beta + c\beta \wedge \beta$$

and so the structure equations are of the form

$$d\beta = \alpha \wedge \beta + \beta \wedge \beta$$

3.3.2 Filtration

To create the filtration we will first create the Λ 's defined in the previous chapter. It will be done using the new α and β notation. To save space we will start using an implied wedge notation. That means

$$\alpha \wedge \beta = \alpha\beta$$

and when not using specific forms

$$\alpha \wedge \alpha = \alpha^2$$

When using specific forms of the same type

$$\alpha_1 \wedge \alpha_2 = \alpha_{12}$$

$$\alpha_1 \wedge \beta_1 \wedge \beta_3 = \alpha_1 \beta_{13}$$

So now back to defining Λ .

$$\Lambda^1 = \{\alpha, \beta\}$$

$$\Lambda^2 = \{\alpha^2, \alpha\beta, \beta^2\}$$

$$\Lambda^3 = \{\alpha^3, \alpha^2\beta, \alpha\beta^2, \beta^3\}$$

$$\Lambda^4 = \{\alpha^4, \alpha^3\beta, \alpha^2\beta^2, \alpha\beta^3, \beta^4\}$$

Now the definition of the filtration is:

$$F^p \Lambda^{p+q} = \{\omega \in \Omega^{p+q} \mid \omega(x_1 \dots x_q y_{q+1} \dots y_{p+q}) = 0\}$$

β is the annihilator of x . So β hook $x = 0$. So if ω has q or less α 's then $\omega(x_1 \dots x_q y_{q+1} \dots y_{p+q}) = 0$ because one of the x 's must hook up with a β .

The filtration:

$$F^0 \Lambda^1 = \{\alpha, \beta\}$$

$$F^1 \Lambda^1 = \{\beta\}$$

$$F^2 \Lambda^1 = \{0\}$$

$$F^0 \Lambda^2 = \{\alpha^2, \alpha\beta, \beta^2\}$$

$$F^1 \Lambda^2 = \{\alpha\beta, \beta^2\}$$

$$F^2 \Lambda^2 = \{\beta^2\}$$

$$F^0 \Lambda^3 = \{\alpha^3, \alpha^2\beta, \alpha\beta^2, \beta^3\}$$

$$F^1 \Lambda^3 = \{\alpha^2 \beta, \alpha \beta^2, \beta^3\}$$

$$F^2 \Lambda^3 = \{\alpha \beta^2, \beta^3\}$$

$$F^3 \Lambda^3 = \{\beta^3\}$$

These subspaces reflect the fact all you know is β .

You start with \mathfrak{h} and you take it's annihilators and you get the β 's. You cannot count the α 's nor can you talk about the α^n ! As a vector space all you can do is talk about the quotients

3.3.3 Creating the squares from the filtration

We will now use the filtration to figure out what forms are in each square of Page 0. The squares will be quotient space created from the following quotient:

$$E_0^{pq} = \frac{F^p \Lambda^{p+q}}{F^{p+1} \Lambda^{p+q}}$$

Examples:

$$E_0^{01} = \frac{F^0 \Lambda^1}{F^1 \Lambda^1} = \{[\alpha]_0\}$$

$[\alpha]_0$ means that $[\alpha]$ is module β and is on the first page

$$E_0^{10} = \frac{F^1 \Lambda^1}{F^0 \Lambda^2} = \{[\beta]_0\}$$

Now that we have shown how the filtration works and how to create the squares all that is left is to calculate the sequence.

3.4 Example 1

Now we are ready to solve our first example using the spectral sequence.

3.4.1 Lie Algebra

First we will define a Lie algebra by its Multiplication table.

	e_1	e_2	e_3	e_4	e_5
e_1	0	0	0	0	e_1
e_2	0	0	e_1	0	0
e_3	0	$-e_1$	0	0	$e_2 + e_4$
e_4	0	0	0	0	$e_1 + e_4$
e_5	$-e_1$	$-e_2 - e_4$	0	$-e_1 - e_4$	0

Next we will calculate the total exterior differential of the one forms. This will be useful in calculating the differentials on each page.

$d(\theta_1) = -\theta_1\theta_5 + \theta_2\theta_3 - \theta_4\theta_5$	$d(\theta_3) = 0$
$d(\theta_2) = -\theta_2\theta_5$	$d(\theta_4) = -\theta_2\theta_5 - \theta_4\theta_5$
	$d(\theta_5) = 0$

3.4.2 The Setup

The subalgebra

Now that we've defined our Lie algebra, the next step is to pick a subalgebra. One clear subalgebra would be e_1 and e_2 . This would make e_3 , e_4 , and e_5 the complementary space. To denote this we will replace the subalgebra with x 's and the complementary space with y 's. The new multiplication table is as follows:

	x_1	x_2	y_1	y_2	y_3
x_1	0	0	0	0	x_1
x_2	0	0	x_1	0	0
y_1	0	$-x_1$	0	0	$x_2 + y_2$
y_2	0	0	0	0	$x_1 + y_2$
y_3	$-x_1$	$-x_2 - y_2$	0	$-x_1 - y_2$	0

Furthermore we will use alpha to denote the dual to the subalgebra and beta for the dual of the complementary space. This would make the new table of differentials as follows:

$d(\alpha_1) = -\alpha_1\beta_3 + \alpha_2\beta_1 - \beta_2\beta_3$	$d(\beta_1) = 0$
$d(\alpha_2) = -\alpha_2\beta_3$	$d(\beta_2) = -\alpha_2\beta_3 - \beta_2\beta_3$
	$d(\beta_3) = 0$

The filtration

Now that we have identified the subalgebra and the complementary space we can make the filtration.

$$F^0\Lambda^1 = \{\alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3\}$$

$$F^1\Lambda^1 = \{\beta_1, \beta_2, \beta_3\}$$

$$F^2\Lambda^1 = \{0\}$$

$$F^0\Lambda^2 = \{\alpha_{12}, \alpha_1\beta_1, \alpha_1\beta_2, \alpha_1\beta_3, \alpha_2\beta_1, \alpha_2\beta_2, \alpha_2\beta_3, \beta_{12}, \beta_{13}, \beta_{23}\}$$

$$F^1\Lambda^2 = \{\alpha_1\beta_1, \alpha_1\beta_2, \alpha_1\beta_3, \alpha_2\beta_1, \alpha_2\beta_2, \alpha_2\beta_3, \beta_{12}, \beta_{13}, \beta_{23}\}$$

$$F^2\Lambda^2 = \{\beta_{12}, \beta_{13}, \beta_{23}\}$$

$$F^1\Lambda^3 = \{\alpha_{12}\beta_1, \alpha_{12}\beta_2, \alpha_{12}\beta_3, \alpha_1\beta_{12}, \alpha_1\beta_{13}, \alpha_1\beta_{23}, \alpha_2\beta_{12}, \alpha_2\beta_{13}, \alpha_2\beta_{23}, \beta_{123}\}$$

$$F^2\Lambda^3 = \{\alpha_1\beta_{12}, \alpha_1\beta_{13}, \alpha_1\beta_{23}, \alpha_2\beta_{12}, \alpha_2\beta_{13}, \alpha_2\beta_{23}, \beta_{123}\}$$

$$F^3\Lambda^3 = \{\beta_{123}\}$$

$$F^2\Lambda^4 = \{\alpha_1\beta_{123}, \alpha_2\beta_{123}, \alpha_{12}\beta_{12}, \alpha_{12}\beta_{13}, \alpha_{12}\beta_{23}\}$$

$$F^3\Lambda^4 = \{\alpha_1\beta_{123}, \alpha_2\beta_{123}\}$$

$$F^3\Lambda^5 = \{\alpha_{12}\beta_{123}\}$$

Calculating the squares

Now we are ready to set up the squares. Using $E^{pq} = \frac{F^p\Lambda^{p+q}}{F^{p+1}\Lambda^{p+q}}$ we can figure out which forms go into which square.

$$E^{0,1} = \frac{F^0\Lambda^{0+1}}{F^{0+1}\Lambda^{0+1}} = \frac{F^0\Lambda^1}{F^1\Lambda^1} = \frac{\{\alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3\}}{\{\beta_1, \beta_2, \beta_3\}} = \{\alpha_1, \alpha_2\}$$

$$E^{0,2} = \frac{F^0\Lambda^2}{F^1\Lambda^2} = \frac{\{\alpha_{12}, \alpha_1\beta_1, \alpha_1\beta_2, \alpha_1\beta_3, \alpha_2\beta_1, \alpha_2\beta_2, \alpha_2\beta_3, \beta_{12}, \beta_{13}, \beta_{23}\}}{\{\alpha_1\beta_1, \alpha_1\beta_2, \alpha_1\beta_3, \alpha_2\beta_1, \alpha_2\beta_2, \alpha_2\beta_3, \beta_{12}, \beta_{13}, \beta_{23}\}} = \{\alpha_{12}\}$$

$$E^{1,0} = \frac{F^1\Lambda^1}{F^2\Lambda^1} = \frac{\{\beta_1, \beta_2, \beta_3\}}{\{\}} = \{\beta_1, \beta_2, \beta_3\}$$

$$E^{1,1} = \frac{F^1\Lambda^2}{F^2\Lambda^2} = \frac{\{\alpha_1\beta_1, \alpha_1\beta_2, \alpha_1\beta_3, \alpha_2\beta_1, \alpha_2\beta_2, \alpha_2\beta_3, \beta_{12}, \beta_{13}, \beta_{23}\}}{\{\beta_{12}, \beta_{13}, \beta_{23}\}} = \{\alpha_1\beta_1, \alpha_1\beta_2, \alpha_1\beta_3, \alpha_2\beta_1, \alpha_2\beta_2, \alpha_2\beta_3\}$$

$$E^{1,2} = \frac{F^1\Lambda^3}{F^2\Lambda^3} = \frac{\{\alpha_{12}\beta_1, \alpha_{12}\beta_2, \alpha_{12}\beta_3, \alpha_1\beta_{12}, \alpha_1\beta_{13}, \alpha_1\beta_{23}, \alpha_2\beta_{12}, \alpha_2\beta_{13}, \alpha_2\beta_{23}, \beta_{123}\}}{\{\alpha_1\beta_{12}, \alpha_1\beta_{13}, \alpha_1\beta_{23}, \alpha_2\beta_{12}, \alpha_2\beta_{13}, \alpha_2\beta_{23}, \beta_{123}\}} = \{\alpha_{12}\beta_1, \alpha_{12}\beta_2, \alpha_{12}\beta_3\}$$

$$E^{2,0} = \frac{F^2\Lambda^2}{F^3\Lambda^2} = \frac{\{\beta_{12}, \beta_{13}, \beta_{23}\}}{\{\}} = \{\beta_{12}, \beta_{13}, \beta_{23}\}$$

$$E^{2,1} = \frac{F^2\Lambda^3}{F^3\Lambda^3} = \frac{\{\alpha_1\beta_{12}, \alpha_1\beta_{13}, \alpha_1\beta_{23}, \alpha_2\beta_{12}, \alpha_2\beta_{13}, \alpha_2\beta_{23}, \beta_{123}\}}{\{\beta_{123}\}} = \{\alpha_1\beta_{12}, \alpha_1\beta_{13}, \alpha_1\beta_{23}, \alpha_2\beta_{12}, \alpha_2\beta_{13}, \alpha_2\beta_{23}\}$$

$$E^{2,2} = \frac{F^2 \Lambda^4}{F^3 \Lambda^4} = \frac{\{\alpha_1 \beta_{123}, \alpha_2 \beta_{123}, \alpha_{12} \beta_{12}, \alpha_{12} \beta_{13}, \alpha_{12} \beta_{23}\}}{\{\alpha_1 \beta_{123}, \alpha_2 \beta_{123}\}} = \{\alpha_{12} \beta_{12}, \alpha_{12} \beta_{13}, \alpha_{12} \beta_{23}\}$$

$$E^{3,0} = \frac{F^3 \Lambda^3}{F^4 \Lambda^3} = \frac{\{\beta_{123}\}}{\{\}} = \{\beta_{123}\}$$

$$E^{3,1} = \frac{F^3 \Lambda^4}{F^4 \Lambda^4} = \frac{\{\alpha_1 \beta_{123}, \alpha_2 \beta_{123}\}}{\{\}} = \{\alpha_1 \beta_{123}, \alpha_2 \beta_{123}\}$$

$$E^{3,2} = \frac{F^3 \Lambda^5}{F^4 \Lambda^5} = \frac{\{\alpha_{12} \beta_{123}\}}{\{\}} = \{\alpha_{12} \beta_{123}\}$$

Setting up the page

Now that we have calculated the squares we can start calculating the sequence. To make things easier to visualize we will use a grid. Each spot on the grid will be give a coordinate as if it was a typical Cartesian x-y plot. So the bottom left square would be the $(0, 0)$ position. The $E^{x,y}$ square will go into the (x, y) position. Using this idea we can set up the grid as such below. Notice that the number of α 's correspond to the y-coordinate and the number of β 's correspond the x-coordinate. This was done on purpose and will make it easier to jump straight to this step without have to figure out all the filtrations first.

α_{12}	$\alpha_{12} \beta_1$	$\alpha_{12} \beta_2$	$\alpha_{12} \beta_3$	$\alpha_{12} \beta_{12}$	$\alpha_{12} \beta_{13}$	$\alpha_{12} \beta_{23}$	$\alpha_{12} \beta_{123}$
α_1	$\alpha_1 \beta_1$	$\alpha_1 \beta_2$	$\alpha_1 \beta_3$	$\alpha_1 \beta_{12}$	$\alpha_1 \beta_{13}$	$\alpha_1 \beta_{23}$	$\alpha_1 \beta_{123}$
α_2	$\alpha_2 \beta_1$	$\alpha_2 \beta_2$	$\alpha_2 \beta_3$	$\alpha_2 \beta_{12}$	$\alpha_2 \beta_{13}$	$\alpha_2 \beta_{23}$	$\alpha_2 \beta_{123}$
	β_1	β_2	β_3	β_{12}	β_{13}	β_{23}	β_{123}

3.4.3 E_0

The next step is to calculate the first page. The first page of the spectral sequence book will always be page 0. To calculate this we will use the E_0 differential, d_0 . Now remember the general equation for the Serre Hochschild spectral sequence differential is:

$$d_r : E_r^{p,q} \rightarrow E_r^{p+r,q-r+1}$$

So that means that differential on the first page would be:

$$d_0 : E_0^{p,q} \rightarrow E_0^{p,q+1}$$

Now to calculate d_0 for each form let's first look at table of differential for each 1-form again:

$d(\alpha_1) = -\alpha_1\beta_3 + \alpha_2\beta_1 - \beta_2\beta_3$	$d(\beta_1) = 0$
$d(\alpha_2) = -\alpha_2\beta_3$	$d(\beta_2) = -\alpha_2\beta_3 - \beta_2\beta_3$
	$d(\beta_3) = 0$

Now let's use the above table to calculate the differential for all the forms.

$d(\alpha_{12}) = 2\alpha_{12}\beta_3 - \alpha_2\beta_{23}$	$d(\alpha_{12}\beta_1) = -2\alpha_{12}\beta_{13} - \alpha_2\beta_{123}$ $d(\alpha_{12}\beta_2) = -3\alpha_{12}\beta_{23}$ $d(\alpha_{12}\beta_3) = 0$	$d(\alpha_{12}\beta_{12}) = 3\alpha_{12}\beta_{123}$ $d(\alpha_{12}\beta_{13}) = 0$ $d(\alpha_{12}\beta_{23}) = 0$	$d(\alpha_{12}\beta_{123}) = 0$
$d(\alpha_1) = -\alpha_1\beta_3 + \alpha_2\beta_1 - \beta_2\beta_3$ $d(\alpha_2) = -\alpha_2\beta_3$	$d(\alpha_1\beta_1) = \alpha_1\beta_{13} - \beta_{123}$ $d(\alpha_1\beta_2) = \alpha_{12}\beta_3 + 2\alpha_1\beta_{23} + \alpha_2\beta_{12}$ $d(\alpha_1\beta_3) = \alpha_2\beta_{13}$ $d(\alpha_2\beta_1) = \alpha_2\beta_{13}$ $d(\alpha_2\beta_2) = 2\alpha_2\beta_{23}$ $d(\alpha_2\beta_3) = 0$	$d(\alpha_1\beta_{12}) = \alpha_{12}\beta_{13} - 2\alpha_1\beta_{123}$ $d(\alpha_1\beta_{13}) = 0$ $d(\alpha_1\beta_{23}) = \alpha_2\beta_{123}$ $d(\alpha_2\beta_{12}) = -2\alpha_2\beta_{123}$ $d(\alpha_2\beta_{13}) = 0$ $d(\alpha_2\beta_{23}) = 0$	$d(\alpha_1\beta_{123}) = 0$ $d(\alpha_2\beta_{123}) = 0$
	$d(\beta_1) = 0$ $d(\beta_2) = -\alpha_2\beta_3 - \beta_2\beta_3$ $d(\beta_3) = 0$	$d(\beta_{12}) = -\alpha_2\beta_{13} + \beta_{123}$ $d(\beta_{13}) = 0$ $d(\beta_{23}) = 0$	$d(\beta_{123}) = 0$

From the above table we can calculate d_0 . Because $d_0 : E_0^{p,q} \rightarrow E_0^{p,q+1}$ and $d_o = \Pi^{p,q+1} \circ d$, we can see that d_0 will be part of the d that has been truncated to only include the portion in the square that the differential is being projected onto. Now $d_0(x)$ projects onto the square above x 's square so it will only include forms that are in the square above. For example $\alpha_1\beta_2$ is in the $E^{1,1}$ square so $d_0(\alpha_1\beta_2)$ would include only forms that are in the $E^{1,2}$ square. Now since $d(\alpha_1\beta_2) = \alpha_{12}\beta_3 + 2\alpha_1\beta_{23} + \alpha_2\beta_{12}$ so $d_0(\alpha_1\beta_2) = \alpha_{12}\beta_3$. α_2 is in $E^{0,1}$ so d_0 would project it to $E^{0,2}$. Since $d(\alpha_2) = -\alpha_2\beta_3$ that means $d_0(\alpha_2) = 0$. Below is the table for all of d_0 . Only the nonzero d_0 is listed.

	$d_0(\alpha_1\beta_2) = \alpha_{12}\beta_3$	$d_0(\alpha_1\beta_{12}) = \alpha_{12}\beta_{13}$	
	$d_0(\beta_2) = -\alpha_2\beta_3$	$d_0(\beta_{12}) = -\alpha_2\beta_{13}$	

If a form x has a derivative y , on that page, both the form and it's derivative are paired up and cancel each other out. x has "killed" y . So in this example $d_0(\beta_2) = -\alpha_2\beta_3$ which means β_2 and $-\alpha_2\beta_3$ are paired up and β_2 killed $\alpha_2\beta_3$. This relationship will be denoted by an arrow on the grid. This processes is eliminating the exact forms and the complementary space from the sequence. Below is the complete page 0 including the paired forms that cancel each other out. These canceled forms will not be on the next page.

 E_0 Page

α_{12}	$\alpha_{12}\beta_1$	$\alpha_{12}\beta_2$	$\alpha_{12}\beta_3$	$\alpha_{12}\beta_{12}$	$\alpha_{12}\beta_{13}$	$\alpha_{12}\beta_{23}$	$\alpha_{12}\beta_{123}$
α_1	$\alpha_1\beta_1$	$\alpha_1\beta_2$	$\alpha_1\beta_3$	$\alpha_1\beta_{12}$	$\alpha_1\beta_{13}$	$\alpha_1\beta_{23}$	$\alpha_1\beta_{123}$
α_2	$\alpha_2\beta_1$	$\alpha_2\beta_2$	$\alpha_2\beta_3$	$\alpha_2\beta_{12}$	$\alpha_2\beta_{13}$	$\alpha_2\beta_{23}$	$\alpha_2\beta_{123}$
	β_1	β_2	β_3	β_{12}	β_{13}	β_{23}	β_{123}

 d_0 table

	$d_0(\alpha_1\beta_2) = -\alpha_{12}\beta_3$	$d_0(\alpha_1\beta_{12}) = \alpha_{12}\beta_{13}$	
	$d_0(\beta_2) = -\alpha_2\beta_3$	$d_0(\beta_{12}) = -\alpha_2\beta_{13}$	

3.4.4 E_1

The next step in the sequence is calculating the E_1 page. Forms that were paired up on the last page are no longer part of the sequence. On the grid we will denote forms that have already been paired up as a circle. So our starting point for E_1 looks like the following:

α_{12}	$\alpha_{12}\beta_1$	$\alpha_{12}\beta_2$	\circ	$\alpha_{12}\beta_{12}$	\circ	$\alpha_{12}\beta_{23}$	$\alpha_{12}\beta_{123}$
α_1	$\alpha_1\beta_1$	\circ	$\alpha_1\beta_3$	\circ	$\alpha_1\beta_{13}$	$\alpha_1\beta_{23}$	$\alpha_1\beta_{123}$
α_2	$\alpha_2\beta_1$	$\alpha_2\beta_2$	\circ	$\alpha_2\beta_{12}$	\circ	$\alpha_2\beta_{23}$	$\alpha_2\beta_{123}$
	β_1	\circ	β_3	\circ	β_{13}	β_{23}	β_{123}

Next step is to calculate d_1 . Using the formula:

$$d_r : E_r^{p,q} \rightarrow E_r^{p+r,q-r+1}$$

would mean that

$$d_1 : E_1^{p,q} \rightarrow E_1^{p+1,q}$$

So we are look at the projection of d into the square to the right this time. Now let's look at the differential table for the remaining forms to calculate d_1 .

$d(\alpha_{12}) = 2\alpha_{12}\beta_3 - \alpha_2\beta_{23}$	$d(\alpha_{12}\beta_1) = -2\alpha_{12}\beta_{13} - \alpha_2\beta_{123}$ $d(\alpha_{12}\beta_2) = -3\alpha_{12}\beta_{23}$	$d(\alpha_{12}\beta_{12}) = 3\alpha_{12}\beta_{123}$ $d(\alpha_{12}\beta_{23}) = 0$	$d(\alpha_{12}\beta_{123}) = 0$
$d(\alpha_1) = -\alpha_1\beta_3 + \alpha_2\beta_1 - \beta_2\beta_3$ $d(\alpha_2) = -\alpha_2\beta_3$	$d(\alpha_1\beta_1) = \alpha_1\beta_{13} - \beta_{123}$ $d(\alpha_1\beta_3) = \alpha_2\beta_{13}$ $d(\alpha_2\beta_1) = \alpha_2\beta_{13}$ $d(\alpha_2\beta_2) = 2\alpha_2\beta_{23}$	$d(\alpha_1\beta_{13}) = 0$ $d(\alpha_1\beta_{23}) = \alpha_2\beta_{123}$ $d(\alpha_2\beta_{12}) = -2\alpha_2\beta_{123}$ $d(\alpha_2\beta_{23}) = 0$	$d(\alpha_1\beta_{123}) = 0$ $d(\alpha_2\beta_{123}) = 0$
	$d(\beta_1) = 0$ $d(\beta_3) = 0$	$d(\beta_{13}) = 0$ $d(\beta_{23}) = 0$	$d(\beta_{123}) = 0$

So taking the differentials from the above table and taking only the parts that project in to the square to right we get this table:

$\hat{d}(\alpha_{12}) = 2\alpha_{12}\beta_3$	$\hat{d}(\alpha_{12}\beta_1) = -2\alpha_{12}\beta_{13}$ $\hat{d}(\alpha_{12}\beta_2) = -3\alpha_{12}\beta_{23}$	$\hat{d}(\alpha_{12}\beta_{12}) = 2\alpha_{12}\beta_{123}$	
$\hat{d}(\alpha_1) = -\alpha_1\beta_3 + \alpha_2\beta_1$ $\hat{d}(\alpha_2) = -\alpha_2\beta_3$	$\hat{d}(\alpha_1\beta_1) = \alpha_1\beta_{13}$ $\hat{d}(\alpha_2\beta_2) = \alpha_2\beta_{23}$ $\hat{d}(\alpha_1\beta_3) = 2\alpha_2\beta_{13}$ $\hat{d}(\alpha_2\beta_1) = 2\alpha_2\beta_{13}$	$\hat{d}(\alpha_2\beta_{12}) = \alpha_2\beta_{123}$ $\hat{d}(\alpha_1\beta_{23}) = -2\alpha_2\beta_{123}$	

Now there are a couple of issues with the above table that need to be addressed before we can finish calculating the page. The first issue we have is that several forms are projecting onto forms that are no longer on the page. For instance

$$\hat{d}(\alpha_{12}) = 2\alpha_{12}\beta_3$$

Now if we recall

$$\alpha_{12}\beta_3 = d_0(\alpha_1\beta_2)$$

So

$$\hat{d}(\alpha_{12}) = 2\alpha_{12}\beta_3 = 2 * d_0(\alpha_1\beta_2)$$

Whenever a $d_r(x)$ hits something that is no longer on the page it forms a chain. So in this example α_{12} is going to form a chain with $\alpha_1\beta_2$. Since

$$\hat{d}(\alpha_{12}) = 2\alpha_{12}\beta_3 = 2 * d_0(\alpha_1\beta_2)$$

we will chain $\alpha_1\beta_2$ to α_{12} so that the derivative is zero. We will signify a chain in our charts with a \square .

$$[\alpha_{12}]_1 = \alpha_{12} - 2\alpha_{12}\beta_3$$

$$\hat{d}([\alpha_{12}]_1) = 0.$$

This also means from now on instead of α_{12} we now have a chain $[\alpha_{12}]_1$. When you take the derivative of the form you are taking the derivative of the whole chain. So it is essentially

$$d([\alpha_{12}]_1) = d(\alpha_{12} - 2\alpha_1\beta_{12})$$

The next two issues are different but require the same solution. We have forms that's differential is going into the proper square but the form isn't explicitly in our grid. An example of this is

$$\hat{d}(\alpha_1) = -\alpha_1\beta_3 + \alpha_2\beta_1$$

The other issue is we have forms whose differential are the same. Such as

$$\hat{d}(\alpha_1\beta_3) = \alpha_2\beta_{13}$$

$$\hat{d}(\alpha_2\beta_1) = \alpha_2\beta_{13}$$

These problems can be easily rectified if you recall that each square is a subspace. The forms displayed in each square just form a basis, but the basis is not unique. Essentially what is happening is we are calculating the cohomology from square a into square b. So what we need to do is calculate a basis for the complementary space, exact space, and cohomology space in each square with respect to that page's differential. In more complicated examples we would use an algorithm similar to the one described by the cohomology example to find the basis of those three spaces. This problem is simple enough that we can do it by inspection. To fix both problems we will just change the basis in the $E^{1,1}$ square. Let

$$\gamma_1 = -\alpha_1\beta_3 + \alpha_2\beta_1$$

$$\gamma_2 = \alpha_1\beta_3 + \alpha_2\beta_1$$

Now

$$\hat{d}(\alpha_1) = \gamma_1$$

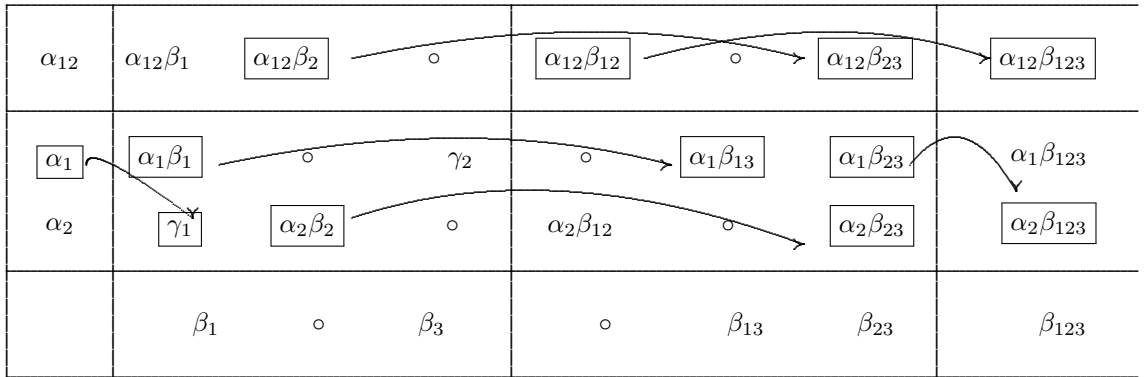
$$\hat{d}(\gamma_1) = 0$$

$$\hat{d}(\gamma_2) = 2\alpha_2\beta_{13} = -2 * d_0(\alpha_2\beta_{13}) = 0$$

Now with these in mind we can finish the d_1 table:

$d_1(\alpha_{12}) = 2\alpha_{12}\beta_3 = 2d_0(\alpha_1\beta_2) = 0$	$d_1(\alpha_{12}\beta_1) = -2\alpha_{12}\beta_{13} = 2d_0(\alpha_1\beta_{12}) = 0$ $d_1(\alpha_{12}\beta_2) = -3\alpha_{12}\beta_{23}$	$d_1(\alpha_{12}\beta_{12}) = 2\alpha_{12}\beta_{123}$	
$d_1(\alpha_1) = \gamma_1$ $d_1(\alpha_2) = -\alpha_2\beta_3 = d_0(\beta_2) = 0$	$d_1(\alpha_1\beta_1) = \alpha_1\beta_{13}$ $d_1(\alpha_2\beta_2) = \alpha_2\beta_{23}$ $d_1(\gamma_2) = 2\alpha_2\beta_{13} = -2d_0(\alpha_2\beta_{13}) = 0$	$d_1(\alpha_2\beta_{12}) = \alpha_2\beta_{123}$ $d_1(\alpha_1\beta_{23}) = -2\alpha_2\beta_{123}$	

So now we are ready to calculate the page.



d_1 table

$d_1(\alpha_{12}) = 2\alpha_{12}\beta_3 = 2d_0(\alpha_1\beta_2) = 0$	$d_1(\alpha_{12}\beta_1) = -2\alpha_{12}\beta_{13} = 2d_0(\alpha_1\beta_{12}) = 0$ $d_1(\alpha_{12}\beta_2) = -3\alpha_{12}\beta_{23}$	$d_1(\alpha_{12}\beta_{12}) = 2\alpha_{12}\beta_{123}$	
$d_1(\alpha_1) = \gamma_1$ $d_1(\alpha_2) = -\alpha_2\beta_3 = d_0(\beta_2) = 0$	$d_1(\alpha_1\beta_1) = \alpha_1\beta_{13}$ $d_1(\alpha_2\beta_2) = \alpha_2\beta_{23}$ $d_1(\gamma_2) = 2\alpha_2\beta_{13} = -2d_0(\alpha_2\beta_{13}) = 0$	$d_1(\alpha_2\beta_{12}) = \alpha_2\beta_{123}$ $d_1(\alpha_1\beta_{23}) = -2\alpha_2\beta_{123}$	

3.4.5 E_2

Now before we start calculating page 2 note that several chains were created in E_1 . We need to keep track of these chains. If we forget that these forms are chains we risk not getting the complete answer.

Chain table

$[\alpha_2]_1 = \alpha_2 - \beta_2$	$[\alpha_{12}]_1 = \alpha_{12} - 2\alpha_1\beta_2$
$[\alpha_{12}\beta_1]_1 = \alpha_{12}\beta_1 + 2\alpha_1\beta_{12}$	$[\gamma_2]_1 = \alpha_1\beta_3 + \alpha_2\beta_1 + 2\beta_{12}$

Now lets look at what forms are left in the sequence at this point:

$[\alpha_{12}]_1$	$[\alpha_{12}\beta_1]_1$	○	○				
○	○	○	$[\gamma_2]_1$	○	○	○	$\alpha_1 \beta_{123}$
$[\alpha_2]_1$	○	○	○	$\alpha_2 \beta_{12}$	○	○	
	β_1	○	β_3		β_{13}	β_{23}	β_{123}

Now lets calculate the E_2 differential. First lets calculate the differential of all the remaining forms and chains.

$d([\alpha_{12}]_1) = -4\alpha_1\beta_{23} - 2\alpha_2\beta_{12} - \alpha_2\beta_{23}$	$d([\alpha_{12}\beta_1]_1) = -4\alpha_1\beta_{123} - \alpha_2\beta_{123}$		
$d([\alpha_2]_1) = \beta_{23}$	$d([\gamma_2]_1) = 2\beta_{123}$	$d(\alpha_2\beta_{12}) = -2\alpha_2\beta_{123}$	$d(\alpha_1\beta_{123}) = 0$
	$d(\beta_1) = 0$ $d(\beta_3) = 0$	$d(\beta_{13}) = 0$ $d(\beta_{23}) = 0$	$d(\beta_{123}) = 0$

Now we can use this to calculate d_2 .

$$d_r : E_r^{p,q} \rightarrow E_r^{p+r,q-r+1}$$

would mean that

$$d_2 : E_2^{p,q} \rightarrow E_2^{p+2,q-1}$$

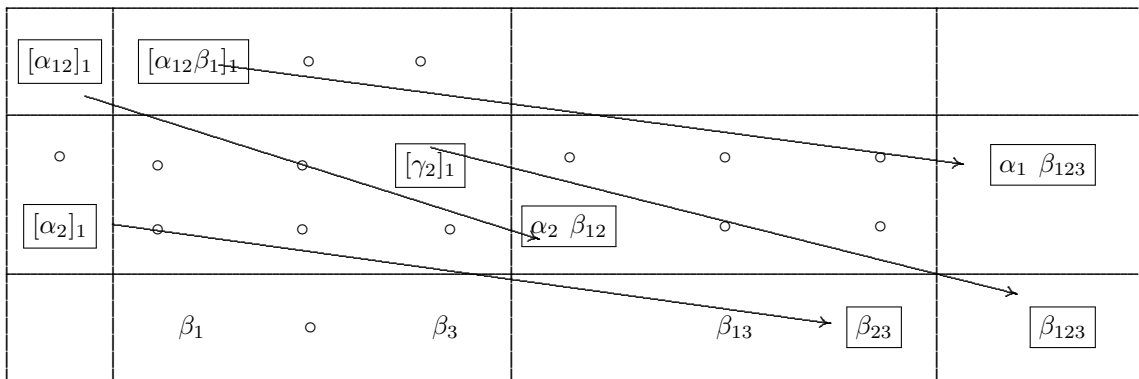
So d_2 is projecting to a square one down and two to the right of the original square. Using this knowledge we can calculate d_2 .

d_2 table

$d_2([\alpha_{12}]_1) = -4\alpha_1\beta_{23} - 2\alpha_2\beta_{12} - \alpha_2\beta_{23}$	$d_2([\alpha_{12}\beta_1]_1) = -4\alpha_1\beta_{123} - \alpha_2\beta_{123}$		
$d_2([\alpha_2]_1) = \beta_{23}$	$d_2([\gamma_2]_1) = 2\beta_{123}$		

Notice that the above table is creating more chains.

E_2 Page



$[\alpha_2]_1 = \alpha_2 - \beta_2$	$[\alpha_{12}]_1 = \alpha_{12} - 2\alpha_1\beta_2$
$[\alpha_{12}\beta_1]_1 = \alpha_{12}\beta_1 + 2\alpha_1\beta_{12}$	$[\gamma_2]_1 = \alpha_1\beta_3 + \alpha_2\beta_1 + 2\beta_{12}$

3.4.6 E_3

Now we are ready to calculate the E_3 page. This page only has three forms left. Let's calculate the differential of each of the three remaining forms.

	$d(\beta_1) = 0$	$d(\beta_{13}) = 0$	
	$d(\beta_3) = 0$		

Now since each of these forms have a differential of zero their E_3 page differential will also be the same. So the page and differential table will be as follows:

E₃ Page

	β_1	β_3	β_{13}

d₃ table

	$d_3(\beta_1) = 0$	$d_3(\beta_{13}) = 0$	
	$d_3(\beta_3) = 0$		

Now the table has converged. Because all of the remaining forms have a differential of zero we know that each consequent page's differential will be the same. So the remaining forms are the cohomology of this lie algebra.

3.4.7 Cohomology

β_1, β_3	β_{13}		
--------------------	--------------	--	--

3.5 Example 2

This next example will show that sometimes chains end up as part of the cohomology.

3.5.1 Lie algebra

We will let the following tables define our Lie algebra. We will let x denote our sub-algebra and y denote our complementary space. α is the dual to the sub-algebra β is the dual to the complementary space.

Multiplication table

	x_1	x_2	y_1	y_2	y_3
x_1	0	0	0	0	0
x_2	0	0	0	x_1	y_1
y_1	0	0	0	0	x_1
y_2	0	$-x_1$	0	0	x_2
y_3	0	$-y_1$	$-x_1$	$-x_2$	0

Structure Equations

$d(\alpha_1) = -\alpha_2\beta_2 - \beta_{13}$	$d(\beta_1) = -\alpha_2\beta_3$
$d(\alpha_2) = -\beta_{23}$	$d(\beta_2) = 0$
	$d(\beta_3) = 0$

3.5.2 E_0

Now lets set up the first page. This time we are going to skip the filtration. When you go through all of those steps you will always end up with the same type of construction. The x coordinate for the square will tell you how many β 's, complementary space dual, are in each form in the square and the y coordinate will tell you how many α 's, sub-algebra dual, are in each form in the square. The bottom left hand square is always $(0,0)$. So the setup for the 0 page is:

α_{12}	$\alpha_{12}\beta_1$	$\alpha_{12}\beta_2$	$\alpha_{12}\beta_3$	$\alpha_{12}\beta_{12}$	$\alpha_{12}\beta_{13}$	$\alpha_{12}\beta_{23}$	$\alpha_{12}\beta_{123}$
α_1	$\alpha_1\beta_1$	$\alpha_1\beta_2$	$\alpha_1\beta_3$	$\alpha_1\beta_{12}$	$\alpha_1\beta_{13}$	$\alpha_1\beta_{23}$	$\alpha_1\beta_{123}$
α_2	$\alpha_2\beta_1$	$\alpha_2\beta_2$	$\alpha_2\beta_3$	$\alpha_2\beta_{12}$	$\alpha_2\beta_{13}$	$\alpha_2\beta_{23}$	$\alpha_2\beta_{123}$
	β_1	β_2	β_3	β_{12}	β_{13}	β_{23}	β_{123}

By using the structure equations we can calculate the differential table for all forms.

Differential table

$d(\alpha_{12}) = \alpha_1\beta_{23} - \alpha_2\beta_{13}$	$d(\alpha_{12}\beta_1) = \alpha_1\beta_{123}$ $d(\alpha_{12}\beta_2) = \alpha_2\beta_{123}$ $d(\alpha_{12}\beta_3) = 0$	$d(\alpha_{12}\beta_{12}) = 0$ $d(\alpha_{12}\beta_{13}) = 0$ $d(\alpha_{12}\beta_{23}) = 0$	$d(\alpha_{12}\beta_{123}) = 0$
$d(\alpha_1) = -\alpha_2\beta_2 - \beta_{13}$ $d(\alpha_2) = -\beta_{23}$	$d(\alpha_1\beta_1) = \alpha_{12}\beta_3 + \alpha_2\beta_{12}$ $d(\alpha_1\beta_2) = \beta_{123}$ $d(\alpha_1\beta_3) = -\alpha_2\beta_{23}$ $d(\alpha_2\beta_1) = -\beta_{123}$ $d(\alpha_2\beta_2) = 0$ $d(\alpha_2\beta_3) = 0$	$d(\alpha_1\beta_{12}) = -\alpha_{12}\beta_{23}$ $d(\alpha_1\beta_{13}) = \alpha_2\beta_{123}$ $d(\alpha_1\beta_{23}) = 0$ $d(\alpha_2\beta_{12}) = 0$ $d(\alpha_2\beta_{13}) = 0$ $d(\alpha_2\beta_{23}) = 0$	$d(\alpha_1\beta_{123}) = 0$ $d(\alpha_2\beta_{123}) = 0$
	$d(\beta_1) = -\alpha_2\beta_3$ $d(\beta_2) = 0$ $d(\beta_3) = 0$	$d(\beta_{12}) = \alpha_2\beta_{23}$ $d(\beta_{13}) = 0$ $d(\beta_{23}) = 0$	$d(\beta_{123}) = 0$

From this we can calculate d_0 . d_0 is the part of the differential that is projected into the square above. Now using the same process as the previous example we get:

d_0 table

$d_0(\alpha_1\beta_1) = \alpha_{12}\beta_3$	$d_0(\alpha_1\beta_{12}) = -\alpha_{12}\beta_{23}$	
$d_0(\beta_1) = -\alpha_2\beta_3$	$d_0(\beta_{12}) = \alpha_2\beta_{23}$	

From this we can now calculate the E_0 page.

E_0 Page

α_{12}	$\alpha_{12}\beta_1$ $\alpha_{12}\beta_2$ $\boxed{\alpha_{12}\beta_3}$	$\alpha_{12}\beta_{12}$ $\alpha_{12}\beta_{13}$ $\boxed{\alpha_{12}\beta_{23}}$	$\alpha_{12}\beta_{123}$
α_1	$\boxed{\alpha_1\beta_1}$ $\alpha_1\beta_2$ $\alpha_1\beta_3$	$\boxed{\alpha_1\beta_{12}}$ $\alpha_1\beta_{13}$ $\alpha_1\beta_{23}$	$\alpha_1\beta_{123}$
α_2	$\alpha_2\beta_1$ $\alpha_2\beta_2$ $\boxed{\alpha_2\beta_3}$	$\alpha_2\beta_{12}$ $\alpha_2\beta_{13}$ $\boxed{\alpha_2\beta_{23}}$	$\alpha_2\beta_{123}$
	$\boxed{\beta_1}$ β_2 β_3	$\boxed{\beta_{12}}$ β_{13} β_{23}	β_{123}

3.5.3 E_1

Now we are going to calculate the E_1 page. This is where our first chains will start appearing. First lets take stock on what forms are remaining:

α_{12}	$\alpha_{12}\beta_1$	$\alpha_{12}\beta_2$	\circ	$\alpha_{12}\beta_{12}$	$\alpha_{12}\beta_{13}$	\circ	$\alpha_{12}\beta_{123}$
α_1	\circ	$\alpha_1\beta_2$	$\alpha_1\beta_3$	\circ	$\alpha_1\beta_{13}$	$\alpha_1\beta_{23}$	$\alpha_1\beta_{123}$
α_2	$\alpha_2\beta_1$	$\alpha_2\beta_2$	\circ	$\alpha_2\beta_{12}$	$\alpha_2\beta_{13}$	\circ	$\alpha_2\beta_{123}$
	\circ	β_2	β_3	\circ	β_{13}	β_{23}	β_{123}

Now let's calculate the d_1 differential. The d_1 differential is the part of the differential that is projected into the square to the right. So we'll take the differentials from the table below and calculate d_1 .

$d(\alpha_{12}) = \alpha_1\beta_{23} - \alpha_2\beta_{13}$	$d(\alpha_{12}\beta_1) = \alpha_1\beta_{123}$ $d(\alpha_{12}\beta_2) = \alpha_2\beta_{123}$	$d(\alpha_{12}\beta_{12}) = 0$ $d(\alpha_{12}\beta_{13}) = 0$	$d(\alpha_{12}\beta_{123}) = 0$
$d(\alpha_1) = -\alpha_2\beta_2 - \beta_{13}$ $d(\alpha_2) = -\beta_{23}$	$d(\alpha_1\beta_2) = \beta_{123}$ $d(\alpha_1\beta_3) = -\alpha_2\beta_{23}$ $d(\alpha_2\beta_1) = -\beta_{123}$ $d(\alpha_2\beta_2) = 0$	$d(\alpha_1\beta_{13}) = \alpha_2\beta_{123}$ $d(\alpha_1\beta_{23}) = 0$ $d(\alpha_2\beta_{12}) = 0$ $d(\alpha_2\beta_{13}) = 0$	$d(\alpha_1\beta_{123}) = 0$ $d(\alpha_2\beta_{123}) = 0$
	$d(\beta_2) = 0$ $d(\beta_3) = 0$	$d(\beta_{13}) = 0$ $d(\beta_{23}) = 0$	$d(\beta_{123}) = 0$

The d_1 is as follows. If you notice we are creating one chain on this page. $\alpha_1\beta_3$ is creating a chain with β_{12} .

$d_1(\alpha_1) = -\alpha_2\beta_2$	$d_1(\alpha_1\beta_{13}) = \alpha_2\beta_{123}$	$d_1(\alpha_1\beta_3) = -\alpha_2\beta_{23} = -d_0(\beta_{12})$	

Now we can calculate the E_1 page.

E_1 Page

α_{12}	$\alpha_{12}\beta_1$	$\alpha_{12}\beta_2$	\circ	$\alpha_{12}\beta_{12}$	$\alpha_{12}\beta_{13}$	\circ	$\alpha_{12}\beta_{123}$
α_1	\circ	$\alpha_1\beta_2$	$\alpha_1\beta_3$	\circ	$\alpha_1\beta_{13}$	$\alpha_1\beta_{23}$	$\alpha_1\beta_{123}$
α_2	$\alpha_2\beta_1$	$\alpha_2\beta_2$	\circ	$\alpha_2\beta_{12}$	$\alpha_2\beta_{13}$	\circ	$\alpha_2\beta_{123}$
	\circ	β_2	β_3	\circ	β_{13}	β_{23}	β_{123}

 d_1 table

$d_1(\alpha_1) = -\alpha_2\beta_2$	$d_1(\alpha_1\beta_{13}) = \alpha_2\beta_{123}$	$d_1(\alpha_1\beta_3) = -\alpha_2\beta_{23} = -d_0(\beta_{12})$	

3.5.4 E_2

Now to calculate the E_2 page. Let's look at what forms are left:

α_{12}	$\alpha_{12}\beta_1$	$\alpha_{12}\beta_2$	\circ	$\alpha_{12}\beta_{12}$	$\alpha_{12}\beta_{13}$	\circ	$\alpha_{12}\beta_{123}$
\circ	\circ	$\alpha_1\beta_2$	$[\alpha_1\beta_3]_1$	\circ	\circ	$\alpha_1\beta_{23}$	$\alpha_1\beta_{123}$
α_2	$\alpha_2\beta_1$	\circ	\circ	$\alpha_2\beta_{12}$	$\alpha_2\beta_{13}$	\circ	\circ
	\circ	β_2	β_3	\circ	β_{13}	β_{23}	β_{123}

Now let's calculate the differential of the remaining forms and chains.

Differential table

$d(\alpha_{12}) = \alpha_1\beta_{23} - \alpha_2\beta_{13}$	$d(\alpha_{12}\beta_1) = \alpha_1\beta_{123}$ $d(\alpha_{12}\beta_2) = \alpha_2\beta_{123}$	$d(\alpha_{12}\beta_{12}) = 0$ $d(\alpha_{12}\beta_{13}) = 0$	$d(\alpha_{12}\beta_{123}) = 0$
$d(\alpha_2) = -\beta_{23}$	$d(\alpha_1\beta_2) = \beta_{123}$ $d([\alpha_1\beta_3]_1) = 0$ $d(\alpha_2\beta_1) = -\beta_{123}$	$d(\alpha_1\beta_{23}) = 0$ $d(\alpha_2\beta_{12}) = 0$ $d(\alpha_2\beta_{13}) = 0$	$d(\alpha_1\beta_{123}) = 0$
	$d(\beta_2) = 0$ $d(\beta_3) = 0$	$d(\beta_{13}) = 0$ $d(\beta_{23}) = 0$	$d(\beta_{123}) = 0$

Chain table

$$\boxed{[\alpha_1\beta_3]_1 = \alpha_1\beta_3 + \beta_{12}}$$

From here we can calculate d_2 .

$d_2(\alpha_{12}) = \alpha_1\beta_{23} - \alpha_2\beta_{13}$	$d_2(\alpha_{12}\beta_1) = \alpha_1\beta_{123}$ $d_2(\alpha_{12}\beta_2) = \alpha_2\beta_{123}$		
$d_2(\alpha_2) = -\beta_{23}$	$d_2(\alpha_1\beta_2) = \beta_{123}$ $d_2(\alpha_2\beta_1) = -\beta_{123}$		

From this table it is clear that we need to change the basis. $\alpha_1\beta_2$ and $\alpha_2\beta_1$ are projecting onto the same form. To fix this we need to change the basis in $E_2^{1,1}$. Furthermore $d_2(\alpha_{12}) = \alpha_1\beta_{23} - \alpha_2\beta_{13}$ and so we need to change the basis in square $E_2^{2,1}$.

First lets deal with α_{12} and the $E_2^{2,1}$ square. If we notice both $\alpha_1\beta_{23}$ and $\alpha_2\beta_{13}$ are both in the square. So we can combine them to create a new basis. Let

$$\gamma_1 = \alpha_1\beta_{23} - \alpha_2\beta_{13}$$

and

$$\gamma_2 = \alpha_1\beta_{23} + \alpha_2\beta_{13}$$

So now

$$d_2(\alpha_{12}) = \gamma_1$$

Next lets deal with two forms projecting onto the same form. Let

$$\gamma_3 = \alpha_1\beta_2 - \alpha_2\beta_1$$

and

$$\gamma_4 = \alpha_1\beta_2 + \alpha_2\beta_1$$

Now

$$d_2(\gamma_3) = 2\beta_{123}$$

$$d_2(\gamma_4) = 0$$

The d_2 table now looks like the following:

d_2 table

$d_2(\alpha_{12}) = \gamma_1$	$d_2(\alpha_{12}\beta_1) = \alpha_1\beta_{123}$		
$d_2(\alpha_2) = -\beta_{23}$	$d_2(\alpha_{12}\beta_2) = \alpha_2\beta_{123} = d_1(\alpha_1\beta_{13})$		
	$d_2(\gamma_3) = 2\beta_{123}$		

E_2 Page

α_{12}	$\alpha_{12}\beta_1$	$\alpha_{12}\beta_2$	\circ	$\alpha_{12}\beta_{12}$	$\alpha_{12}\beta_{13}$	\circ	$\alpha_{12}\beta_{123}$
\circ	\circ	γ_3	$[\alpha_1\beta_3]_1$	\circ	\circ	γ_2	$\alpha_1\beta_{123}$
α_2	γ_4	\circ	\circ	$\alpha_2\beta_{12}$	γ_1	\circ	\circ
	\circ	β_2	β_3	\circ	β_{13}	β_{23}	β_{123}

Chain and change of basis table

$[\alpha_1\beta_3]_1 = \alpha_1\beta_3 + \beta_{12}$
$\gamma_1 = \alpha_1\beta_{23} - \alpha_2\beta_{13}$
$\gamma_2 = \alpha_1\beta_{23} + \alpha_2\beta_{13}$
$\gamma_3 = \alpha_1\beta_2 - \alpha_2\beta_1$
$\gamma_4 = \alpha_1\beta_2 + \alpha_2\beta_1$

3.5.5 E_3

This will be the last page. The only forms that will have a differential still on the page are in square $E^{0,2}$. This square is empty so all the other derivatives will be zero. This comes directly from the formula

$$d_3 : E_3^{p,q} \rightarrow E_0^{p+3,q-3+1}$$

Now the q values range from 0 to 2. The only non negative y coordinate possible is when $q=2$. Now the p values range from 0 to 3. The x coordinates range from 0 to 3. The only possible p value that would give an x coordinate less than or equal 3 is $p=0$. So only $E_3^{0,2}$ has differentials on

the page. The rest of the squares do not. It is clear to see any subsequent pages will not have any differentials on the page and so all the differentials will be zero. So the final page is as follows:

E_3 page

\circ	\circ	$[\alpha_{12}\beta_2]_2$	\circ	$\alpha_{12}\beta_{12}$	$\alpha_{12}\beta_{13}$	\circ	$\alpha_{12}\beta_{123}$
\circ	\circ	\circ	$[\alpha_1\beta_3]_1$	\circ	\circ	γ_2	\circ
\circ	γ_4	\circ	\circ	$\alpha_2\beta_{12}$	\circ	\circ	\circ
	\circ	β_2	β_3	\circ	β_{13}	\circ	\circ

Chain and change of basis table

$[\alpha_1\beta_3]_1 = \alpha_1\beta_3 + \beta_{12}$
$[\alpha_{12}\beta_2]_2 = \alpha_{12}\beta_2 - \alpha_1\beta_{13}$
$\gamma_2 = \alpha_1\beta_{23} + \alpha_2\beta_{13}$
$\gamma_4 = \alpha_1\beta_2 + \alpha_2\beta_1$

Cohomology

β_2, β_3	$\alpha_1\beta_2 + \alpha_2\beta_1, \alpha_1\beta_3 + \beta_{12}, \beta_{13}$	$\alpha_{12}\beta_2 - \alpha_1\beta_{13}, \alpha_2\beta_{12}, \alpha_1\beta_{23} + \alpha_2\beta_{13}$	$\alpha_{12}\beta_{12}, \alpha_{12}\beta_{13}$	$\alpha_{12}\beta_{123}$
--------------------	---	--	--	--------------------------

3.6 Example 3

In this example we will look at a Lie algebra with coefficients in a representation.

3.6.1 Lie algebra

Let V be a vector space with basis $\{x_1, x_2\}$. Let g be a Lie algebra with coefficients in the representation V . Let $\{z_1, z_2\}$ denote the sub-algebra in g that will be used to make the sequence. $\{y_1\}$ will be the complementary space. α will be the dual of the sub-algebra and β will be the dual of the complement. The Lie algebra will be defined by the following multiplication table and structure equations.

Multiplication Table

	z_1	z_2	y_1
z_1	0	z_2	0
z_2	$-z_2$	0	z_2
y_1	0	$-z_2$	0

Structure Equations

$d(x_1) = x_1\alpha_1$	$d(x_2) = x_1\alpha_2 + x_2\beta_1$
$d(\alpha_1) = 0$	$d(\alpha_2) = -\alpha_{12} - \alpha_2\beta_1$
$d(\beta_1) = 0$	

3.6.2 E_0

Setting up the first page is the same as the previous examples. The only difference is in the $E^{0,0}$ square. That square is filled with the coefficients. In this case, x_1, x_2 .

$x_1\alpha_{12}$	$x_2\alpha_{12}$	$x_1\alpha_{12}\beta_1$	$x_2\alpha_{12}\beta_1$
$x_1\alpha_2$	$x_2\alpha_2$	$x_1\alpha_2\beta_1$	$x_2\alpha_2\beta_2$
$x_1\alpha_1$	$x_2\alpha_1$	$x_1\alpha_1\beta_1$	$x_2\alpha_1\beta_2$
x_1	x_2	$x_1\beta_1$	$x_2\beta_1$

Next let's calculate the differential table for all of the forms.

Differential Table

$d(x_1\alpha_{12}) = x_1\alpha_{12}\beta_1$ $d(x_2\alpha_{12}) = 2x_2\alpha_{12}\beta_1$	$d(x_1\alpha_{12}\beta_1) = 0$ $d(x_2\alpha_{12}\beta_1) = 0$
$d(x_1\alpha_1) = 0$ $d(x_1\alpha_2) = -x_1\alpha_2\beta_1$ $d(x_2\alpha_1) = -x_1\alpha_{12} - x_2\alpha_1\beta_1$ $d(x_2\alpha_2) = -x_2\alpha_{12} - 2x_2\alpha_2\beta_1$	$d(x_1\alpha_1\beta_1) = 0$ $d(x_1\alpha_2\beta_1) = 0$ $d(x_2\alpha_1\beta_1) = -x_1\alpha_{12}\beta_1$ $d(x_2\alpha_2\beta_1) = -x_2\alpha_{12}\beta_1$
$d(x_1) = x_1\alpha_1$ $d(x_2) = x_1\alpha_2 + x_2\beta_1$	$d(x_1\beta_1) = x_1\alpha_1\beta_1$ $d(x_2\beta_1) = x_1\alpha_2\beta_1$

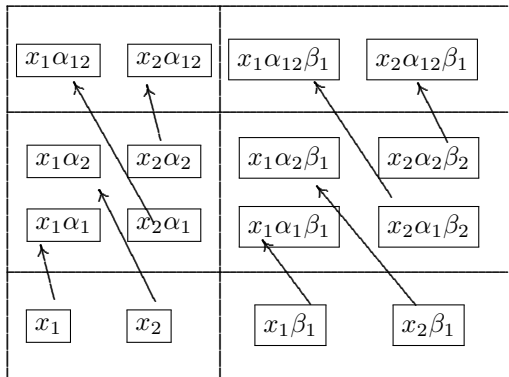
From here we can calculate d_0 . That is the part of d that is projected into the square above.

d_0 Table

$d_0(x_2\alpha_2) = -x_2\alpha_{12}$ $d_0(x_2\alpha_1) = -x_1\alpha_{12}$	$d_0(x_2\alpha_2\beta_1) = -x_2\alpha_{12}\beta_1$ $d_0(x_2\alpha_1\beta_1) = -x_1\alpha_{12}\beta_1$
$d_0(x_1) = x_1\alpha_1$ $d_0(x_2) = x_1\alpha_2$	$d_0(x_1\beta_1) = x_1\alpha_1\beta_1$ $d_0(x_2\beta_1) = x_1\alpha_2\beta_1$

Now to the E_0 page will be as follows:

E_0 Page



3.6.3 E_1 Page

There are no forms that survived the E_0 page. So the sequence has converged. Some sequences can converge in as few as one page as this example showed.

Cohomology

--	--	--	--

3.7 Example 4

In this example we will be doing a more complicated sequence of a Lie algebra with coefficients in a representation. We will also be checking our answer against maple and showing that they are equivalent.

3.7.1 Lie algebra

Let V be a vector space with basis $\{x_1, x_2, x_3\}$. Let g be a Lie algebra with coefficients in V . Now let $\{z_1, z_2\}$ be the basis for the sub-algebra in g that we will make this sequence with. Let $\{y_1, y_2\}$ form the complementary space. α will be the dual of the sub-algebra. β will be the dual of the complementary space. The following multiplication table and Lie coefficients table will describe how the forms and coefficients interact.

Multiplication Table

	z_1	z_2	y_1	y_2
z_1	0	0	$-z_1$	$-z_2$
z_2	0	0	$-z_2$	0
y_1	z_1	z_2	0	z_1
y_2	z_2	0	$-z_1$	0

Lie Coefficients Table

	x_1	x_2	x_3
z_1	0	$-x_1$	0
z_2	0	x_3	0
y_1	x_1	0	x_3
y_2	$-x_3$	$-x_1$	0

3.7.2 Calculating the exterior differential in a Lie algebra with coefficients in a representation

While we have calculated the exterior derivative of forms in a Lie algebra many times already in this thesis we have yet to talk about how the coefficients affect the exterior derivative. The coefficients don't actually change how we calculate the exterior derivative much. The exterior derivative follows Leibniz's rule. So if x is a coefficient and α is a form then

$$d(x\alpha) = d(x) \wedge \alpha - x d \wedge (\alpha)$$

Since we already know how to calculate $d(\alpha)$ using the multiplication table for the Lie algebra, the only question is how to calculate $d(x)$. To calculate $d(x)$ we will use the Lie coefficients table.

	x_1	x_2	x_3
z_1	0	$-x_1$	0
z_2	0	x_3	0
y_1	x_1	0	x_3
y_2	$-x_3$	$-x_1$	0

To find the exterior derivative of each coefficient we need only look at the columns of this table. Let x be a coefficient in this example. Then

$$d(x) = a\alpha_1 + b\alpha_2 + c\beta_1 + d\beta_2$$

where a,b,c,d are each some linear combination of coefficients. Now a,b,c,d are given explicitly in the table. The row with z_1 , the dual to α_1 , gives the coefficient to the α_1 term. The row with y_2 , the dual to β_2 , gives the coefficient to the β_2 term and so forth. So using this

$$d(x_1) = x_1\beta_1 - x_3\beta_2$$

$$d(x_2) = -x_1\alpha_1 + x_3\alpha_2 - x_1\beta_2$$

$$d(x_3) = x_3\beta_1$$

From the multiplication table we get:

$$d(\alpha_1) = \alpha_1\beta_1 - \beta_{12}$$

$$d(\alpha_2) = -\alpha_1\beta_2 - \alpha_2\beta_1$$

$$d(\beta_1) = 0$$

$$d(\beta_2) = 0$$

. So now that we have the differentials of the base forms and the coefficients lets calculate the exterior derivative of the following Lie algebra forms with coefficients in a representation.

Example 1

$$d(x_3\alpha_2) = d(x_3)\alpha_2 + x_3d(\alpha_2)$$

$$d(x_3\alpha_2) = x_3\beta_1\alpha_2 - x_3(-\alpha_1\beta_2 + \alpha_2\beta_1)$$

$$d(x_3\alpha_2) = x_3\beta_1\alpha_2 + x_3\alpha_1\beta_2 + x_3\alpha_2\beta_1$$

$$d(x_3\alpha_2) = -x_3\alpha_2\beta_1 + x_3\alpha_1\beta_2 + x_3\alpha_2\beta_1$$

$$d(x_3\alpha_2) = x_3\alpha_1\beta_2$$

Example 2

$$d(x_1\beta_1) = d(x_1)\beta_1 - x_1d(\beta_1)$$

$$d(x_1\beta_1) = (x_1\beta_1 - x_3\beta_2)\beta_1 - x_1(0)$$

$$d(x_1\beta_1) = x_1\beta_1\beta_1 - x_3\beta_2\beta_1$$

$$d(x_1\beta_1) = x_3\beta_{12}$$

Now we can calculate the entire differential table for all the forms:

Differential Table

$d(x_1\alpha_{12}) = -x_1\alpha_{12}\beta_3 - x_1\alpha_1\beta_{12} - x_3\alpha_{12}\beta_2$ $d(x_2\alpha_{12}) = -x_1\alpha_{12}\beta_2 - 2x_2\alpha_{12}\beta_1 - x_2\alpha_2\beta_{12}$ $d(x_3\alpha_{12}) = -x_3\alpha_{12}\beta_1 - x_3\alpha_1\beta_{12}$	$d(x_1\alpha_{12}\beta_1) = -x_3\alpha_{12}\beta_{12}$ $d(x_2\alpha_{12}\beta_1) = x_1\alpha_{12}\beta_{12}$ $d(x_3\alpha_{12}\beta_1) = 0$ $d(x_1\alpha_{12}\beta_2) = -x_1\alpha_{12}\beta_{12}$ $d(x_2\alpha_{12}\beta_2) = -2x_2\alpha_{12}\beta_{12}$ $d(x_3\alpha_{12}\beta_2) = -x_3\alpha_{12}\beta_{12}$	$d(x_1\alpha_{12}\beta_{12}) =$ $d(x_2\alpha_{12}\beta_{12}) =$ $d(x_3\alpha_{12}\beta_{12}) =$
$d(x_1\alpha_1) = -x_1\beta_{12} + x_3\alpha_1\beta_2$ $d(x_1\alpha_2) = x_1\alpha_1\beta_2 + x_3\alpha_2\beta_2$ $d(x_2\alpha_1) = x_1\alpha_1\beta_4 + x_2\alpha_1\beta_1 - x_2\beta_{12} - x_3\alpha_{12}$ $d(x_2\alpha_2) = -x_1\alpha_{12} + x_1\alpha_2\beta_2 + x_2\alpha_1\beta_2 + x_2\alpha_2\beta_1$ $d(x_3\alpha_1) = -x_3\beta_{12}$ $d(x_3\alpha_2) = x_3\alpha_1\beta_2$	$d(x_1\alpha_1\beta_1) = -x_3\alpha_1\beta_{12}$ $d(x_1\alpha_2\beta_1) = -x_1\alpha_1\beta_{12} - x_3\alpha_2\beta_{12}$ $d(x_2\alpha_1\beta_1) = -x_1\alpha_1\beta_{12} - x_3\alpha_{12}\beta_1$ $d(x_2\alpha_2\beta_1) = -x_1\alpha_{12}\beta_1 - x_1\alpha_2\beta_{12} - x_2\alpha_1\beta_{12}$ $d(x_3\alpha_1\beta_1) = 0$ $d(x_3\alpha_2\beta_1) = -x_3\alpha_1\beta_{12}$ $d(x_1\alpha_1\beta_2) = 0$ $d(x_1\alpha_2\beta_2) = 0$ $d(x_2\alpha_1\beta_2) = x_2\alpha_1\beta_{12} - x_3\alpha_{12}\beta_2$ $d(x_2\alpha_2\beta_2) = -x_1\alpha_{12}\beta_2 + x_2\alpha_2\beta_{12}$ $d(x_3\alpha_1\beta_2) = 0$ $d(x_3\alpha_2\beta_2) = 0$	$d(x_1\alpha_1\beta_{12}) = 0$ $d(x_1\alpha_2\beta_{12}) = 0$ $d(x_2\alpha_1\beta_{12}) = -x_3\alpha_{12}\beta_{12}$ $d(x_2\alpha_2\beta_{12}) = -x_1\alpha_{12}\beta_{12}$ $d(x_3\alpha_1\beta_{12}) = 0$ $d(x_3\alpha_2\beta_{12}) = 0$
$d(x_1) = x_1\beta_1 - x_3\beta_2$ $d(x_2) = -x_1\alpha_1 - x_1\beta_2 + x_3\alpha_2$ $d(x_3) = x_3\beta_1$	$d(x_1\beta_1) = x_3\beta_{12}$ $d(x_1\beta_2) = x_1\beta_{12}$ $d(x_2\beta_1) = -x_1\alpha_1\beta_1 + x_1\beta_{12} + x_3\alpha_2\beta_1$ $d(x_2\beta_2) = -x_1\alpha_1\beta_2 + x_3\alpha_2\beta_2$ $d(x_3\beta_1) = 0$ $d(x_3\beta_2) = x_3\beta_{12}$	$d(x_1\beta_{12}) = 0$ $d(x_2\beta_{12}) = -x_1\alpha_1\beta_{12} + x_3\alpha_2\beta_{12}$ $d(x_3\beta_{12}) = 0$

3.7.3 E_0

The next step is to calculate d_0 . If we recall the formula for d_0 is

$$d_0 : E_0^{p,q} \rightarrow E_0^{p,q+1}$$

Which essentially means that d_0 is the part of the differential that is projected to the square above.

d_0 Table

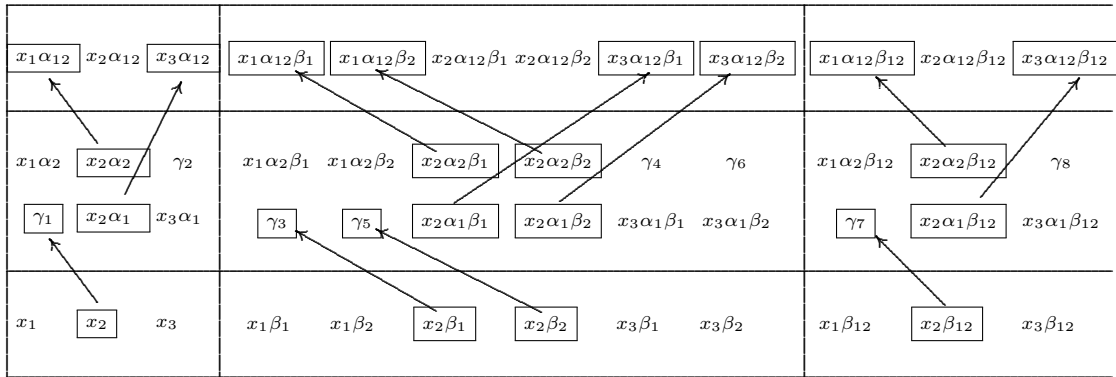
$d_0(x_1\alpha_{12}) = 0$ $d_0(x_2\alpha_{12}) = 0$ $d_0(x_3\alpha_{12}) = 0$	$d_0(x_1\alpha_{12}\beta_1) = 0$ $d_0(x_2\alpha_{12}\beta_1) = 0$ $d_0(x_3\alpha_{12}\beta_1) = 0$ $d_0(x_1\alpha_{12}\beta_2) = 0$ $d_0(x_2\alpha_{12}\beta_2) = 0$ $d_0(x_3\alpha_{12}\beta_2) = 0$	$d_0(x_1\alpha_{12}\beta_{12}) =$ $d(x_2\alpha_{12}\beta_{12}) =$ $d(x_3\alpha_{12}\beta_{12}) =$
$d_0(x_1\alpha_1) = 0$ $d_0(x_1\alpha_2) = 0$ $d_0(x_2\alpha_1) = -x_3\alpha_{12}$ $d_0(x_2\alpha_2) = -x_1\alpha_{12}$ $d_0(x_3\alpha_1) = 0$ $d_0(x_3\alpha_2) = 0$	$d_0(x_1\alpha_1\beta_1) = 0$ $d_0(x_1\alpha_2\beta_1) = 0$ $d_0(x_2\alpha_1\beta_1) = -x_3\alpha_{12}\beta_1$ $d_0(x_2\alpha_2\beta_1) = -x_1\alpha_{12}\beta_1$ $d_0(x_3\alpha_1\beta_1) = 0$ $d_0(x_3\alpha_2\beta_1) = 0$ $d_0(x_1\alpha_1\beta_2) = 0$ $d_0(x_1\alpha_2\beta_2) = 0$ $d_0(x_2\alpha_1\beta_2) = -x_3\alpha_{12}\beta_2$ $d_0(x_2\alpha_2\beta_2) = -x_1\alpha_{12}\beta_2$ $d_0(x_3\alpha_1\beta_2) = 0$ $d_0(x_3\alpha_2\beta_2) = 0$	$d_0(x_1\alpha_1\beta_{12}) = 0$ $d_0(x_1\alpha_2\beta_{12}) = 0$ $d_0(x_2\alpha_1\beta_{12}) = -x_3\alpha_{12}\beta_{12}$ $d_0(x_2\alpha_2\beta_{12}) = -x_1\alpha_{12}\beta_{12}$ $d_0(x_3\alpha_1\beta_{12}) = 0$ $d_0(x_3\alpha_2\beta_{12}) = 0$
$d_0(x_1) = 0$ $d_0(x_2) = -x_1\alpha_1 + x_3\alpha_2$ $d_0(x_3) = 0$	$d_0(x_1\beta_1) = 0$ $d_0(x_1\beta_2) = 0$ $d(x_2\beta_1) = -x_1\alpha_1\beta_1 + x_3\alpha_2\beta_1$ $d(x_2\beta_2) = -x_1\alpha_1\beta_2 + x_3\alpha_2\beta_2$ $d(x_3\beta_1) = 0$ $d(x_3\beta_2) = 0$	$d_0(x_1\beta_{12}) = 0$ $d(x_2\beta_{12}) = -x_1\alpha_1\beta_{12} + x_3\alpha_2\beta_{12}$ $d(x_3\beta_{12}) = 0$

From the above table it is clear that we are going to need to change to basis of several of the squares. The following chart has the new changes.

Change of basis table

$\gamma_1 = -x_1\alpha_1 + x_3\alpha_2$	$\gamma_2 = x_1\alpha_1 + x_3\alpha_2$
$\gamma_3 = -x_1\alpha_1\beta_1 + x_3\alpha_2\beta_1$	$\gamma_4 = x_3\alpha_2\beta_1 + x_1\alpha_1\beta_3$
$\gamma_5 = -x_1\alpha_1\beta_2 + x_3\alpha_2\beta_2$	$\gamma_6 = x_1\alpha_1\beta_2 + x_3\alpha_2\beta_2$
$\gamma_7 = -x_1\alpha_1\beta_{12} + x_3\alpha_2\beta_{12}$	$\gamma_8 = x_1\alpha_1\beta_{12} + x_3\alpha_2\beta_{12}$

E_0 Page



3.7.4 E_1

Let's look at what forms are left.

\circ $x_{2\alpha_{12}}$ \circ	\circ \circ $x_{2\alpha_{12}\beta_1}$ $x_{2\alpha_{12}\beta_2}$ \circ \circ	\circ $x_{2\alpha_{12}\beta_{12}}$ \circ
$x_{1\alpha_2}$ \circ γ_2	$x_{1\alpha_2\beta_1}$ $x_{1\alpha_2\beta_2}$ \circ \circ γ_4 γ_6	$x_{1\alpha_2\beta_{12}}$ \circ γ_8
\circ \circ $x_{3\alpha_1}$	\circ \circ \circ \circ $x_{3\alpha_1\beta_1}$ $x_{3\alpha_1\beta_2}$	\circ \circ $x_{3\alpha_1\beta_{12}}$
x_1 \circ x_3	$x_{1\beta_1}$ $x_{1\beta_2}$ \circ \circ $x_{3\beta_1}$ $x_{3\beta_2}$	$x_{1\beta_{12}}$ \circ $x_{3\beta_{12}}$

From the remaining forms we will calculate d_1 . Recall that d_1 is the part of the differential that is projected into the square on the right. It's formula is

$$d_1 : E_1^{p,q} \rightarrow E_0^{p+1,q}$$

The following is the d_1 table.

d_1 Table

$d_1(x_2\alpha_{12}) = -2x_2\alpha_{12}\beta_1$	$d_1(x_2\alpha_{12}\beta_1) = 0$ $d_1(x_2\alpha_{12}\beta_2) = -2x_2\alpha_{12}\beta_{12}$	$d_1(x_2\alpha_{12}\beta_{12}) = 0$
$d_1(x_1\alpha_2) = x_1\alpha_1\beta_2 + x_3\alpha_2\beta_2 = \gamma_6$ $d_1(x_3\alpha_1) = 0$ $d_1(\gamma_2) = x_3\alpha_1\beta_2$	$d_1(x_1\alpha_2\beta_1) = -x_1\alpha_1\beta_{12} - x_3\alpha_2\beta_{12} = -\gamma_8$ $d_1(x_3\alpha_1\beta_1) = 0$ $d_1(\gamma_4) = -x_3\alpha_1\beta_{12}$ $d_1(x_1\alpha_2\beta_2) = 0$ $d_1(x_3\alpha_1\beta_2) = 0$ $d_1(\gamma_6) = 0$	$d_1(x_1\alpha_2\beta_{12}) = 0$ $d_1(x_3\alpha_1\beta_{12}) = 0$ $d_1(\gamma_8) = 0$
$d_1(x_1) = x_1\beta_1 - x_3\beta_2 = \gamma_9$ $d_1(x_3) = x_3\beta_1$	$d_1(\gamma_9) = 0$ $d_1(x_1\beta_2) = x_1\beta_{12}$ $d_1(x_3\beta_1) = 0$ $d_1(\gamma_{10}) = 2x_3\beta_{12}$	$d_1(x_1\beta_{12}) = 0$ $d_1(x_3\beta_{12}) = 0$

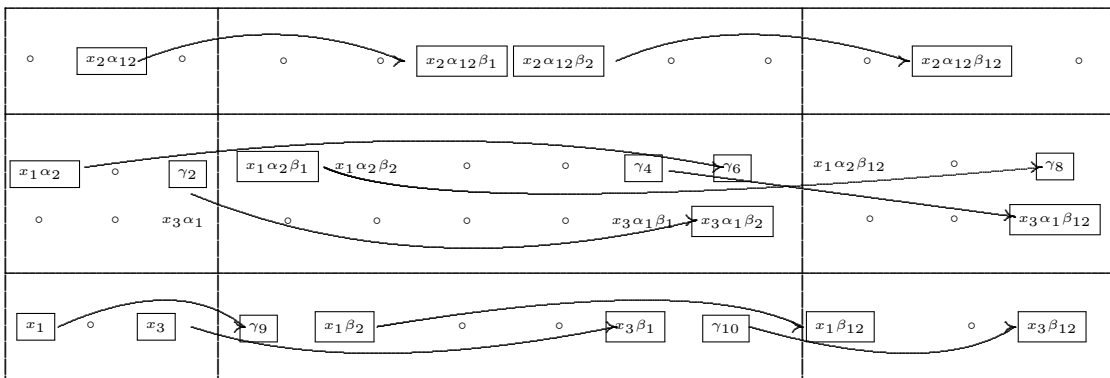
from the table we must again change the basis of a square. We will change the basis of the $E^{1,0}$ square. The following table will now show the changes:

Change of basis table

$\gamma_1 = -x_1\alpha_1 + x_3\alpha_2$	$\gamma_2 = x_1\alpha_1 + x_3\alpha_2$
$\gamma_3 = -x_1\alpha_1\beta_1 + x_3\alpha_2\beta_1$	$\gamma_4 = x_3\alpha_2\beta_1 + x_1\alpha_1\beta_3$
$\gamma_5 = -x_1\alpha_1\beta_2 + x_3\alpha_2\beta_2$	$\gamma_6 = x_1\alpha_1\beta_2 + x_3\alpha_2\beta_2$
$\gamma_7 = -x_1\alpha_1\beta_{12} + x_3\alpha_2\beta_{12}$	$\gamma_8 = x_1\alpha_1\beta_{12} + x_3\alpha_2\beta_{12}$
$\gamma_9 = x_1\beta_1 - x_3\beta_2$	$\gamma_{10} = x_1\beta_1 + x_3\beta_2$

Now we can calculate E_1 .

E_1 Page



3.7.5 E_2

The sequence will converge on the E_2 page. Only four forms are left in the sequence. The d_2 table for them is the following:

 d_2 Table

$d_2(x_3\alpha_1) = -x_3\beta_{12} = -\frac{1}{2}d_1(\gamma_{10})$	$d_2(x_1\alpha_2\beta_2) = 0$ $d_2(x_3\alpha_1\beta_1) = 0$	$d_2(x_1\alpha_2\beta_{12}) = 0$

One chain is created the rest are zero. So the page becomes:

 E_2 Page

○ ○ ○	○ ○ ○ ○ ○ ○ ○	○ ○ ○
○ ○ ○	○ $x_1\alpha_2\beta_2$ ○ ○ ○ ○ ○	$x_1\alpha_2\beta_{12}$ ○ ○
○ ○ $[x_3\alpha_1]$	○ ○ ○ ○ $x_3\alpha_1\beta_1$ ○	○ ○ ○
○ ○ ○	○ ○ ○ ○ ○ ○ ○	○ ○ ○

We know that this sequence has converged because all of the differentials on any subsequent page will be projecting to squares not on the grid. So the forms and chains that are left form the cohomology of this Lie algebra.

3.7.6 Checking The Answer

We will now check our answer against one that was found in maple. The first list is the cohomology calculated by our by hand method:

Cohomology from calculating the sequence:			
$x_3\alpha_1 + \frac{1}{2}x_1\beta_1 + \frac{1}{2}x_3\beta_2$	$x_1\alpha_2\beta_2$	$x_1\alpha_2\beta_{12}$	
	$x_3\alpha_1\beta_1$		

The second list is the cohomology calculated in maple using the cohomology command.

Cohomology from maple:			
$x_3\alpha_1 + \frac{1}{2}x_1\beta_1 + \frac{1}{2}x_3\beta_2$	$\frac{1}{3}x_1\alpha_{12} - \frac{1}{3}x_2\alpha_1\beta_2 - \frac{1}{3}x_2\alpha_2\beta_1 + x_1\alpha_2\beta_2$	$-\frac{2}{3}x_1\alpha_{12}\beta_1 + x_1\alpha_2\beta_{12} - \frac{1}{3}x_2\alpha_1\beta_{12} - \frac{1}{3}x_3\alpha_{12}\beta_2$	
	$x_3\alpha_1\beta_1$		

Now the cohomology space for the 1-forms is the same. It is $x_3\alpha_1\frac{1}{2}x_1\beta_1 + \frac{1}{2}x_3\beta_2$ for both. For the 2-forms it is similar. Both contain the form $x_3\alpha_1\beta_1$, but they differ in the second form. Cohomology spaces are vector quotient spaces. So two forms are equal if they just differ by an exact form.

Cohomology Decomposition of $x_1\alpha_2\beta_2$ is

$$\frac{1}{4}x_1\alpha_{12} - \frac{1}{4}x_2\alpha_1\beta_2 - \frac{1}{4}x_2\alpha_1\beta_2 + \frac{3}{4}x_1\alpha_2\beta_2, \quad \frac{1}{4}x_2\alpha_2$$

So the two forms differ by an exact form $x_2\alpha_2$.

Next we need to check the 3-forms. They also are different. By calculating their cohomology decomposition we find out that they are the same.

Cohomology Decomposition of $x_1\alpha_2\beta_{12}$

$$-\frac{2}{5}x_1\alpha_{12}\beta_1 + \frac{3}{5}x_1\alpha_2\beta_{12} - \frac{1}{5}x_2\alpha_1\beta_{12} - \frac{1}{5}x_3\alpha_{12}\beta_2, \quad -\frac{2}{5}x_1\alpha_{12} + \frac{1}{5}x_2\alpha_1\beta_2$$

4. PROGRAMS

To calculate the Cohomology and Serre Hochschild spectral sequences I wrote a number of the programs in maple. This chapter deals with the software that I wrote. I first wrote a number of conversion programs so that I was able to quickly go back and forth between forms and vectors. I then wrote software to calculate the closed forms and exact forms. Lastly I wrote a framework to calculate the Serre Hochschild spectral sequences.

4.1 Conversion Programs

4.1.1 Counting an ordered set of Basis Elements of Differential Forms

In maple the basis elements are placed in a set in ascending order. To quickly convert the object into a vector a counting procedure is used to find its place in the set.

Differential Forms

Example Basis Ordered List

Let the basis of 1-forms of Lie-Algebra be $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ The basis for the forms is listed in ascending order. For example the basis for two forms will be

$$\{\theta_1 \wedge \theta_2, \theta_1 \wedge \theta_3, \theta_1 \wedge \theta_4, \theta_2 \wedge \theta_3, \theta_2 \wedge \theta_4, \theta_3 \wedge \theta_4\}$$

Notice that $\theta_n \wedge \theta_n$ for any n is not part of the list. That is due to the alternating properties of the wedge product. Any such wedge product would be zero.

Also note that $\theta_2 \wedge \theta_1$ is not part of the list nor is any $\theta_k \wedge \theta_j$ for $k > j$. By the alternating property $\theta_k \wedge \theta_j = -1 * \theta_j \wedge \theta_k$. Now since $k > j$ the basis element $\theta_j \wedge \theta_k$ has already been listed.

Thus the way to count the number of basis elements of a certain size r with a 1-form basis size n is equivalent to counting the number of unordered lists of size r from a group of size n or $\binom{n}{r}$. For the above example with a 1-form basis of size 4 and a k -form dimension of 2 the number of basis elements is $\binom{4}{2}$ which is 6. This was the number of basis elements that was listed.

Finding out what position a basis element has in the ordered list of differential forms

Hockey Stick Identity

The Hockey Stick Identity is as follows:

$$\sum_{i=r}^n \binom{i}{r} = \binom{n+1}{r+1}$$

In the above counting algorithm in the first iteration we count

$$\sum_{i=0}^k \binom{n-i-1}{r}$$

Now using the Hockey Stick Identity this becomes

$$\sum_{i=0}^k \binom{n-i-1}{r} = \binom{n}{r+1} - \binom{n-k-1}{r+1}$$

Proof

$\sum_{i=0}^k \binom{n-i-1}{r} = \binom{n-1}{r} + \binom{n-2}{r} + \dots + \binom{n-k-1}{r}$ can be rewritten with a change of indices as $\sum_{i=n-k-1}^{n-1} \binom{i}{r}$

Now using the Hockey Stick Identity from $i = 0$ to $i = n-1$ gives $\binom{n}{r+1}$. This the original summation went from $i = n-k-1$ to $i = n-1$ so the summation from $i = 0$ to $i = n-k-2$ must be subtracted from the other giving the equation above.

4.1.2 Program for Form Base Elements: FormToList

X is a list of components of monomial form

K is the dimension of space of 1-forms

Example

$X = [3,4,5]$, $K=6$

The number of terms with a theta1 equals the number of two forms in 5 variables= $C(5,2)$.

The number of terms with a theta2 equals the number of two forms in 4 variables= $C(4,2)$.

This sum is computed in the first for loop.

Call $X[1]-1=a$. Computing $\sum_{i=1}^{a-1} \binom{K-i}{n-1}$.

Calling Sequence

FormToList(S1)

Parameters

S1 - A list of differential forms

Description

- Let S1 be a list of differential forms. FormToList will convert the form or list of forms into its vector representation.

Examples

```
> with(Library);  
      [Browse, DGplot, DGshow, MetricSearch, References, Retrieve, Search] (1)
```

```
> LD:=Retrieve("Winternitz",1,[6,19],alg1);  
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3]=e6, [e2,  
      e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0, [e4, e5]=0, [e4,  
      e6]=0, [e5, e6]=0 (2)
```

```
> DGsetup(LD);  
      Lie algebra: alg1 (3)
```

```
> MultiplicationTable();  
      alg1 | e1 e2 e3 e4 e5 e6  
      e1 | 0 e3 e4 e5 e6 0  
      e2 | -e3 0 e6 0 0 0  
      e3 | -e4 -e6 0 0 0 0  
      e4 | -e5 0 0 0 0 0  
      e5 | -e6 0 0 0 0 0  
      e6 | 0 0 0 0 0 0 (4)
```

Example 1

Finding the vector representation of one form

```
alg1 > FormToList([theta1]);
```

```
      [ 1 ]  
      [ 0 ]  
      [ 0 ]  
      [ 0 ]  
      [ 0 ]  
      [ 0 ] (5)
```

Example 2

```
alg1 > TestForm1 := evalDG(theta1 + theta4);
```

$$\text{TestForm1} := \theta_1 + \theta_4$$

(6)

```
alg1 > FormToList([TestForm1]);
```

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

(7)

Example 3

Finding vector representation of multiple forms at once

```
alg1 > TestForm2 := evalDG(2 * theta3 + theta1);
```

$$\text{TestForm2} := \theta_1 + 2 \theta_3$$

(8)

```
alg1 > TestForm3 := evalDG(3 * theta5 - theta2);
```

$$\text{TestForm3} := -\theta_2 + 3 \theta_5$$

(9)

```
alg1 > FormToList([TestForm2, TestForm3]);
```

$$\begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 3 \\ 0 \end{bmatrix}$$

(10)

Example 4

Finding the vector representation of a 2-form

```
alg1 > interface(rtablesize=60):
```

```
alg1 > B := evalDG([theta1, theta2, theta3, theta4, theta5, theta6]);
```

$$B := [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$$

(11)

```
alg1 > C := GenerateForms(B, 2);
```

```
C := [\theta_1 \wedge \theta_2, \theta_1 \wedge \theta_3, \theta_1 \wedge \theta_4, \theta_1 \wedge \theta_5, \theta_1 \wedge \theta_6, \theta_2 \wedge \theta_3, \theta_2 \wedge \theta_4, \theta_2 \wedge \theta_5, \theta_2 \wedge \theta_6, \theta_3 \wedge \theta_4, \theta_3 \wedge \theta_5, \theta_3 \wedge \theta_6, \theta_4 \wedge \theta_5, \theta_4 \wedge \theta_6, \theta_5 \wedge \theta_6]
```

(12)

(13)

20

(14)

```
alg1 > TestForm5 := evalDG(sum(i * C[i], i = 1..15));
```

```
TestForm5 := \theta_1 \wedge \theta_2 + 2 \theta_1 \wedge \theta_3 + 3 \theta_1 \wedge \theta_4 + 4 \theta_1 \wedge \theta_5 + 5 \theta_1 \wedge \theta_6 + 6 \theta_2 \wedge \theta_3 + 7 \theta_2 \wedge \theta_4 + 8 \theta_2 \wedge \theta_5 + 9 \theta_2 \wedge \theta_6 + 10 \theta_3 \wedge \theta_4 + 11 \theta_3 \wedge \theta_5 + 12 \theta_3 \wedge \theta_6 + 13 \theta_4 \wedge \theta_5 + 14 \theta_4 \wedge \theta_6 + 15 \theta_5 \wedge \theta_6
```

(15)

```
alg1 > FormToList([TestForm5]);
```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

```

(16)

Example 5

Finding the vector representation of a 3-form

```
alg1 > E := GenerateForms(B, 3);
```

```
E := [ θ1 ∧ θ2 ∧ θ3, θ1 ∧ θ2 ∧ θ4, θ1 ∧ θ2 ∧ θ5, θ1 ∧ θ2 ∧ θ6, θ1 ∧ θ3 ∧ θ4, θ1 ∧ θ3 ∧ θ5,
      θ1 ∧ θ3 ∧ θ6, θ1 ∧ θ4 ∧ θ5, θ1 ∧ θ4 ∧ θ6, θ1 ∧ θ5 ∧ θ6, θ2 ∧ θ3 ∧ θ4, θ2 ∧ θ3 ∧ θ5, θ2
      ∧ θ3 ∧ θ6, θ2 ∧ θ4 ∧ θ5, θ2 ∧ θ4 ∧ θ6, θ2 ∧ θ5 ∧ θ6, θ3 ∧ θ4 ∧ θ5, θ3 ∧ θ4 ∧ θ6, θ3
      ∧ θ5 ∧ θ6, θ4 ∧ θ5 ∧ θ6 ]
```

(17)

```
alg1 > TestForm6 := evalDG(sum(i * E[i], i = 1..20));
```

```
TestForm6 := θ1 ∧ θ2 ∧ θ3 + 2 θ1 ∧ θ2 ∧ θ4 + 3 θ1 ∧ θ2 ∧ θ5 + 4 θ1 ∧ θ2 ∧ θ6 + 5 θ1
      ∧ θ3 ∧ θ4 + 6 θ1 ∧ θ3 ∧ θ5 + 7 θ1 ∧ θ3 ∧ θ6 + 8 θ1 ∧ θ4 ∧ θ5 + 9 θ1 ∧ θ4 ∧ θ6
      + 10 θ1 ∧ θ5 ∧ θ6 + 11 θ2 ∧ θ3 ∧ θ4 + 12 θ2 ∧ θ3 ∧ θ5 + 13 θ2 ∧ θ3 ∧ θ6 + 14 θ2
      ∧ θ4 ∧ θ5 + 15 θ2 ∧ θ4 ∧ θ6 + 16 θ2 ∧ θ5 ∧ θ6 + 17 θ3 ∧ θ4 ∧ θ5 + 18 θ3 ∧ θ4 ∧ θ6
      + 19 θ3 ∧ θ5 ∧ θ6 + 20 θ4 ∧ θ5 ∧ θ6
```

(18)

```
alg1 > FormToList([TestForm6]);
```

[

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20

4.1.3 *Tensor To list*

This program takes a tensor and converts it to a vector.

Calling Sequence

TensorToList(S1)

Parameters

S1 - A list of tensors

Description

- Let S1 be a list of tensors. TensorToList takes a tensor or list of tensors and returns a list of their vector representations.

Examples

```
> with(Library);
    [Browse, DGplot, DGshow, MetricSearch, References, Retrieve, Search] (1)
```

```
> LD:=Retrieve("Winternitz",1,[6,19],alg1);
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3]=e6, [e2,
    e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0, [e4, e5]=0, [e4,
    e6]=0, [e5, e6]=0 (2)
```

```
> DGsetup(LD);
    Lie algebra: alg1 (3)
```

```
> MultiplicationTable();
    alg1 | e1 e2 e3 e4 e5 e6
    e1 | 0 e3 e4 e5 e6 0
    e2 | -e3 0 e6 0 0 0
    e3 | -e4 -e6 0 0 0 0
    e4 | -e5 0 0 0 0 0
    e5 | -e6 0 0 0 0 0
    e6 | 0 0 0 0 0 0 (4)
```

```
alg1 > B:=evalDG([theta1, theta2, theta3, theta4, theta5, theta6])
;
    B := [theta1, theta2, theta3, theta4, theta5, theta6] (5)
```

```
alg1 > C:=GenerateTensors([B,B]);
C := [theta1 theta1, theta1 theta2, theta1 theta3, theta1 theta4, theta1 theta5, theta1 theta6, theta2 theta1, theta2 theta2, theta2 theta3, theta2
    theta4, theta2 theta5, theta2 theta6, theta3 theta1, theta3 theta2, theta3 theta3, theta3 theta4, theta3 theta5, theta3 theta6, theta4 theta1,
    theta4 theta2, theta4 theta3, theta4 theta4, theta4 theta5, theta4 theta6, theta5 theta1, theta5 theta2, theta5 theta3, theta5 theta4, theta5
    theta5, theta5 theta6, theta6 theta1, theta6 theta2, theta6 theta3, theta6 theta4, theta6 theta5, theta6 theta6] (6)
```

```
alg1 > interface(rtablesize=60);
    10 (7)
```

Example 1

Converting one tensor to a vector

```
alg1 > TensorToList([theta1]);
```

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

(8)

Example 2

Converting multiple tensors to a list.

```
alg1 > TensorToList(B);
```

$$\begin{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix}$$

(9)

Example 3

Converting larger tensors to a list

```
alg1 > test := evalDG(sum(i * C[i], i = 1..36));
```

$test := \theta_1 \otimes \theta_1 + 2 \theta_1 \otimes \theta_2 + 3 \theta_1 \otimes \theta_3 + 4 \theta_1 \otimes \theta_4 + 5 \theta_1 \otimes \theta_5 + 6 \theta_1 \otimes \theta_6 + 7 \theta_2 \otimes \theta_1$
 $+ 8 \theta_2 \otimes \theta_2 + 9 \theta_2 \otimes \theta_3 + 10 \theta_2 \otimes \theta_4 + 11 \theta_2 \otimes \theta_5 + 12 \theta_2 \otimes \theta_6 + 13 \theta_3 \otimes \theta_1 + 14 \theta_3$
 $\otimes \theta_2 + 15 \theta_3 \otimes \theta_3 + 16 \theta_3 \otimes \theta_4 + 17 \theta_3 \otimes \theta_5 + 18 \theta_3 \otimes \theta_6 + 19 \theta_4 \otimes \theta_1 + 20 \theta_4 \otimes \theta_2$
 $+ 21 \theta_4 \otimes \theta_3 + 22 \theta_4 \otimes \theta_4 + 23 \theta_4 \otimes \theta_5 + 24 \theta_4 \otimes \theta_6 + 25 \theta_5 \otimes \theta_1 + 26 \theta_5 \otimes \theta_2$
 $+ 27 \theta_5 \otimes \theta_3 + 28 \theta_5 \otimes \theta_4 + 29 \theta_5 \otimes \theta_5 + 30 \theta_5 \otimes \theta_6 + 31 \theta_6 \otimes \theta_1 + 32 \theta_6 \otimes \theta_2$
 $+ 33 \theta_6 \otimes \theta_3 + 34 \theta_6 \otimes \theta_4 + 35 \theta_6 \otimes \theta_5 + 36 \theta_6 \otimes \theta_6$

(10)

```
alg1 > TensorToList([test,C[3]]);
```

1	0
2	0
3	1
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0
30	0
31	0
32	0
33	0
34	0

4.1.4 *Tensor valued form to list*

This program takes a tensor valued form and converts it to a vector.

Calling Sequence

TensorValuedFormToList(S1)

Parameters

S1 - A list of tensor valued forms.

Description

- Let S1 be a list of tensor valued forms. TensorValuedFormToList returns a list of vectors corresponding to vector representation of the the tensor valued forms.

Examples

```

> A:=[Matrix([[1,0],[0,0]]),Matrix([[0,1],[0,0]]),Matrix([[0,0],[0,1]])];

```

$$A := \left[\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right] \quad (1)$$

```

> DGEEnvironment[VectorSpace](2,V,vectorlabels=[x1,x2])
      Vector Space: V

```

```

> LD:=LieAlgebraData(A,alg1);
      LD := [e1, e2]=e2, [e1, e3]=0, [e2, e3]=e2

```

```

> DGsetup(LD);
      Lie algebra: alg1

```

```

alg1 > rho:=Representation(alg1,V,A);

```

$$\rho := e1 \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, e2 \rightarrow \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, e3 \rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (5)$$

```

alg1 > DGEEnvironment[LieAlgebraWithCoefficients](rho,alg2);
      Lie algebra with coefficients: alg2

```

```

alg2 > MultiplicationTable();

```

alg2	e1	e2	e3
e1	0	e2	0
e2	-e2	0	e2
e3	0	-e2	0

```

alg2 > q:=RelativeChains([]);
q := [[x1 ⊗ 1, x2 ⊗ 1], [x1 ⊗ θ1, x1 ⊗ θ2, x1 ⊗ θ3, x2 ⊗ θ1, x2 ⊗ θ2, x2 ⊗ θ3], [x1 ⊗ θ1
      ∧ θ2, x1 ⊗ θ1 ∧ θ3, x1 ⊗ θ2 ∧ θ3, x2 ⊗ θ1 ∧ θ2, x2 ⊗ θ1 ∧ θ3, x2 ⊗ θ2 ∧ θ3], [x1 ⊗ θ1
      ∧ θ2 ∧ θ3, x2 ⊗ θ1 ∧ θ2 ∧ θ3], []]

```

Example 1

Finding the vector corresponding to a tensor valued form.

```
alg2 > test1 := q[1][1];
```

$$test1 := x1 \otimes 1$$

(9)

```
alg2 > TensorValuedFormToList([test1]);
```

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

(10)

Example 2

Finding the vectors corresponding to multiple tensor valued forms.

```
alg2 > q[3];
```

$$[x1 \otimes \theta1 \wedge \theta2, x1 \otimes \theta1 \wedge \theta3, x1 \otimes \theta2 \wedge \theta3, x2 \otimes \theta1 \wedge \theta2, x2 \otimes \theta1 \wedge \theta3, x2 \otimes \theta2 \wedge \theta3]$$

(11)

```
alg2 > TensorValuedFormToList(q[3]);
```

$$\begin{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix}$$

(12)

Example 3

Finding the vector corresponding to one compound tensor valued form

```
alg2 > test2 := evalDG( 2 *q[4][1] - q[4][2]);
```

$$test2 := 2 x1 \otimes \theta1 \wedge \theta2 \wedge \theta3 - x2 \otimes \theta1 \wedge \theta2 \wedge \theta3$$

(13)

```
alg2 > TensorValuedFormToList([test2]);
```

$$\begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

(14)

4.1.5 *Complementary Basis*

This program calculate the complementary basis.

Calling Sequence

JacComp(S1)

Parameters

S1 - A list of differential forms.

Description

- Let S1 be a list of differential forms. JacComp calculates the complementary basis of S1 in the basis, B1, that the forms come from. It returns a list of forms, C1, such that the union of S1 and C1 create a spanning set of B1.

Examples

```
> LD:=SimpleLieAlgebraData("so(3,2)",so32):
> DGsetup(LD);
Lie algebra: so32 (1)
so32 > B:=[seq(cat(theta,i),i=1..10)];
B := [theta1, theta2, theta3, theta4, theta5, theta6, theta7, theta8, theta9, theta10] (2)
Example 1
Complementary basis of a basis of the 1-forms in so32
so32 > JacComp(B);
[] (3)
Example 2
Complementary basis of a subset of the 1-forms in so32
so32 > X1:=evalDG([2*theta1+theta2, theta3-theta4, theta1]);
X1 := [2 theta1 + theta2, theta3 - theta4, theta1] (4)
so32 > JacComp(X1);
[theta4, theta5, theta6, theta7, theta8, theta9, theta10] (5)
Example 3
Complementary basis of a subset of 2-forms in so32
so32 > X:=evalDG([theta1 &w theta2, theta3 &w theta10, theta4
&w theta5, theta4 &w theta6]);
X := [theta1 & theta2, theta3 & theta10, theta4 & theta5, theta4 & theta6] (6)
so32 > JacComp(X);
[theta1 & theta3, theta1 & theta4, theta1 & theta5, theta1 & theta6, theta1 & theta7, theta1 & theta8, theta1 & theta9, theta1 & theta10, theta2 & theta3, theta2 & theta4,
theta2 & theta5, theta2 & theta6, theta2 & theta7, theta2 & theta8, theta2 & theta9, theta2 & theta10, theta3 & theta4, theta3 & theta5, theta3 & theta6, theta3
& theta7, theta3 & theta8, theta3 & theta9, theta4 & theta7, theta4 & theta8, theta4 & theta9, theta4 & theta10, theta5 & theta6, theta5 & theta7, theta5 & theta8,
theta5 & theta9, theta5 & theta10, theta6 & theta7, theta6 & theta8, theta6 & theta9, theta6 & theta10, theta7 & theta8, theta7 & theta9, theta7 & theta10, theta8
& theta9, theta8 & theta10, theta9 & theta10] (7)
(8)
```

4.2 *Cohomology Programs*

4.2.1 *Closed Forms*

This program calculates the closed forms.

Calling Sequence

Closed(S1)

Parameters

S1 - A list of differential forms

Description

• Let S1 be a list differential forms. Closed calculates the basis for the closed forms in S1.

Examples

Using the same Lie algebra from 2.3 example 1

```
> DGEEnvironment[LieAlgebra] ("Winternitz", 1, [5, 5], alg);  
Lie algebra: alg
```

(1)

```
> MultiplicationTable();
```

alg	$e1$	$e2$	$e3$	$e4$	$e5$
$e1$	0	0	0	0	0
$e2$	0	0	0	0	$e1$
$e3$	0	0	0	$e1$	$e2$
$e4$	0	0	$-e1$	0	0
$e5$	0	$-e1$	$-e2$	0	0

(2)

```
alg > RC := RelativeChains([]);
```

```
RC := [[ ], [  $\theta1, \theta2, \theta3, \theta4, \theta5$  ], [  $\theta1 \wedge \theta2, \theta1 \wedge \theta3, \theta1 \wedge \theta4, \theta1 \wedge \theta5, \theta2 \wedge \theta3, \theta2 \wedge \theta4, \theta2 \wedge \theta5, \theta3 \wedge \theta4, \theta3 \wedge \theta5, \theta4 \wedge \theta5$  ], [  $\theta1 \wedge \theta2 \wedge \theta3, \theta1 \wedge \theta2 \wedge \theta4, \theta1 \wedge \theta2 \wedge \theta5, \theta1 \wedge \theta3 \wedge \theta4, \theta1 \wedge \theta3 \wedge \theta5, \theta1 \wedge \theta4 \wedge \theta5, \theta2 \wedge \theta3 \wedge \theta4, \theta2 \wedge \theta3 \wedge \theta5, \theta2 \wedge \theta4 \wedge \theta5, \theta3 \wedge \theta4 \wedge \theta5$  ], [  $\theta1 \wedge \theta2 \wedge \theta3 \wedge \theta4, \theta1 \wedge \theta2 \wedge \theta3 \wedge \theta5, \theta1 \wedge \theta2 \wedge \theta4 \wedge \theta5, \theta1 \wedge \theta3 \wedge \theta4 \wedge \theta5, \theta2 \wedge \theta3 \wedge \theta4 \wedge \theta5$  ], [  $\theta1 \wedge \theta2 \wedge \theta3 \wedge \theta4 \wedge \theta5$  ], [ ]]
```

(3)

Example 1

The closed 1-forms

```
alg > Closed(RC[2]);
```

[$\theta5, \theta4, \theta3$]

(4)

Example 2

The closed 2-forms

```
alg > Closed(RC[3]);
```

[$\theta3 \wedge \theta5, \theta3 \wedge \theta4, \theta2 \wedge \theta5, \theta1 \wedge \theta5 + \theta2 \wedge \theta4, \theta2 \wedge \theta3, \theta4 \wedge \theta5$]

(5)

Example 3

The closed 3-forms

```
alg > Closed(RC[4]);
```


$$\left[\theta_3 \wedge \theta_4 \wedge \theta_5, \theta_2 \wedge \theta_4 \wedge \theta_5, \theta_2 \wedge \theta_3 \wedge \theta_5, \theta_2 \wedge \theta_3 \wedge \theta_4, \theta_1 \wedge \theta_4 \wedge \theta_5, \theta_1 \wedge \theta_3 \wedge \theta_5, -\theta_1 \wedge \theta_2 \wedge \theta_5 + \theta_1 \wedge \theta_3 \wedge \theta_4, \theta_1 \wedge \theta_2 \wedge \theta_3 \right] \quad (6)$$

Example 4

The closed 4-forms

alg > Closed(RC[5]);

$$\left[\theta_2 \wedge \theta_3 \wedge \theta_4 \wedge \theta_5, \theta_1 \wedge \theta_3 \wedge \theta_4 \wedge \theta_5, \theta_1 \wedge \theta_2 \wedge \theta_4 \wedge \theta_5, \theta_1 \wedge \theta_2 \wedge \theta_3 \wedge \theta_5, \theta_1 \wedge \theta_2 \wedge \theta_3 \wedge \theta_4 \right] \quad (7)$$

4.2.2 *Exact Forms*

This program calculates the exact forms.

Calling Sequence

Exact(S1)

Parameters

S1 - A list of differential forms

Description

• Let S1 a list of differential forms. Exact returns a basis for the exact forms in S1.

Examples

```

Using the same Lie algebra from 2.3 example 1
> DGEnvironment[LieAlgebra] ("Winternitz", 1, [5, 5], alg);
                                Lie algebra: alg
> MultiplicationTable();

```

alg	e1	e2	e3	e4	e5
e1	0	0	0	0	0
e2	0	0	0	0	e1
e3	0	0	0	e1	e2
e4	0	0	-e1	0	0
e5	0	-e1	-e2	0	0

```

> RC := RelativeChains([]);
RC := [[ ], [θ1, θ2, θ3, θ4, θ5], [θ1 ∧ θ2, θ1 ∧ θ3, θ1 ∧ θ4, θ1 ∧ θ5, θ2 ∧ θ3, θ2 ∧ θ4, θ2
  ∧ θ5, θ3 ∧ θ4, θ3 ∧ θ5, θ4 ∧ θ5], [θ1 ∧ θ2 ∧ θ3, θ1 ∧ θ2 ∧ θ4, θ1 ∧ θ2 ∧ θ5, θ1 ∧ θ3
  ∧ θ4, θ1 ∧ θ3 ∧ θ5, θ1 ∧ θ4 ∧ θ5, θ2 ∧ θ3 ∧ θ4, θ2 ∧ θ3 ∧ θ5, θ2 ∧ θ4 ∧ θ5, θ3 ∧ θ4
  ∧ θ5], [θ1 ∧ θ2 ∧ θ3 ∧ θ4, θ1 ∧ θ2 ∧ θ3 ∧ θ5, θ1 ∧ θ2 ∧ θ4 ∧ θ5, θ1 ∧ θ3 ∧ θ4 ∧ θ5,
  θ2 ∧ θ3 ∧ θ4 ∧ θ5], [θ1 ∧ θ2 ∧ θ3 ∧ θ4 ∧ θ5], [ ]]

```

Example 1
The exact 2-forms
> Exact(RC[2]);

$$[\theta_2 \wedge \theta_5 + \theta_3 \wedge \theta_4, \theta_3 \wedge \theta_5]$$

Example 2
The exact 3-forms
alg > Exact(RC[3]);

$$[\theta_1 \wedge \theta_3 \wedge \theta_5 - \theta_2 \wedge \theta_3 \wedge \theta_4, \theta_2 \wedge \theta_3 \wedge \theta_5, \theta_2 \wedge \theta_4 \wedge \theta_5, \theta_3 \wedge \theta_4 \wedge \theta_5]$$

Example 3
The exact 4-forms
alg > Exact(RC[4]);

$$[\theta_1 \wedge \theta_3 \wedge \theta_4 \wedge \theta_5, \theta_2 \wedge \theta_3 \wedge \theta_4 \wedge \theta_5] \quad (6)$$

Example 4

The exact 5-forms

alg > **Exact**(RC[5]);

$$[] \quad (7)$$

4.2.3 *Calculating Cohomology*

This program calculates the cohomology.

Calling Sequence

Closed(S1)

Parameters

S1 - A list differential forms

Description

- Let S1 be a list of k-forms. JacCohomology calculates the cohomology on the k-forms. It returns a 3 part list. The first part is the cohomology. The second part is the exact forms. The third part is complementary space.

Examples

Note that this is the same example as 2.3 example 1.

```
> DGEEnvironment[LieAlgebra] ("Winternitz", 1, [5, 5], alg);  
Lie algebra: alg (1)
```

```
> MultiplicationTable();
```

alg	e1	e2	e3	e4	e5
e1	0	0	0	0	0
e2	0	0	0	0	e1
e3	0	0	0	e1	e2
e4	0	0	-e1	0	0
e5	0	-e1	-e2	0	0

(2)

```
> RC := RelativeChains([]);  
RC := [[ ], [θ1, θ2, θ3, θ4, θ5], [θ1 ∧ θ2, θ1 ∧ θ3, θ1 ∧ θ4, θ1 ∧ θ5, θ2 ∧ θ3, θ2 ∧ θ4, θ2  
(3)
```

```
∧ θ5, θ3 ∧ θ4, θ3 ∧ θ5, θ4 ∧ θ5], [θ1 ∧ θ2 ∧ θ3, θ1 ∧ θ2 ∧ θ4, θ1 ∧ θ2 ∧ θ5, θ1 ∧ θ3  
∧ θ4, θ1 ∧ θ3 ∧ θ5, θ1 ∧ θ4 ∧ θ5, θ2 ∧ θ3 ∧ θ4, θ2 ∧ θ3 ∧ θ5, θ2 ∧ θ4 ∧ θ5, θ3 ∧ θ4  
∧ θ5], [θ1 ∧ θ2 ∧ θ3 ∧ θ4, θ1 ∧ θ2 ∧ θ3 ∧ θ5, θ1 ∧ θ2 ∧ θ4 ∧ θ5, θ1 ∧ θ3 ∧ θ4 ∧ θ5,  
θ2 ∧ θ3 ∧ θ4 ∧ θ5], [θ1 ∧ θ2 ∧ θ3 ∧ θ4 ∧ θ5], [ ]]
```

```
> JacCohomology(RC[2]);  
[[θ3 ∧ θ4, θ1 ∧ θ5 + θ2 ∧ θ4, θ2 ∧ θ3, θ4 ∧ θ5], [θ2 ∧ θ5 + θ3 ∧ θ4, θ3 ∧ θ5], [θ1 ∧ θ2,  
(4)  
θ1 ∧ θ3, θ1 ∧ θ4, θ2 ∧ θ4]]
```

4.3 *Serre Hofschild Spectral Sequence Programs*

4.3.1 *User Interface Function SHSS*

The main program in SHSSLAWC2. From this program you can create the spectral sequence. You can also calculate additional pages of the sequence.

Calling Sequence

SHSSLAWC2(SubAlgebra, ComplementarySpace, Page, {NumberOfPages = 1, VectorSpace = [], FormRange = [], RelativeChain = [] })

Parameters

- SubAlgebra - The sub algebra that you wish to create the spectral sequence from. Or it is a Serre Hochschild spectral sequence book you wish to manipulate.
- Complementary Space - The complementary space to the subalgebra. This is an optional input.
- Page - A Serre Hochschild spectral sequence book or page. This is an optional input.
- NumberOfPages - The number of pages you wish to calculate
-
- VectorSpace - The Basis of the vector space associated with the Lie algebra with coefficients in a representation.
-
- FormRange - A list [a,b]. This will calculate the spectral sequence using k-forms from degree a to degree b.
-
- Relative Chain - A list of forms that you wish the corresponding answer to be relative to.

Description

- SHSSLAWC2 is the main function for creating and manipulating Serre Hochschild spectral sequences
- By inputting a subalgebra the function will create a spectral sequence and calculate the first page
- Giving a book to the program it will automatically calculate the next page in the book
- Using NumberOfPages you can specify how many additional pages you want the program to calculate.

Examples

```
Example 1
Setting up the sequence
> with (Library) :
> LD:=Retrieve("Winternitz",1,[6,19],alg1);
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3]=e6, [e2, e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0, [e4, e5]=0, [e4, e6]=0, [e5, e6]=0
> DGsetup(LD);
Lie algebra: alg1
```

(1)

(2)


```

> MultiplicationTable();

```

alg1	<i>e1</i>	<i>e2</i>	<i>e3</i>	<i>e4</i>	<i>e5</i>	<i>e6</i>
<i>e1</i>	0	<i>e3</i>	<i>e4</i>	<i>e5</i>	<i>e6</i>	0
<i>e2</i>	- <i>e3</i>	0	<i>e6</i>	0	0	0
<i>e3</i>	- <i>e4</i>	- <i>e6</i>	0	0	0	0
<i>e4</i>	- <i>e5</i>	0	0	0	0	0
<i>e5</i>	- <i>e6</i>	0	0	0	0	0
<i>e6</i>	0	0	0	0	0	0

(3)

```

alg1 > B:=[e4,e5,e6,e1,e2,e3]
          B := [e4, e5, e6, e1, e2, e3]

```

(4)

```

alg1 > DGEEnvironment[LieAlgebra](B,vectorlabels=[x1,x2,x3,y1,
y2,y3],formlabels=[alpha1,alpha2,alpha3,beta1,beta2,
beta3],alg2);
          Lie algebra: alg2

```

(5)

```

alg2 > MultiplicationTable();

```

alg2	<i>x1</i>	<i>x2</i>	<i>x3</i>	<i>y1</i>	<i>y2</i>	<i>y3</i>
<i>x1</i>	0	0	0	- <i>x2</i>	0	0
<i>x2</i>	0	0	0	- <i>x3</i>	0	0
<i>x3</i>	0	0	0	0	0	0
<i>y1</i>	<i>x2</i>	<i>x3</i>	0	0	<i>y3</i>	<i>x1</i>
<i>y2</i>	0	0	0	- <i>y3</i>	0	<i>x3</i>
<i>y3</i>	0	0	0	- <i>x1</i>	- <i>x3</i>	0

(6)

```

alg2 > Alpha:=evalDG([alpha1,alpha2,alpha3]);
          A := [α1, α2, α3]

```

(7)

```

alg2 > Beta1:=ComplementaryBasis(Alpha);
          B1 := [β1, β2, β3]

```

(8)

```

Creating the Serre Hochschild spectral sequence
alg2 > SST := SHSSLAWC2(Alpha,Beta1);
          SST := Book

```

(9)

Example 2

Calculating one additional page

```

alg2 > SHSSLAWC2(SST);
          Book

```

(10)

Example 3

Calculating multiple additional pages at once

```
alg2 > SHSSLAWC2(SST, NumberOfPages=3);  
Book (11)
```

Example 4

Form range

```
alg2 > SST2 := SHSSLAWC2(Alpha, Beta1, FormRange = [1,4]);  
SST2 := Book (12)
```

Example 5

Relative Chains

```
alg2 > RC := evalDG([x1,x2,y1-y2]);  
RC := [x1,x2,y1-y2] (13)
```

```
alg2 > SST3 := SHSSLAWC2(Alpha, Beta1, RelativeChain = RC);  
SST3 := Book (14)
```

Example 6

Constructing a sequence with only the subalgebra

```
alg2 > SHSSLAWC2(Alpha);  
Book (15)
```

4.3.2 Setting up the Spectral Sequence

This function `SpectralSequenceTableLAWC2` sets up the spectral sequence. The sequence is a collection of pages called a book. Each page is set up like the following:

```
[[Page Number, # of Squares, SubalgebraSize, ComplementarySpaceSize] , [ [ [p,q], [PageCohomology] , [Exact], [Complementary]] ] ]
```

The first list inside is a table called page info. Page contains the Page number, number of squares in the page, the size of the sub-algebra, and the size of the complementary space.

The second list is a table for each square. Each square is called by its p,q coordinate. It contains three parts. The first part is the forms that are PageCohomology meaning they are the closed forms minus the exact forms from on page. The exact list is the list of exact forms. The complementary list is the list of forms that are complementary to the closed forms.

Furthermore each form in the three lists is split into a smaller table:

```
Page[i,j][PageCohomology] := table([Forms = [], Chains = [], ChainDerivative = [] ] );
```

```
Page[i,j][Exact] := table([Forms = [], Chains = [], ChainDerivative = [] ] );
```

```
Page[i,j][Complement] := table([Forms = [], Chains = [], ChainDerivative = [] ] );
```

At the start the form and chain for each form will be the same. The chain derivative will just be the exterior derivative of the form. As chains are created the chains will have forms added to it and the chain derivative will have forms added to it. This is so the exterior derivative needs only be calculated once.

Calling Sequence

SpectralSequenceTableLAWC2 := proc(Alpha, Beta, Rep, {VectorSpaceInternal := [], FormRangeInternal := [], RelativeChainInternal := [] })

Parameters

- Alpha - The sub algebra that you wish to create the spectral sequence from.
- Beta - The complementary space to the subalgebra.
- Rep - 0 for Lie algebra, 1 for Lie algebra with coefficients in a representation

VectorSpace - The Basis of the vector space associated with the Lie algebra with coefficients in a representation.

FormRange - A list [a,b]. This will calculate the spectral sequence using k-forms from degree a to degree b.

Relative Chain - A list of forms that you wish the corresponding answer to be relative to.

Description

- SpectralSequenceTableLAWC2 sets up a Serre Hochschild spectral sequence for either a Lie algebra or a Lie algebra with coefficients in a representation.

Examples

```

> with(Library) :
> LD:=Retrieve("Winternitz",1,[6,19],alg1);
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3]=e6,
[e2, e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0, [e4, e5
]=0, [e4, e6]=0, [e5, e6]=0

```

```

> DGsetup(LD);
Lie algebra: alg1

```

> MultiplicationTable();

alg1	e1	e2	e3	e4	e5	e6
e1	0	e3	e4	e5	e6	0
e2	-e3	0	e6	0	0	0
e3	-e4	-e6	0	0	0	0
e4	-e5	0	0	0	0	0
e5	-e6	0	0	0	0	0
e6	0	0	0	0	0	0

(5.3)

alg1 > B:=[e4,e5,e6,e1,e2,e3]

B := [e4, e5, e6, e1, e2, e3]

(5.4)

alg1 > DGEEnvironment[LieAlgebra](B,vectorlabels=[x1,x2,x3,y1,y2,y3],formlabels=[alpha1,alpha2,alpha3,beta1,beta2,beta3],alg2);

Lie algebra: alg2

(5.5)

alg2 > MultiplicationTable();

alg2	x1	x2	x3	y1	y2	y3
x1	0	0	0	-x2	0	0
x2	0	0	0	-x3	0	0
x3	0	0	0	0	0	0
y1	x2	x3	0	0	y3	x1
y2	0	0	0	-y3	0	x3
y3	0	0	0	-x1	-x3	0

(5.6)

alg2 > Alpha:=evalDG([alpha1,alpha2,alpha3]);

A := [α1, α2, α3]

(1)

alg2 > Beta1:=ComplementaryBasis(Alpha);

B1 := [β1, β2, β3]

(2)

Creating the Serre Hochschild spectral sequence

alg2 > SST := SpectralSequenceTableLAWC2(Alpha,Beta1,0);

SST := table([PageInfo=table([Min=-1,X=0,Max=7, Vectors=0, SizeOfSubAlgebra=3,

(3)

SizeOfComplementarySpace=3, NumberOfSquares=16, PageNumber=-1]), (2, 1)

= table([Complement=table([Chains=[], Forms=[], ChainDerivative=[]]),

PageCohomology=table([Chains=[α1 ∧ β1 ∧ β2, α1 ∧ β1 ∧ β3, α1 ∧ β2 ∧ β3, α2 ∧ β1

∧ β2, α2 ∧ β1 ∧ β3, α2 ∧ β2 ∧ β3, α3 ∧ β1 ∧ β2, α3 ∧ β1 ∧ β3, α3 ∧ β2 ∧ β3], Forms

= [α1 ∧ β1 ∧ β2, α1 ∧ β1 ∧ β3, α1 ∧ β2 ∧ β3, α2 ∧ β1 ∧ β2, α2 ∧ β1 ∧ β3, α2 ∧ β2

$\wedge \beta_1 \wedge \beta_2 \wedge \beta_3$], $Forms = [\alpha_1 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3, \alpha_2 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3, \alpha_3 \wedge \beta_1 \wedge \beta_2$
 $\wedge \beta_3]$, $ChainDerivative = [0 \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2, 0 \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2, 0 \alpha_1$
 $\wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2]]$, $Exact = table([Chains = [], Forms = [], ChainDerivative$
 $= []])$), $(1, 0) = table([Complement = table([Chains = [], Forms = [], ChainDerivative$
 $= []])$, $PageCohomology = table([Chains = [\beta_1, \beta_2, \beta_3], Forms = [\beta_1, \beta_2, \beta_3],$
 $ChainDerivative = [0 \alpha_1 \wedge \alpha_2, 0 \alpha_1 \wedge \alpha_2, -\beta_1 \wedge \beta_2])$, $Exact = table([Chains = [],$
 $Forms = [], ChainDerivative = []])$)])

4.3.3 *Calculating the next page of the Spectral Sequence*

This program is work horse of the spectral sequence. It takes a page and calculates the next page. It takes the forms that are still in play and takes their chain derivative. It then calculates the projection of all the chain derivative onto the appropriate square depending on the page. Then the program then solves for the exact space. Then the program makes the appropriate changes. For example it chains forms together if they need to be chained. If a form is exact then it moves it to the exact list and so forth. After every form has either been chained to something, moved to exact or complementary space, or deemed part of the page cohomology the program finishes and returns a page to the book.

Calling Sequence

NextPageTableLAWC2(S1)

Parameters

S1 - A Serre Hochschild spectral sequence page

Description

- NextPageTableLAWC2 takes a Serre Hochschild spectral sequence page and returns the next page.

Examples

Setting up the Lie algebra

```
> with(Library) :
```

```
> LD:=Retrieve("Winternitz",1,[6,19],alg1) ;
```

```
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3]=e6, (5.1)  
      [e2, e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0, [e4, e5]  
      ]=0, [e4, e6]=0, [e5, e6]=0
```

```
> DGsetup(LD) ;
```

Lie algebra: alg1

(5.2)

```
> MultiplicationTable() ;
```

alg1	e1	e2	e3	e4	e5	e6
e1	0	e3	e4	e5	e6	0
e2	-e3	0	e6	0	0	0
e3	-e4	-e6	0	0	0	0
e4	-e5	0	0	0	0	0
e5	-e6	0	0	0	0	0
e6	0	0	0	0	0	0

(5.3)

```
alg1 > B:=[e4,e5,e6,e1,e2,e3]
```

B := [e4, e5, e6, e1, e2, e3]

(5.4)

```
alg1 > DGEEnvironment[LieAlgebra](B,vectorlabels=[x1,x2,x3,  
y1,y2,y3],formlabels=[alpha1,alpha2,alpha3,beta1,  
beta2,beta3],alg2) ;
```

Lie algebra: alg2

(5.5)

```
alg2 > MultiplicationTable() ;
```

alg2	x1	x2	x3	y1	y2	y3
x1	0	0	0	-x2	0	0
x2	0	0	0	-x3	0	0
x3	0	0	0	0	0	0
y1	x2	x3	0	0	y3	x1
y2	0	0	0	-y3	0	x3
y3	0	0	0	-x1	-x3	0

(5.6)

```
alg2 > Alpha:=evalDG([alpha1,alpha2,alpha3]);
```

```
A := [alpha1, alpha2, alpha3]
```

(1)

```
alg2 > Beta1:=ComplementaryBasis(Alpha);
```

```
B1 := [beta1, beta2, beta3]
```

(2)

Creating the Serre Hochschild spectral sequence

```
alg2 > SST := SpectralSequenceTableLAWC2(Alpha,Beta1,0):
```

The -1 page. This is the set up for the sequence.

```
alg2 > ReadTable(SST);
```

```
table([PageInfo=table([SizeOfComplementarySpace=3, SizeOfSubAlgebra=3,
```

(3)

```
NumberOfSquares=16, PageNumber=-1]), (1,3)=[alpha1^alpha2^alpha3^beta1, alpha1^alpha2^alpha3^beta2, alpha1^alpha2^alpha3^beta3], (0,2)=[alpha1^alpha2, alpha1^alpha3, alpha2^alpha3], (0,1)=[alpha1, alpha2, alpha3], (0,0)=[ ], (2,1)=[alpha1^beta1^beta2, alpha1^beta1^beta3, alpha1^beta2^beta3, alpha2^beta1^beta2, alpha2^beta1^beta3, alpha2^beta2^beta3, alpha3^beta1^beta2, alpha3^beta1^beta3, alpha3^beta2^beta3], (3,2)=[alpha1^alpha2^beta1^beta2^beta3, alpha1^alpha3^beta1^beta2^beta3, alpha2^alpha3^beta1^beta2^beta3], (2,2)=[alpha1^alpha2^beta1^beta2, alpha1^alpha2^beta1^beta3, alpha1^alpha2^beta2^beta3, alpha1^alpha3^beta1^beta2, alpha1^alpha3^beta1^beta3, alpha1^alpha3^beta2^beta3, alpha2^alpha3^beta1^beta2, alpha2^alpha3^beta1^beta3, alpha2^alpha3^beta2^beta3], (2,3)=[alpha1^alpha2^alpha3^beta1^beta2, alpha1^alpha2^alpha3^beta1^beta3, alpha1^alpha2^alpha3^beta2^beta3], (3,0)=[beta1^beta2^beta3], (1,2)=[alpha1^alpha2^beta1, alpha1^alpha2^beta2, alpha1^alpha2^beta3, alpha1^alpha3^beta1, alpha1^alpha3^beta2, alpha1^alpha3^beta3, alpha2^alpha3^beta1, alpha2^alpha3^beta2, alpha2^alpha3^beta3], (2,0)=[beta1^beta2, beta1^beta3, beta2^beta3], (1,1)=[alpha1^beta1, alpha1^beta2, alpha1^beta3, alpha2^beta1, alpha2^beta2, alpha2^beta3, alpha3^beta1, alpha3^beta2, alpha3^beta3], (3,3)=[alpha1^alpha2^alpha3^beta1^beta2^beta3], (0,3)=[alpha1^alpha2^alpha3], (1,0)=[beta1, beta2, beta3], (3,1)=[alpha1^beta1^beta2^beta3, alpha2^beta1^beta2^beta3, alpha3^beta1^beta2^beta3]])
```

Calculating page 0

```
alg2 > Page0 := NextPageTableLAWC(SST):
```

The forms on page 0

alg2 > ReadTable (Page0) ;

table([PageInfo = table([SizeOfComplementarySpace = 3, SizeOfSubAlgebra = 3, (4)
NumberOfSquares = 16, PageNumber = 0]), (1, 3) = [$\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1, \alpha_1 \wedge \alpha_2 \wedge \alpha_3$
 $\wedge \beta_2, \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_3], (0, 2) = [\alpha_2 \wedge \alpha_3, \alpha_1 \wedge \alpha_3, \alpha_1 \wedge \alpha_2], (0, 1) = [\alpha_3, \alpha_2, \alpha_1],$
(0, 0) = [], (2, 1) = [$\alpha_1 \wedge \beta_1 \wedge \beta_3, \alpha_1 \wedge \beta_1 \wedge \beta_2, \alpha_3 \wedge \beta_2 \wedge \beta_3, \alpha_3 \wedge \beta_1 \wedge \beta_3, \alpha_3 \wedge \beta_1$
 $\wedge \beta_2, \alpha_2 \wedge \beta_2 \wedge \beta_3, \alpha_2 \wedge \beta_1 \wedge \beta_3, \alpha_2 \wedge \beta_1 \wedge \beta_2, \alpha_1 \wedge \beta_2 \wedge \beta_3], (3, 2) = [\alpha_2 \wedge \alpha_3 \wedge \beta_1$
 $\wedge \beta_2 \wedge \beta_3, \alpha_1 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3, \alpha_1 \wedge \alpha_2 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3], (2, 2) = [\alpha_2 \wedge \alpha_3 \wedge \beta_2$
 $\wedge \beta_3, \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_3, \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2, \alpha_1 \wedge \alpha_3 \wedge \beta_2 \wedge \beta_3, \alpha_1 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_3,$
 $\alpha_1 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2, \alpha_1 \wedge \alpha_2 \wedge \beta_2 \wedge \beta_3, \alpha_1 \wedge \alpha_2 \wedge \beta_1 \wedge \beta_3, \alpha_1 \wedge \alpha_2 \wedge \beta_1 \wedge \beta_2], (2, 3)$
= [$\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2, \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_3, \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_2 \wedge \beta_3], (3, 0)$
= [$\beta_1 \wedge \beta_2 \wedge \beta_3], (1, 2) = [\alpha_2 \wedge \alpha_3 \wedge \beta_2, \alpha_2 \wedge \alpha_3 \wedge \beta_1, \alpha_1 \wedge \alpha_3 \wedge \beta_3, \alpha_1 \wedge \alpha_3 \wedge \beta_2,$
 $\alpha_1 \wedge \alpha_3 \wedge \beta_1, \alpha_1 \wedge \alpha_2 \wedge \beta_3, \alpha_1 \wedge \alpha_2 \wedge \beta_2, \alpha_1 \wedge \alpha_2 \wedge \beta_1, \alpha_2 \wedge \alpha_3 \wedge \beta_3], (2, 0) = [\beta_2$
 $\wedge \beta_3, \beta_1 \wedge \beta_3, \beta_1 \wedge \beta_2], (1, 1) = [\alpha_3 \wedge \beta_3, \alpha_3 \wedge \beta_2, \alpha_3 \wedge \beta_1, \alpha_2 \wedge \beta_3, \alpha_2 \wedge \beta_2, \alpha_2$
 $\wedge \beta_1, \alpha_1 \wedge \beta_3, \alpha_1 \wedge \beta_2, \alpha_1 \wedge \beta_1], (3, 3) = [\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3], (0, 3) = [\alpha_1$
 $\wedge \alpha_2 \wedge \alpha_3], (1, 0) = [\beta_3, \beta_2, \beta_1], (3, 1) = [\alpha_3 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3, \alpha_2 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3, \alpha_1$
 $\wedge \beta_1 \wedge \beta_2 \wedge \beta_3]]]$

Calculating page 1

alg2 > Page1 := NextPageTableLAWC (Page0) :

The forms that are on page 1

alg2 > ReadTable (Page1) ;

table([PageInfo = table([SizeOfComplementarySpace = 3, SizeOfSubAlgebra = 3, (5)
NumberOfSquares = 16, PageNumber = 1]), (1, 3) = [$\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_2, \alpha_1 \wedge \alpha_2 \wedge \alpha_3$
 $\wedge \beta_1], (0, 2) = [\alpha_1 \wedge \alpha_2], (0, 1) = [\alpha_1], (0, 0) = [], (2, 1) = [\alpha_1 \wedge \beta_2 \wedge \beta_3, \alpha_2 \wedge \beta_1$
 $\wedge \beta_3, \alpha_3 \wedge \beta_1 \wedge \beta_3], (3, 2) = [\alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3], (2, 2) = [\alpha_1 \wedge \alpha_2 \wedge \beta_2 \wedge \beta_3,$
 $\alpha_1 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_3, \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_3], (2, 3) = [\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_2 \wedge \beta_3, \alpha_1 \wedge \alpha_2$
 $\wedge \alpha_3 \wedge \beta_1 \wedge \beta_3], (3, 0) = [\beta_1 \wedge \beta_2 \wedge \beta_3], (1, 2) = [\alpha_1 \wedge \alpha_2 \wedge \beta_2, \alpha_1 \wedge \alpha_2 \wedge \beta_3 - \alpha_1$
 $\wedge \alpha_3 \wedge \beta_2, \alpha_2 \wedge \alpha_3 \wedge \beta_1], (2, 0) = [\beta_1 \wedge \beta_3, \beta_2 \wedge \beta_3], (1, 1) = [\alpha_1 \wedge \beta_2, \alpha_1 \wedge \beta_3 - \alpha_2$
 $\wedge \beta_2, \alpha_3 \wedge \beta_1], (3, 3) = [\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3], (0, 3) = [\alpha_1 \wedge \alpha_2 \wedge \alpha_3], (1,$
 $0) = [\beta_2, \beta_1], (3, 1) = [\alpha_3 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3]]]$

4.4 *Serre Hofschild Spectral Sequence Toolkit*

4.4.1 *Read Table*

This program takes a page and returns a list of all the forms in each square.

Calling Sequence

ReadTable(S1)

Parameters

S1 - A Serre Hochschild spectral sequence page

Description

- Let S1 be a Serre Hochschild spectral sequence page. ReadTable returns a list of forms that are still on the page in each square of the page. It also returns the page info containing the page number, size of the subalgebra used to calculate the sequence, the size of the complementary space of the sub algebra, and the number of squares on the page.

Examples

Setting up the Lie algebra

```
> with(Library) :
```

```
> LD:=Retrieve("Winternitz",1,[6,19],alg1) ;
```

```
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3]=e6,
[e2, e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0, [e4, e5]
]=0, [e4, e6]=0, [e5, e6]=0
```

(5.1)

```
> DGsetup(LD) ;
```

Lie algebra: alg1

(5.2)

```
> MultiplicationTable() ;
```

alg1	e1	e2	e3	e4	e5	e6
e1	0	e3	e4	e5	e6	0
e2	-e3	0	e6	0	0	0
e3	-e4	-e6	0	0	0	0
e4	-e5	0	0	0	0	0
e5	-e6	0	0	0	0	0
e6	0	0	0	0	0	0

(5.3)

```
alg1 > B:=[e4,e5,e6,e1,e2,e3]
```

B := [e4, e5, e6, e1, e2, e3]

(5.4)

```
alg1 > DGEEnvironment[LieAlgebra](B,vectorlabels=[x1,x2,x3,
y1,y2,y3],formlabels=[alpha1,alpha2,alpha3,beta1,
beta2,beta3],alg2) ;
```

Lie algebra: alg2

(5.5)

```
alg2 > MultiplicationTable();
```

alg2	$x1$	$x2$	$x3$	$y1$	$y2$	$y3$
$x1$	0	0	0	$-x2$	0	0
$x2$	0	0	0	$-x3$	0	0
$x3$	0	0	0	0	0	0
$y1$	$x2$	$x3$	0	0	$y3$	$x1$
$y2$	0	0	0	$-y3$	0	$x3$
$y3$	0	0	0	$-x1$	$-x3$	0

(5.6)

```
alg2 > Alpha:=evalDG([alpha1,alpha2,alpha3]);
```

$A := [\alpha1, \alpha2, \alpha3]$

(1)

```
alg2 > Beta1:=ComplementaryBasis(Alpha);
```

$B1 := [\beta1, \beta2, \beta3]$

(2)

Creating the Serre Hochschild spectral sequence

```
> SST := SHSSLAWC2(Alpha,Beta1);
```

$SST := Book$

(3)

```
alg2 > SHSSLAWC2(SST);
```

$Book$

(4)

```
alg2 > ReadTable(SST[2]);
```

$table([(3,3) = [\alpha1 \wedge \alpha2 \wedge \alpha3 \wedge \beta1 \wedge \beta2 \wedge \beta3], (3,0) = [], (3,2) = [], (0,2) = [], (0,0) = [], (1,1) = [\alpha1 \wedge \beta2 + \alpha3 \wedge \beta1, -\alpha1 \wedge \beta3 + \alpha2 \wedge \beta2], PageInfo = table([PageNumber = 2, NumberOfSquares = 16, SizeOfSubAlgebra = 3, SizeOfComplementarySpace = 3]), (2,2) = [\alpha1 \wedge \alpha2 \wedge \beta2 \wedge \beta3, \alpha1 \wedge \alpha3 \wedge \beta1 \wedge \beta3], (0,3) = [], (1,3) = [\alpha1 \wedge \alpha2 \wedge \alpha3 \wedge \beta1], (0,1) = [], (2,3) = [\alpha1 \wedge \alpha2 \wedge \alpha3 \wedge \beta2 \wedge \beta3, \alpha1 \wedge \alpha2 \wedge \alpha3 \wedge \beta1 \wedge \beta3], (2,1) = [\alpha1 \wedge \beta2 \wedge \beta3, \alpha3 \wedge \beta1 \wedge \beta3], (2,0) = [\beta2 \wedge \beta3], (1,2) = [\alpha2 \wedge \alpha3 \wedge \beta1, \alpha1 \wedge \alpha2 \wedge \beta2], (3,1) = [], (1,0) = [\beta1, \beta2]])$

(5)

4.4.2 *Read Exact*

This program takes a page and returns all of the exact forms in each square.

Calling Sequence

ReadExact(S1)

Parameters

S1 - A Serre Hochschild spectral sequence page

Description

- Let S1 be a Serre Hochschild spectral sequence page. ReadExact returns a list of forms that for the exact space in each square of the page. It also returns the page info containing the page number, size of the subalgebra used to calculate the sequence, the size of the complementary space of the sub algebra, and the number of squares on the page.

Examples

Setting up the Lie algebra

```
> with(Library) :
```

```
> LD:=Retrieve("Winternitz",1,[6,19],alg1) ;
```

```
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3]=e6, (5.1)  
[e2, e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0, [e4, e5]  
]=0, [e4, e6]=0, [e5, e6]=0
```

```
> DGsetup(LD) ;
```

Lie algebra: alg1 (5.2)

```
> MultiplicationTable() ;
```

alg1	e1	e2	e3	e4	e5	e6
e1	0	e3	e4	e5	e6	0
e2	-e3	0	e6	0	0	0
e3	-e4	-e6	0	0	0	0
e4	-e5	0	0	0	0	0
e5	-e6	0	0	0	0	0
e6	0	0	0	0	0	0

(5.3)

```
alg1 > B:=[e4,e5,e6,e1,e2,e3]
```

```
B := [e4, e5, e6, e1, e2, e3]
```

(5.4)

```
alg1 > DGEEnvironment[LieAlgebra](B,vectorlabels=[x1,x2,x3,  
y1,y2,y3],formlabels=[alpha1,alpha2,alpha3,beta1,  
beta2,beta3],alg2) ;
```

Lie algebra: alg2

(5.5)

alg2 > MultiplicationTable();

alg2	$x1$	$x2$	$x3$	$y1$	$y2$	$y3$
$x1$	0	0	0	$-x2$	0	0
$x2$	0	0	0	$-x3$	0	0
$x3$	0	0	0	0	0	0
$y1$	$x2$	$x3$	0	0	$y3$	$x1$
$y2$	0	0	0	$-y3$	0	$x3$
$y3$	0	0	0	$-x1$	$-x3$	0

(5.6)

alg2 > Alpha:=evalDG([alpha1,alpha2,alpha3]);

$A := [\alpha1, \alpha2, \alpha3]$

(1)

alg2 > Beta1:=ComplementaryBasis(Alpha);

$B1 := [\beta1, \beta2, \beta3]$

(2)

Creating the Serre Hochschild spectral sequence

> SST := SHSSLAWC2(Alpha,Beta1);

$SST := Book$

(3)

alg2 > SHSSLAWC2(SST);

$Book$

(4)

alg2 > ReadExact(SST[2]);

$table([(2,0) = [-\beta1 \wedge \beta2, -\beta1 \wedge \beta3], (3,2) = [-\alpha1 \wedge \alpha3 \wedge \beta1 \wedge \beta2 \wedge \beta3, -\alpha1 \wedge \alpha2 \wedge \beta1 \wedge \beta2 \wedge \beta3, \alpha2 \wedge \alpha3 \wedge \beta1 \wedge \beta2 \wedge \beta3], (3,0) = [\beta1 \wedge \beta2 \wedge \beta3], (1,1) = [\alpha2 \wedge \beta1, \alpha1 \wedge \beta1], (1,0) = [], (3,3) = [], (2,2) = [-\alpha1 \wedge \alpha2 \wedge \beta1 \wedge \beta3 - \alpha1 \wedge \alpha3 \wedge \beta1 \wedge \beta2, -\alpha1 \wedge \alpha2 \wedge \beta1 \wedge \beta2, -\alpha1 \wedge \alpha3 \wedge \beta1 \wedge \beta3 - \alpha2 \wedge \alpha3 \wedge \beta1 \wedge \beta2, -\alpha1 \wedge \alpha3 \wedge \beta1 \wedge \beta2, -\alpha1 \wedge \alpha2 \wedge \beta2 \wedge \beta3 - \alpha2 \wedge \alpha3 \wedge \beta1 \wedge \beta3], (0,0) = [], (0,2) = [], (3,1) = [\alpha1 \wedge \beta1 \wedge \beta2 \wedge \beta3, \alpha2 \wedge \beta1 \wedge \beta2 \wedge \beta3, -\alpha3 \wedge \beta1 \wedge \beta2 \wedge \beta3], (2,3) = [\alpha1 \wedge \alpha2 \wedge \alpha3 \wedge \beta1 \wedge \beta2], PageInfo = table([SizeOfSubAlgebra = 3, PageNumber = 2, SizeOfComplementarySpace = 3, NumberOfSquares = 16]), (1,3) = [], (2,1) = [\alpha1 \wedge \beta1 \wedge \beta3 + \alpha2 \wedge \beta1 \wedge \beta2, \alpha1 \wedge \beta1 \wedge \beta2, \alpha2 \wedge \beta1 \wedge \beta3 + \alpha3 \wedge \beta1 \wedge \beta2, \alpha2 \wedge \beta1 \wedge \beta2, -\alpha2 \wedge \beta1 \wedge \beta3], (1,2) = [-\alpha1 \wedge \alpha3 \wedge \beta1, -\alpha1 \wedge \alpha2 \wedge \beta1], (0,3) = [], (0,1) = []])$

(5)

4.4.3 *Read Complement*

This program takes a page and returns all of the forms in the complement of the closed space in each square.

Calling Sequence

ReadComplement(S1)

Parameters

S1 - A Serre Hochschild spectral sequence page

Description

- Let S1 be a Serre Hochschild spectral sequence page. ReadExact returns a list of forms that for the complement space in each square of the page. It also returns the page info containing the page number, size of the subalgebra used to calculate the sequence, the size of the complementary space of the sub algebra, and the number of squares on the page.

Examples

Setting up the Lie algebra

```
> with(Library) :
```

```
> LD:=Retrieve("Winternitz",1,[6,19],alg1) ;
```

```
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3]=e6, (5.1)  
      [e2, e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0, [e4, e5]  
      ]=0, [e4, e6]=0, [e5, e6]=0
```

```
> DGsetup(LD) ;
```

Lie algebra: alg1

(5.2)

```
> MultiplicationTable() ;
```

alg1	e1	e2	e3	e4	e5	e6
e1	0	e3	e4	e5	e6	0
e2	-e3	0	e6	0	0	0
e3	-e4	-e6	0	0	0	0
e4	-e5	0	0	0	0	0
e5	-e6	0	0	0	0	0
e6	0	0	0	0	0	0

(5.3)

```
alg1 > B:=[e4,e5,e6,e1,e2,e3]
```

```
      B := [e4, e5, e6, e1, e2, e3]
```

(5.4)

```
alg1 > DGEEnvironment[LieAlgebra](B,vectorlabels=[x1,x2,x3,  
      y1,y2,y3],formlabels=[alpha1,alpha2,alpha3,beta1,
```

```
beta2,beta3],alg2);
```

Lie algebra: alg2

(5.5)

```
alg2 > MultiplicationTable();
```

alg2	x1	x2	x3	y1	y2	y3
x1	0	0	0	-x2	0	0
x2	0	0	0	-x3	0	0
x3	0	0	0	0	0	0
y1	x2	x3	0	0	y3	x1
y2	0	0	0	-y3	0	x3
y3	0	0	0	-x1	-x3	0

(5.6)

```
alg2 > Alpha:=evalDG([alpha1,alpha2,alpha3]);
```

$A := [\alpha_1, \alpha_2, \alpha_3]$

(1)

```
alg2 > Beta1:=ComplementaryBasis(Alpha);
```

$B1 := [\beta_1, \beta_2, \beta_3]$

(2)

Creating the Serre Hochschild spectral sequence

```
> SST := SHSSLAWC2(Alpha,Beta1);
```

$SST := Book$

(3)

```
alg2 > SHSSLAWC2(SST);
```

$Book$

(4)

```
alg2 > ReadComplement(SST[2]);
```

$table([(3, 2) = [], (3, 0) = [], (0, 3) = [\alpha_1 \wedge \alpha_2 \wedge \alpha_3], (0, 2) = [\alpha_1 \wedge \alpha_2], (2, 1) = [], (3, 3) = [], (1, 1) = [\alpha_1 \wedge \beta_2], (0, 1) = [\alpha_1], (1, 3) = [\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_2], (2, 3) = [], (3, 1) = [], (2, 2) = [], PageInfo = table([SizeOfSubAlgebra = 3, NumberOfSquares = 16, PageNumber = 2, SizeOfComplementarySpace = 3]), (0, 0) = [], (1, 2) = [\alpha_1 \wedge \alpha_2 \wedge \beta_3 - \alpha_1 \wedge \alpha_3 \wedge \beta_2], (2, 0) = [], (1, 0) = []])$

(5)

4.4.4 *Read Cohomology*

This program takes a page and reads off all of the forms on the page. Instead of reporting them in squares it does it by degrees.

Calling Sequence

ReadCohomology(S1)

Parameters

S1 - A Serre Hochschild spectral sequence page

Description

- Let S1 be a Serre Hochschild spectral sequence page. ReadCohomology returns a list of forms that are on the page. If the sequence has converged to the cohomology this will be the cohomology of the Lie algebra. It also returns the page info containing the page number, size of the subalgebra used to calculate the sequence, the size of the complementary space of the sub algebra, and the number of squares on the page.

Examples

Setting up the Lie algebra

```
> with(Library) :
```

```
> LD:=Retrieve("Winternitz",1,[6,19],alg1) ;
```

```
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3]=e6, (5.1)  
      [e2, e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0, [e4, e5]  
      ]=0, [e4, e6]=0, [e5, e6]=0
```

```
> DGsetup(LD) ;
```

Lie algebra: alg1 (5.2)

```
> MultiplicationTable() ;
```

alg1	<i>e1</i>	<i>e2</i>	<i>e3</i>	<i>e4</i>	<i>e5</i>	<i>e6</i>
<i>e1</i>	0	<i>e3</i>	<i>e4</i>	<i>e5</i>	<i>e6</i>	0
<i>e2</i>	- <i>e3</i>	0	<i>e6</i>	0	0	0
<i>e3</i>	- <i>e4</i>	- <i>e6</i>	0	0	0	0
<i>e4</i>	- <i>e5</i>	0	0	0	0	0
<i>e5</i>	- <i>e6</i>	0	0	0	0	0
<i>e6</i>	0	0	0	0	0	0

(5.3)

```
alg1 > B:=[e4,e5,e6,e1,e2,e3]
```

```
B := [e4, e5, e6, e1, e2, e3] (5.4)
```

```
alg1 > DGEEnvironment[LieAlgebra](B,vectorlabels=[x1,x2,x3,  
y1,y2,y3],formlabels=[alpha1,alpha2,alpha3,beta1,  
beta2,beta3],alg2) ;
```

(5.5)

Lie algebra: alg2

(5.5)

```
alg2 > MultiplicationTable();
```

alg2	x1	x2	x3	y1	y2	y3
x1	0	0	0	-x2	0	0
x2	0	0	0	-x3	0	0
x3	0	0	0	0	0	0
y1	x2	x3	0	0	y3	x1
y2	0	0	0	-y3	0	x3
y3	0	0	0	-x1	-x3	0

(5.6)

```
alg2 > Alpha:=evalDG([alpha1,alpha2,alpha3]);
```

A := [α_1 , α_2 , α_3]

(1)

```
alg2 > Beta1:=ComplementaryBasis(Alpha);
```

B1 := [β_1 , β_2 , β_3]

(2)

Creating the Serre Hochschild spectral sequence

```
> SST := SHSSLAWC2(Alpha,Beta1);
```

SST := Book

(3)

```
alg2 > SHSSLAWC2(SST);
```

Book

(4)

```
alg2 > ReadCohomology(SST[2]);
```

```
table([0 = [], 1 = [ $\beta_1$ ,  $\beta_2$ ], 2 = [ $\alpha_1 \wedge \beta_2 + \alpha_3 \wedge \beta_1$ ,  $\alpha_1 \wedge \beta_3 - \alpha_2 \wedge \beta_2$ ,  $\beta_2 \wedge \beta_3$ ], 3 = [ $\alpha_2 \wedge \alpha_3 \wedge \beta_1 - \alpha_3 \wedge \beta_2 \wedge \beta_3$ ,  $\alpha_1 \wedge \alpha_2 \wedge \beta_2 - \alpha_3 \wedge \beta_2 \wedge \beta_3$ ,  $\alpha_3 \wedge \beta_1 \wedge \beta_3$ ,  $\alpha_1 \wedge \beta_2 \wedge \beta_3$ ], 4 = [ $\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 - \alpha_1 \wedge \alpha_3 \wedge \beta_2 \wedge \beta_3$ ,  $\alpha_1 \wedge \alpha_2 \wedge \beta_2 \wedge \beta_3$ ,  $\alpha_1 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_3$ ], 5 = [ $\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_2 \wedge \beta_3$ ,  $\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_3$ ], 6 = [ $\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3$ ], PageInfo = table([PageNumber = 2, SizeOfComplementarySpace = 3, SizeOfSubAlgebra = 3, NumberOfSquares = 16])])
```

(5)

4.4.5 ZigZag

Zigzag is an external program. It allows you to calculate the chain of a form in the spectral sequence.

Calling Sequence

Zigzag(Sigma, Page)

Parameters

Sigma - The differential form you wish to find the chain of.

The page that you wish to calculate the Serre Hochschild chain from.

Page

Description

- Zigzag calculates the chain on a form in the Serre Hochschild spectral sequence.

Examples

Example 1

Setting up the Lie algebra

```
> with(Library):
```

```
> LD:=Retrieve("Winternitz",1,[6,19],alg1):
```

```
> DGsetup(LD);
```

Lie algebra: alg1

```
> MultiplicationTable();
```

alg1	$e1$	$e2$	$e3$	$e4$	$e5$	$e6$
$e1$	0	$e3$	$e4$	$e5$	$e6$	0
$e2$	$-e3$	0	$e6$	0	0	0
$e3$	$-e4$	$-e6$	0	0	0	0
$e4$	$-e5$	0	0	0	0	0
$e5$	$-e6$	0	0	0	0	0
$e6$	0	0	0	0	0	0

(1)

```
alg1 > B:=[e4,e5,e6,e1,e2,e3]
```

B := [e4, e5, e6, e1, e2, e3]

(3)

```
alg1 > DGEEnvironment[LieAlgebra](B,vectorlabels=[x1,x2,x3,y1,  
y2,y3],formlabels=[alpha1,alpha2,alpha3,beta1,beta2,  
beta3],alg2);
```

Lie algebra: alg2

(4)

```
alg2 > MultiplicationTable();
```

alg2	$x1$	$x2$	$x3$	$y1$	$y2$	$y3$
$x1$	0	0	0	$-x2$	0	0
$x2$	0	0	0	$-x3$	0	0
$x3$	0	0	0	0	0	0
$y1$	$x2$	$x3$	0	0	$y3$	$x1$
$y2$	0	0	0	$-y3$	0	$x3$
$y3$	0	0	0	$-x1$	$-x3$	0

(5)

```
alg2 > Alpha:=evalDG([alpha1,alpha2,alpha3]);
      A := [alpha1, alpha2, alpha3]
```

(6)

```
alg2 > Beta1:=ComplementaryBasis(Alpha);
      B1 := [beta1, beta2, beta3]
```

(7)

Creating the sequence

```
alg2 > SHSS := SHSSLAWC2(Alpha, Beta1, NumberOfPages=3);
      SHSS := Book
```

(8)

```
alg2 > testforms := eval(SHSS[2][1,2][PageCohomology][Forms]);
      testforms := [alpha2 ^ alpha3 ^ beta1, alpha1 ^ alpha2 ^ beta2]
```

(9)

Creating a form to test the function

```
alg2 > Gamma := evalDG(2 * testforms[1] - testforms[2]);
      Gamma := -alpha1 ^ alpha2 ^ beta2 + 2 * alpha2 ^ alpha3 ^ beta1
```

(10)

Calculating the chain of Gamma on the second page.

```
alg2 > GammaZ := ZigZag(Gamma, SHSS[2]);
      GammaZ := -alpha1 ^ alpha2 ^ beta2 + 2 * alpha2 ^ alpha3 ^ beta1 - alpha3 ^ beta2 ^ beta3
```

(11)

4.4.6 *SSprojection*

SSprojection is an external program. It allows you to calculate the projection of the external derivative of a form into a square.

Calling Sequence

SSProjection(Sigma,P,Q, SOSA)

Parameters

- Sigma - A differential form
- P - The number of elements of the complementary space in the form
-
- Q - The number of elements of the subalgebra in the form
-
- Size of the sub algebra
- SOSA -

Description

- Takes a differential form and returns that part of the form that is in the square corresponding to P and Q.

Examples

```
[Example 1
setting up the Lie algebra
> with(Library) :
> LD:=Retrieve("Winternitz",1,[6,19],alg1);
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3]=e6, [e2, e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0, [e4, e5]=0, [e4, e6]=0, [e5, e6]=0 (1)
> DGsetup(LD);
Lie algebra: alg1 (2)
> MultiplicationTable(); (3)
```

alg1	$e1$	$e2$	$e3$	$e4$	$e5$	$e6$
$e1$	0	$e3$	$e4$	$e5$	$e6$	0
$e2$	$-e3$	0	$e6$	0	0	0
$e3$	$-e4$	$-e6$	0	0	0	0
$e4$	$-e5$	0	0	0	0	0
$e5$	$-e6$	0	0	0	0	0
$e6$	0	0	0	0	0	0

(3)

```
alg1 > B:=[e4,e5,e6,e1,e2,e3]
      B := [e4, e5, e6, e1, e2, e3]
```

(4)

```
alg1 > DGEnvironment[LieAlgebra](B, vectorlabels=[x1,x2,x3,y1,
      y2,y3], formlabels=[alpha1,alpha2,alpha3,beta1,beta2,
      beta3], alg2);
      Lie algebra: alg2
```

(5)

```
alg2 > MultiplicationTable();
      alg2 | x1 x2 x3 | y1 y2 y3
      ---|---|---|
      x1 | 0 0 0 | -x2 0 0
      x2 | 0 0 0 | -x3 0 0
      x3 | 0 0 0 | 0 0 0
      y1 | x2 x3 0 | 0 y3 x1
      y2 | 0 0 0 | -y3 0 x3
      y3 | 0 0 0 | -x1 -x3 0
```

(6)

```
alg2 > Alpha:=evalDG([alpha1,alpha2,alpha3]);
      A := [alpha1, alpha2, alpha3]
```

(7)

```
alg2 > Beta1:=ComplementaryBasis(Alpha);
      B1 := [beta1, beta2, beta3]
```

(8)

```
Setting up the sequence
alg2 > SHSS := SHSSLAWC2(Alpha, Beta1, NumberOfPages=3);
      SHSS := Book
```

(9)

```
alg2 > testforms := eval(SHSS[2][1,2][PageCohomology][Forms]);
      testforms := [alpha2 ^ alpha3 ^ beta1, alpha1 ^ alpha2 ^ beta2]
```

(10)

```
alg2 > a :=SSProjection(testforms[1], 2,2,3);
      a := alpha2 ^ alpha3 ^ beta1
```

(11)

4.4.7 TCA

TCA is a function that checks if two cohomology classes are the equivalent. It calls Cohomology Decomposition on each form. If the two sets are equivalent then the program returns true, else it returns false.

Calling Sequence

TCA(S1,S2)

Parameters

S1, - two lists of differential forms
S2

Description

- TCA performs a cohomology decomposition on the forms in S1 with respect to S2 and checks to see if the cohomology answers are equivalent. If they are, the program returns a true. If not it returns a false.

Examples

Example 1

Setting up the Lie algebra

```
> LD:=SimpleLieAlgebraData("so(2,2)",so22);  
LD := [e1, e2]=e2, [e1, e3]=-e3, [e1, e4]=0, [e1, e5]=e5, [e1, e6]=-e6, [e2, e3]=e1  
- e4, [e2, e4]=e2, [e2, e5]=0, [e2, e6]=0, [e3, e4]=-e3, [e3, e5]=0, [e3, e6]=0,  
[e4, e5]=e5, [e4, e6]=-e6, [e5, e6]=-e1 - e4
```

```
> DGsetup(LD);
```

Lie algebra: so22

```
so22 > OneForms := DGinformation(so22,"FrameBaseForms");
```

```
OneForms := [theta1, theta2, theta3, theta4, theta5, theta6]
```

```
so22 > Forms:= seq(GenerateForms(OneForms, i), i = 1 .. 6);
```

```
Forms := [theta1, theta2, theta3, theta4, theta5, theta6], [theta1 ^ theta2, theta1 ^ theta3, theta1 ^ theta4, theta1 ^ theta5, theta1 ^ theta6, theta2 ^ theta3, theta2 ^ theta4, theta2 ^ theta5, theta2 ^ theta6, theta3 ^ theta4, theta3 ^ theta5, theta3 ^ theta6, theta4 ^ theta5, theta4 ^ theta6, theta5 ^ theta6], [theta1 ^ theta2 ^ theta3, theta1 ^ theta2 ^ theta4, theta1 ^ theta2 ^ theta5, theta1 ^ theta2 ^ theta6, theta1 ^ theta3 ^ theta4, theta1 ^ theta3 ^ theta5, theta1 ^ theta3 ^ theta6, theta1 ^ theta4 ^ theta5, theta1 ^ theta4 ^ theta6, theta1 ^ theta5 ^ theta6, theta2 ^ theta3 ^ theta4, theta2 ^ theta3 ^ theta5, theta2 ^ theta3 ^ theta6, theta2 ^ theta4 ^ theta5, theta2 ^ theta4 ^ theta6, theta2 ^ theta5 ^ theta6, theta3 ^ theta4 ^ theta5, theta3 ^ theta4 ^ theta6, theta3 ^ theta5 ^ theta6, theta4 ^ theta5 ^ theta6], [theta1 ^ theta2 ^ theta3 ^ theta4, theta1 ^ theta2 ^ theta3 ^ theta5, theta1 ^ theta2 ^ theta3 ^ theta6, theta1 ^ theta2 ^ theta4 ^ theta5, theta1 ^ theta2 ^ theta4 ^ theta6, theta1 ^ theta2 ^ theta5 ^ theta6, theta1 ^ theta3 ^ theta4 ^ theta5, theta1 ^ theta3 ^ theta4 ^ theta6, theta1 ^ theta3 ^ theta5 ^ theta6, theta1 ^ theta4 ^ theta5 ^ theta6, theta2 ^ theta3 ^ theta4 ^ theta5, theta2 ^ theta3 ^ theta4 ^ theta6, theta2 ^ theta3 ^ theta5 ^ theta6, theta2 ^ theta4 ^ theta5 ^ theta6, theta3 ^ theta4 ^ theta5 ^ theta6], [theta1 ^ theta2 ^ theta3 ^ theta4 ^ theta5, theta1 ^ theta2 ^ theta3 ^ theta4 ^ theta6, theta1 ^ theta2 ^ theta3 ^ theta5 ^ theta6, theta1 ^ theta2 ^ theta4 ^ theta5 ^ theta6, theta1 ^ theta3 ^ theta4 ^ theta5 ^ theta6, theta2 ^ theta3 ^ theta4 ^ theta5 ^ theta6], [theta1 ^ theta2 ^ theta3 ^ theta4 ^ theta5 ^ theta6]
```

```
so22 > RC := RelativeChains([]);
```

```
RC := [[], [theta1, theta2, theta3, theta4, theta5, theta6], [theta1 ^ theta2, theta1 ^ theta3, theta1 ^ theta4, theta1 ^ theta5, theta1 ^ theta6, theta2 ^ theta3, theta2 ^ theta4, theta2 ^ theta5, theta2 ^ theta6, theta3 ^ theta4, theta3 ^ theta5, theta3 ^ theta6, theta4 ^ theta5, theta4 ^ theta6, theta5 ^ theta6], [theta1 ^ theta2 ^ theta3, theta1 ^ theta2 ^ theta4, theta1 ^ theta2 ^ theta5, theta1 ^ theta2 ^ theta6, theta1 ^ theta3 ^ theta4, theta1 ^ theta3 ^ theta5, theta1 ^ theta3 ^ theta6, theta1 ^ theta4 ^ theta5, theta1 ^ theta4 ^ theta6, theta1 ^ theta5 ^ theta6, theta2 ^ theta3 ^ theta4, theta2 ^ theta3 ^ theta5, theta2 ^ theta3 ^ theta6, theta2 ^ theta4 ^ theta5, theta2 ^ theta4 ^ theta6, theta2 ^ theta5 ^ theta6, theta3 ^ theta4 ^ theta5, theta3 ^ theta4 ^ theta6, theta3 ^ theta5 ^ theta6, theta4 ^ theta5 ^ theta6]]
```


$\wedge \theta_3 \wedge \theta_6, \theta_1 \wedge \theta_4 \wedge \theta_5, \theta_1 \wedge \theta_4 \wedge \theta_6, \theta_1 \wedge \theta_5 \wedge \theta_6, \theta_2 \wedge \theta_3 \wedge \theta_4, \theta_2 \wedge \theta_3 \wedge \theta_5, \theta_2$
 $\wedge \theta_3 \wedge \theta_6, \theta_2 \wedge \theta_4 \wedge \theta_5, \theta_2 \wedge \theta_4 \wedge \theta_6, \theta_2 \wedge \theta_5 \wedge \theta_6, \theta_3 \wedge \theta_4 \wedge \theta_5, \theta_3 \wedge \theta_4 \wedge \theta_6, \theta_3$
 $\wedge \theta_5 \wedge \theta_6, \theta_4 \wedge \theta_5 \wedge \theta_6], [\theta_1 \wedge \theta_2 \wedge \theta_3 \wedge \theta_4, \theta_1 \wedge \theta_2 \wedge \theta_3 \wedge \theta_5, \theta_1 \wedge \theta_2 \wedge \theta_3 \wedge \theta_6,$
 $\theta_1 \wedge \theta_2 \wedge \theta_4 \wedge \theta_5, \theta_1 \wedge \theta_2 \wedge \theta_4 \wedge \theta_6, \theta_1 \wedge \theta_2 \wedge \theta_5 \wedge \theta_6, \theta_1 \wedge \theta_3 \wedge \theta_4 \wedge \theta_5, \theta_1 \wedge \theta_3$
 $\wedge \theta_4 \wedge \theta_6, \theta_1 \wedge \theta_3 \wedge \theta_5 \wedge \theta_6, \theta_1 \wedge \theta_4 \wedge \theta_5 \wedge \theta_6, \theta_2 \wedge \theta_3 \wedge \theta_4 \wedge \theta_5, \theta_2 \wedge \theta_3 \wedge \theta_4$
 $\wedge \theta_6, \theta_2 \wedge \theta_3 \wedge \theta_5 \wedge \theta_6, \theta_2 \wedge \theta_4 \wedge \theta_5 \wedge \theta_6, \theta_3 \wedge \theta_4 \wedge \theta_5 \wedge \theta_6], [\theta_1 \wedge \theta_2 \wedge \theta_3 \wedge \theta_4$
 $\wedge \theta_5, \theta_1 \wedge \theta_2 \wedge \theta_3 \wedge \theta_4 \wedge \theta_6, \theta_1 \wedge \theta_2 \wedge \theta_3 \wedge \theta_5 \wedge \theta_6, \theta_1 \wedge \theta_2 \wedge \theta_4 \wedge \theta_5 \wedge \theta_6, \theta_1$
 $\wedge \theta_3 \wedge \theta_4 \wedge \theta_5 \wedge \theta_6, \theta_2 \wedge \theta_3 \wedge \theta_4 \wedge \theta_5 \wedge \theta_6], [\theta_1 \wedge \theta_2 \wedge \theta_3 \wedge \theta_4 \wedge \theta_5 \wedge \theta_6], []]$

Calculating the cohomology using Cohomology

so22 > **H := Cohomology(RC[3..5]);**

$H := [[\theta_1 \wedge \theta_5 \wedge \theta_6 + \theta_4 \wedge \theta_5 \wedge \theta_6, -\theta_1 \wedge \theta_2 \wedge \theta_3 + \theta_2 \wedge \theta_3 \wedge \theta_4]]$

(6)

Calculating the cohomology using JacCohomology

so22 > **JH := JacCohomology(Forms[2]);**

$JH := [[\theta_1 \wedge \theta_5 \wedge \theta_6 + \theta_4 \wedge \theta_5 \wedge \theta_6, \theta_1 \wedge \theta_5 \wedge \theta_6 + \theta_2 \wedge \theta_3 \wedge \theta_4], [\theta_1 \wedge \theta_2 \wedge \theta_3 + \theta_1$

(7)

$\wedge \theta_5 \wedge \theta_6 + \theta_2 \wedge \theta_3 \wedge \theta_4 - \theta_4 \wedge \theta_5 \wedge \theta_6, \theta_1 \wedge \theta_2 \wedge \theta_4 + \theta_2 \wedge \theta_5 \wedge \theta_6, \theta_1 \wedge \theta_2$

$\wedge \theta_5, \theta_1 \wedge \theta_3 \wedge \theta_4 - \theta_3 \wedge \theta_5 \wedge \theta_6, \theta_1 \wedge \theta_3 \wedge \theta_6, \theta_1 \wedge \theta_4 \wedge \theta_5 - \theta_2 \wedge \theta_3 \wedge \theta_5, \theta_1$

$\wedge \theta_4 \wedge \theta_6 + \theta_2 \wedge \theta_3 \wedge \theta_6, \theta_2 \wedge \theta_4 \wedge \theta_6, \theta_3 \wedge \theta_4 \wedge \theta_5], [\theta_1 \wedge \theta_2 \wedge \theta_6, \theta_1 \wedge \theta_3 \wedge \theta_5,$

$\theta_2 \wedge \theta_3 \wedge \theta_5, \theta_2 \wedge \theta_3 \wedge \theta_6, \theta_2 \wedge \theta_4 \wedge \theta_5, \theta_2 \wedge \theta_5 \wedge \theta_6, \theta_3 \wedge \theta_4 \wedge \theta_6, \theta_3 \wedge \theta_5 \wedge \theta_6, \theta_4$

$\wedge \theta_5 \wedge \theta_6]]$

Checking if the answers are equivalent.

so22 > **TCA(H[1], JH[1]);**

"true"

(8)

4.5 *Auxiliary Functions*

4.5.1 *SSProjectionInternal*

This is an internal function that find the projection of an exterior derivative of a form into a certain square. It does this by counting the number of alpha's and beta's in the form.

Calling Sequence

SSProjectionInternal(ExteriorDerivativeList,NewP,NewQ,SizeOfAlpha)

Parameters

ExteriorDeriv
ativeList - A list of differential forms

NewP - The number of elements of the complementary space from the square you are projecting to.

NewQ - The number of elements in the subalgebra in each from in the square that you are projecting to.

SizeOfAlpha - Size of the Subalgebra

Description

- SSProjection is an internal program used in calculating the next page.
- It is used on Lie algebras
- It takes a list of differential forms projects them onto the square given by P and Q
- Returns a list of forms from the ExteriorDerivative List that are on the given square.

4.5.2 *SSProjectionInternalLAWC*

This is a very similar program to *SSProjection*. It has the same input and output. It's internal workings a little different because it's dealing with forms in a Lie algebra with coefficients in a representation.

Calling Sequence

SSProjectionInternalLAWC(ExteriorDerivativeList,NewP,NewQ,SizeOfAlpha, X)

Parameters

ExteriorDerivativeList - A list of differential forms

NewP - The number of elements of the complementary space from the square you are projecting to.

NewQ - The number of elements in the subalgebra in each from in the square that you are projecting to.

SizeOfAlpha - Size of the Subalgebra

X - Size of the vector space.

Description

- SSProjection is an internal program used in calculating the next page.
- It is used on Lie algebras with coefficients in a representation.
- It takes a list of differential forms projects them onto the square given by P and Q
- Returns a list of forms from the ExteriorDerivative List that are on the given square.

4.5.3 *CohoDecompSHSS*

This program calculates the decomposition of the projection. It splits the projection into exact parts from the past page, complementary parts, or closed on the page.

Calling Sequence

SHSSDecomp(Omega, Cohomology, CohoSize, ExactForms, ExactSize, CompSpace, CompSize)

Parameters

Omega - A differential form

Cohomology - List of differential forms

CohoSize - Number of forms in Cohomology list

ExactForms - List of differential forms

ExactSize - Number of forms in ExactForms list

CompSpace - List of differential forms

CompSize - Number of forms in CompSpace

Description

- SHSSDecomp is an internal function used to calculate the next page
- It does a decomposition of a form in parts that are exact, part of the cohomology, and part of the comp space
- It does this for forms that are in a Lie algebra
- It returns a three part list. The first part of the list is the part in the cohomology list, the second part is the part that is exactlist, and the third part is the part that is in the compspace.

4.5.4 *CohoDecompSHSSLAWC*

This program is the same as *CohoDecompSHSS* but it deals with Lie algebra with coefficients in a representation.

Calling Sequence

SHSSDecompLAWC(Omega, Cohomology, CohoSize, ExactForms, ExactSize, CompSpace, CompSize, VectorSize)

Parameters

Omega - A differential form

Cohomology - List of differential forms

CohoSize - Number of forms in Cohomology list

ExactForms - List of differential forms

ExactSize - Number of forms in ExactForms list

CompSpace - List of differential forms

CompSize - Number of forms in CompSpace

VectorSize - Dimension of the vector space associated to the Lie algebra with coefficients in a representation.

Description

- SHSSDecomp is an internal function used to calculate the next page
- It does a decomposition of a form in parts that are exact, part of the cohomology, and part of the comp space
- It does this for forms that are in a Lie algebra with coefficients in a representation.
- It returns a three part list. The first part of the list is the part in the cohomology list, the second part is the part that is exactlist, and the third part is the part that is in the compspace.

4.5.5 PQ

This program takes a form and tells you what square it belongs in.

Calling Sequence

PQ(S1,SAD)

Parameters

S1 - differential forms

SAD - Subalgebra dimension

Description

- PQ calculates the number of elements of the subalgebra and elements of complementary space in S1. It returns a two element list.

Examples

```
> with(Library) :  
> LD:=Retrieve("Winternitz",1,[6,19],alg1) ;  
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3]=e6, [e2,  
e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0, [e4, e5]=0, [e4,  
e6]=0, [e5, e6]=0
```

```
> DGsetup(LD) ;  
Lie algebra: alg1
```

```
> MultiplicationTable() ;
```

alg1	e1	e2	e3	e4	e5	e6
e1	0	e3	e4	e5	e6	0
e2	-e3	0	e6	0	0	0
e3	-e4	-e6	0	0	0	0
e4	-e5	0	0	0	0	0
e5	-e6	0	0	0	0	0
e6	0	0	0	0	0	0

```
alg1 > B:=[e4, e5, e6, e1, e2, e3]  
B := [e4, e5, e6, e1, e2, e3]
```

```
alg1 > DGEEnvironment[LieAlgebra](B, vectorlabels=[x1, x2, x3, y1,  
y2, y3], formlabels=[alpha1, alpha2, alpha3, beta1, beta2,  
beta3], alg2) ;  
Lie algebra: alg2
```

```
alg2 > MultiplicationTable();
```

alg2	x1	x2	x3	y1	y2	y3
x1	0	0	0	-x2	0	0
x2	0	0	0	-x3	0	0
x3	0	0	0	0	0	0
y1	x2	x3	0	0	y3	x1
y2	0	0	0	-y3	0	x3
y3	0	0	0	-x1	-x3	0

(6)

Example 1

```
alg2 > testform1 := evalDG(alpha1);
```

testform1 := α_1

(7)

```
alg2 > PQ(testform1,3);
```

[0, 1]

(8)

Example 2

```
alg2 > testform2 := evalDG(beta1);
```

testform2 := β_1

(9)

```
alg2 > PQ(testform2,3);
```

[1, 0]

(10)

Example 3

```
alg2 > testform3 := evalDG(alpha2 &w beta1 &w beta3);
```

testform3 := $\alpha_2 \wedge \beta_1 \wedge \beta_3$

(11)

```
alg2 > PQ(testform3,3);
```

[2, 1]

(12)

Example 4

```
alg2 > testform4 := evalDG(alpha1 &w alpha2);
```

testform4 := $\alpha_1 \wedge \alpha_2$

(13)

```
alg2 > PQ(testform4,3);
```

[0, 2]

(14)

5. CONCLUSION

5.1 Calculating the Cohomology Directly

In this first demonstration I will be testing the direct methods of calculating Cohomology. I will be testing the speed and results against the built in Cohomology program in maple.

Cohomology Calculated Directly

Setting up the Lie Algebra

```
> LD:=SimpleLieAlgebraData("so(2,2)",so22);
LD := [e1, e2]=e2, [e1, e3]=-e3, [e1, e4]=0, [e1, e5]=e5, [e1, e6]=-e6, [e2, e3] (1.1)
      ]=e1 - e4, [e2, e4]=e2, [e2, e5]=0, [e2, e6]=0, [e3, e4]=-e3, [e3, e5]=0, [e3,
      e6]=0, [e4, e5]=e5, [e4, e6]=-e6, [e5, e6]=-e1 - e4
```

```
> DGsetup(LD);
Lie algebra: so22 (1.2)
```

```
alg2 > OneForms := DGinformation(so22, "FrameBaseForms");
OneForms := [theta1, theta2, theta3, theta4, theta5, theta6] (1.3)
```

```
alg2 > Forms:= seq(GenerateForms(OneForms, i), i = 1 ..
6);
Forms := [theta1, theta2, theta3, theta4, theta5, theta6], [theta1 wedge theta2, theta1 wedge theta3, theta1 wedge theta4, theta1 wedge theta5, theta1 wedge theta6, theta2 wedge theta3, (1.4)
theta2 wedge theta4, theta2 wedge theta5, theta2 wedge theta6, theta3 wedge theta4, theta3 wedge theta5, theta3 wedge theta6, theta4 wedge theta5, theta4 wedge theta6, theta5 wedge theta6], [theta1
wedge theta2 wedge theta3, theta1 wedge theta2 wedge theta4, theta1 wedge theta2 wedge theta5, theta1 wedge theta2 wedge theta6, theta1 wedge theta3 wedge theta4, theta1 wedge theta3 wedge theta5,
theta1 wedge theta3 wedge theta6, theta1 wedge theta4 wedge theta5, theta1 wedge theta4 wedge theta6, theta1 wedge theta5 wedge theta6, theta2 wedge theta3 wedge theta4, theta2 wedge theta3
wedge theta5, theta2 wedge theta3 wedge theta6, theta2 wedge theta4 wedge theta5, theta2 wedge theta4 wedge theta6, theta2 wedge theta5 wedge theta6, theta3 wedge theta4 wedge theta5, theta3
wedge theta4 wedge theta6, theta3 wedge theta5 wedge theta6, theta4 wedge theta5 wedge theta6], [theta1 wedge theta2 wedge theta3 wedge theta4, theta1 wedge theta2 wedge theta3 wedge theta5, theta1
wedge theta2 wedge theta3 wedge theta6, theta1 wedge theta2 wedge theta4 wedge theta5, theta1 wedge theta2 wedge theta4 wedge theta6, theta1 wedge theta2 wedge theta5 wedge theta6, theta1 wedge theta3
wedge theta4 wedge theta5, theta1 wedge theta3 wedge theta4 wedge theta6, theta1 wedge theta3 wedge theta5 wedge theta6, theta1 wedge theta4 wedge theta5 wedge theta6, theta2 wedge theta3 wedge theta4
wedge theta5, theta2 wedge theta3 wedge theta4 wedge theta6, theta2 wedge theta3 wedge theta5 wedge theta6, theta2 wedge theta4 wedge theta5 wedge theta6, theta3 wedge theta4 wedge theta5 wedge theta6],
[theta1 wedge theta2 wedge theta3 wedge theta4 wedge theta5, theta1 wedge theta2 wedge theta3 wedge theta4 wedge theta6, theta1 wedge theta2 wedge theta3 wedge theta5 wedge theta6, theta1 wedge theta2
wedge theta3 wedge theta4 wedge theta5 wedge theta6], [theta1 wedge theta2 wedge theta3
wedge theta4 wedge theta5 wedge theta6]
```

```
alg2 > RC := RelativeChains([]);
RC := [[], [theta1, theta2, theta3, theta4, theta5, theta6], [theta1 wedge theta2, theta1 wedge theta3, theta1 wedge theta4, theta1 wedge theta5, theta1 wedge theta6, theta2 (1.5)
wedge theta3, theta2 wedge theta4, theta2 wedge theta5, theta2 wedge theta6, theta3 wedge theta4, theta3 wedge theta5, theta3 wedge theta6, theta4 wedge theta5, theta4 wedge theta6, theta5
wedge theta6], [theta1 wedge theta2 wedge theta3, theta1 wedge theta2 wedge theta4, theta1 wedge theta2 wedge theta5, theta1 wedge theta2 wedge theta6, theta1 wedge theta3 wedge theta4, theta1
wedge theta3 wedge theta5, theta1 wedge theta3 wedge theta6, theta1 wedge theta4 wedge theta5, theta1 wedge theta4 wedge theta6, theta1 wedge theta5 wedge theta6, theta2 wedge theta3 wedge theta4,
theta2 wedge theta3 wedge theta5, theta2 wedge theta3 wedge theta6, theta2 wedge theta4 wedge theta5, theta2 wedge theta4 wedge theta6, theta2 wedge theta5 wedge theta6, theta3 wedge theta4
wedge theta5, theta3 wedge theta4 wedge theta6, theta3 wedge theta5 wedge theta6, theta4 wedge theta5 wedge theta6], [theta1 wedge theta2 wedge theta3 wedge theta4, theta1 wedge theta2 wedge theta3
wedge theta5, theta1 wedge theta2 wedge theta3 wedge theta6, theta1 wedge theta2 wedge theta4 wedge theta5, theta1 wedge theta2 wedge theta4 wedge theta6, theta1 wedge theta2 wedge theta5 wedge theta6,
theta1 wedge theta3 wedge theta4 wedge theta5, theta1 wedge theta3 wedge theta4 wedge theta6, theta1 wedge theta3 wedge theta5 wedge theta6, theta1 wedge theta4 wedge theta5 wedge theta6, theta2
wedge theta3 wedge theta4 wedge theta5, theta2 wedge theta3 wedge theta4 wedge theta6, theta2 wedge theta3 wedge theta5 wedge theta6, theta2 wedge theta4 wedge theta5 wedge theta6, theta3 wedge theta4
wedge theta5 wedge theta6], [theta1 wedge theta2 wedge theta3 wedge theta4 wedge theta5, theta1 wedge theta2 wedge theta3 wedge theta4 wedge theta6, theta1 wedge theta2 wedge theta3 wedge theta5
wedge theta6, theta1 wedge theta2 wedge theta3 wedge theta4 wedge theta5 wedge theta6]
```

$\wedge \theta_6, \theta_1 \wedge \theta_2 \wedge \theta_4 \wedge \theta_5 \wedge \theta_6, \theta_1 \wedge \theta_3 \wedge \theta_4 \wedge \theta_5 \wedge \theta_6, \theta_2 \wedge \theta_3 \wedge \theta_4 \wedge \theta_5 \wedge \theta_6],$
 $[\theta_1 \wedge \theta_2 \wedge \theta_3 \wedge \theta_4 \wedge \theta_5 \wedge \theta_6], []]$

Calculating the cohomology using the current maple software

```
alg2 > H := Cohomology(RC[3..5]);
H := [[ $\theta_1 \wedge \theta_5 \wedge \theta_6 + \theta_4 \wedge \theta_5 \wedge \theta_6, -\theta_1 \wedge \theta_2 \wedge \theta_3 + \theta_2 \wedge \theta_3 \wedge \theta_4$ ]] (1.6)
```

Calculating the cohomology using the program I wrote

The software splits the answer into three parts. The cohomology, the exact forms, and the complementary space.

```
alg2 > JH := JacCohomology(Forms[2]);
JH := [[ $\theta_1 \wedge \theta_5 \wedge \theta_6 + \theta_4 \wedge \theta_5 \wedge \theta_6, \theta_1 \wedge \theta_5 \wedge \theta_6 + \theta_2 \wedge \theta_3 \wedge \theta_4$ ], [ $\theta_1 \wedge \theta_2 \wedge \theta_3$ 
+  $\theta_1 \wedge \theta_5 \wedge \theta_6 + \theta_2 \wedge \theta_3 \wedge \theta_4 - \theta_4 \wedge \theta_5 \wedge \theta_6, \theta_1 \wedge \theta_2 \wedge \theta_4 + \theta_2 \wedge \theta_5 \wedge \theta_6, \theta_1$ 
 $\wedge \theta_2 \wedge \theta_5, \theta_1 \wedge \theta_3 \wedge \theta_4 - \theta_3 \wedge \theta_5 \wedge \theta_6, \theta_1 \wedge \theta_3 \wedge \theta_6, \theta_1 \wedge \theta_4 \wedge \theta_5 - \theta_2 \wedge \theta_3$ 
 $\wedge \theta_5, \theta_1 \wedge \theta_4 \wedge \theta_6 + \theta_2 \wedge \theta_3 \wedge \theta_6, \theta_2 \wedge \theta_4 \wedge \theta_6, \theta_3 \wedge \theta_4 \wedge \theta_5$ ], [ $\theta_1 \wedge \theta_2 \wedge \theta_6, \theta_1$ 
 $\wedge \theta_3 \wedge \theta_5, \theta_2 \wedge \theta_3 \wedge \theta_5, \theta_2 \wedge \theta_3 \wedge \theta_6, \theta_2 \wedge \theta_4 \wedge \theta_5, \theta_2 \wedge \theta_5 \wedge \theta_6, \theta_3 \wedge \theta_4 \wedge \theta_6,$ 
 $\theta_3 \wedge \theta_5 \wedge \theta_6, \theta_4 \wedge \theta_5 \wedge \theta_6$ ]] (1.7)
```

The cohomology that my software calculated

```
so22 > JH[1];
[ $\theta_1 \wedge \theta_5 \wedge \theta_6 + \theta_4 \wedge \theta_5 \wedge \theta_6, \theta_1 \wedge \theta_5 \wedge \theta_6 + \theta_2 \wedge \theta_3 \wedge \theta_4$ ] (1.8)
```

Testing Answer

They have the same dimension

```
so22 > map(nops, JH);
[2, 9, 9] (1.9)
```

```
alg2 > nops(H[1]);
2 (1.10)
```

Now that we know that they have the same dimension, we will test if they are equivalent. To do this I will show that the forms are either the same or differ by an exact form.

This will be done by using the maple command Cohomology Decomposition.

```
so22 > CohomologyDecomposition(JH[1][1], H[1]);
 $\theta_1 \wedge \theta_5 \wedge \theta_6 + \theta_4 \wedge \theta_5 \wedge \theta_6, 0 \theta_1 \wedge \theta_2$  (1.11)
```

```
so22 > CohomologyDecomposition(JH[1][2], H[1]); (1.12)
```

$$\begin{aligned}
 & -\left(\frac{1}{2}\right) \theta_1 \wedge \theta_2 \wedge \theta_3 + \left(\frac{1}{2}\right) \theta_1 \wedge \theta_5 \wedge \theta_6 + \left(\frac{1}{2}\right) \theta_2 \wedge \theta_3 \wedge \theta_4 + \left(\frac{1}{2}\right) \theta_4 \wedge \theta_5 \wedge \theta_6, - \\
 & \left(\frac{1}{2}\right) \theta_1 \wedge \theta_4
 \end{aligned} \tag{1.12}$$

TCA is a program I wrote to check the answer. TCA stands for Testing Cohomology Answer. It runs the CohomologyDecomposition program on the elements of the first set and shows that it is equivalent to the second set.

```
so22 > TCA(JH[1], H[1]);
"true"
(1.13)
```

Time Trials Cohomology

```
> restart;
> LD:=SimpleLieAlgebraData("so(3,2)", so32);
> DGsetup(LD);
Lie algebra: so32
(2.1)
```

```
so43 > x:=DGinformation(so32, "FrameBaseForms");
x := [theta_1, theta_2, theta_3, theta_4, theta_5, theta_6, theta_7, theta_8, theta_9, theta_10]
(2.2)
```

```
alg > C:=RelativeChains(so32);
Time it took maple to solve the problem
alg > st:=time():Cohomology(C[3..5]):time()-st;
8.813
(2.3)
```

```
Time it took my program to solve the problem
so32 > st:=time():JacCohomology(C[3]):time()-st;
2.828
(2.4)
```


5.2 *Calculating the Cohomology using the spectral sequence*

Now I will be demonstrating the Serre Hochschild programs. I will start with a Lie Algebra with coefficients in a representation.

▼ Lie Algebra With Coefficients

> restart;

Setting up the Lie Algebra

```
V > A1:=Matrix([[1,0,0],[0,0,0],[0,0,1]]);
V > A2:=Matrix([[0,-1,0],[0,0,0],[-1,0,0]]);
V > A3:=Matrix([[0,-1,0],[0,0,0],[0,0,0]]);
V > A4:=Matrix([[0,0,0],[0,0,0],[0,1,0]]);
```

$$A1 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A2 := \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

$$A3 := \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A4 := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

(3.1)

```
> A:=[A1,A2,A3,A4];
```

$$A := \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \right]$$

(3.2)

Setting up the Lie algebra

```
> DGEEnvironment[VectorSpace](3,V,vectorlabels=[x1,x2,x3]);
Vector Space: V
```

(3.3)

```
> LD:=LieAlgebraData(A,alg1);DGsetup(LD);
```

```
LD := [e1, e2] = e3, [e1, e3] = e3, [e1, e4] = e4, [e2, e3] = e4, [e2, e4] = 0, [e3, e4] = 0
```

```
Lie algebra: alg1
```

(3.4)

```
alg2 > MultiplicationTable();
```

alg1	e1	e2	e3	e4	
e1	0	e3	e3	e4	
e2	-e3	0	e4	0	
e3	-e3	-e4	0	0	
e4	-e4	0	0	0	

(3.5)

```
alg3 > B:=evalDG([e3,e4,e1,e2]);
        B := [e3,e4,e1,e2]
```

(3.6)

Relabeling and rearranging the vectors to split up the sub algebra and complementary space

```
alg2 > DGEEnvironment[LieAlgebra](B,vectorlabels=[s1,s2,c1,
        c2],formlabels=[alpha1,alpha2,beta1,beta2],alg2);
        Lie algebra: alg2
```

(3.7)

```
alg3:1 > MultiplicationTable();
```

alg2	s1	s2	c1	c2	
s1	0	0	-s1	-s2	
s2	0	0	-s2	0	
c1	s1	s2	0	s1	
c2	s2	0	-s1	0	

(3.8)

```
alg1 > rho:=Representation(alg2,V,A);
```

$$\rho := s1 \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, s2 \rightarrow \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, c1 \rightarrow \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, c2 \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

(3.9)

Creating the Lie algebra with coefficients

The subspace is {g1,g2}

The complementary space is {f1,f2}

The dual to the subspace is gamma1,gamm2

The dual to the complementary space is iota1, iota2

The coefficients are x1, x2, x3

```
alg1 > DGEEnvironment[LieAlgebraWithCoefficients](rho,alg3,
        vectorlabels = [g1, g2, f1, f2], formlabels =
        [gamma1, gamma2, iota1, iota2]);
        Lie algebra with coefficients: alg3
```

(3.10)

Lie Coefficient Table

```
alg2 > MultiplicationTable("LieTableCoefficients");
```

$alg3$	$x1$	$x2$	$x3$	
$g1$	$x1$	$0x1$	$x3$	
$g2$	$-x3$	$-x1$	$0x1$	(3.11)
$f1$	$0x1$	$-x1$	$0x1$	
$f2$	$0x1$	$x3$	$0x1$	

Multiplication Table

```
alg2 > MultiplicationTable();
```

$alg3$	$g1$	$g2$	$f1$	$f2$	
$g1$	0	0	$-g1$	$-g2$	
$g2$	0	0	$-g2$	0	(3.12)
$f1$	$g1$	$g2$	0	$g1$	
$f2$	$g2$	0	$-g1$	0	

The Annihilator to the complementary space should be the dual to the subspace

```
alg3 > Alpha := Annihilator([f1, f2]);
      A := [γ1, γ2] (3.13)
```

The Annihilator to the subspace should be the dual to the complementary space

```
alg3 > Comp := Annihilator([g1, g2]);
      Comp := [ι1, ι2] (3.14)
```

This will allow us to see what page and square the program is working on as it's working.

```
alg3 > infolevel[SHSSLAWC] := 3;
      infolevelSHSSLAWC := 3 (3.15)
```

Calculating the first 4 pages. Page 0 through page 3.

```
alg3 > P := SHSSLAWC2(Alpha, Comp, NumberOfPages=3,
      VectorSpace = [x1, x2, x3]);
Setting up Sequence
working on square 0 0
working on square 0 1
working on square 0 2
working on square 1 0
working on square 1 1
working on square 1 2
```

```

working on square 2 0
working on square 2 1
Calculating Page 0
      Calculating E[0]^(0 0)
      Calculating E[0]^(0 1)
      Calculating E[0]^(0 2)
      Calculating E[0]^(1 0)
      Calculating E[0]^(1 1)
      Calculating E[0]^(1 2)
      Calculating E[0]^(2 0)
      Calculating E[0]^(2 1)
      Calculating E[0]^(2 2)
Calculating Page 1
      Calculating E[1]^(0 0)
      Calculating E[1]^(0 1)
      Calculating E[1]^(0 2)
      Calculating E[1]^(1 0)
      Calculating E[1]^(1 1)
      Calculating E[1]^(1 2)
      Calculating E[1]^(2 0)
      Calculating E[1]^(2 1)
      Calculating E[1]^(2 2)
Calculating Page 2
      Calculating E[2]^(0 0)
      Calculating E[2]^(0 1)
      Calculating E[2]^(0 2)
      Calculating E[2]^(1 0)
      Calculating E[2]^(1 1)
      Calculating E[2]^(1 2)
      Calculating E[2]^(2 0)
      Calculating E[2]^(2 1)
      Calculating E[2]^(2 2)
Calculating Page 3
      Calculating E[3]^(0 0)
      Calculating E[3]^(0 1)
      Calculating E[3]^(0 2)
      Calculating E[3]^(1 0)
      Calculating E[3]^(1 1)
      Calculating E[3]^(1 2)
      Calculating E[3]^(2 0)
      Calculating E[3]^(2 1)
      Calculating E[3]^(2 2)

```

P := Book (3.16)

Reading the answer from the last page of the sequence.

```

V > Answer := ReadCohomology(P[3]);
Answer := table([0 = [ ], 1 = [ ], 2 = [ ], 3 = [ ], 4 = [ ], PageInfo
  = table([NumberOfSquares = 9, PageNumber = 3, SizeOfSubAlgebra = 2,
  SizeOfComplementarySpace = 2])])

```

(3.17)

Now we will test this against the Cohomology program.

```

alg3 > RC := RelativeChains([]);

```

$$\begin{aligned}
RC := & [[x1 \otimes 1, x2 \otimes 1, x3 \otimes 1], [x1 \otimes \mathfrak{I}, x1 \otimes \mathfrak{J}, x1 \otimes u, x1 \otimes v, x2 \otimes \mathfrak{I}, x2 \otimes \mathfrak{J}, x2 \\
& \otimes u, x2 \otimes v, x3 \otimes \mathfrak{I}, x3 \otimes \mathfrak{J}, x3 \otimes u, x3 \otimes v], [x1 \otimes \mathfrak{I} \wedge \mathfrak{J}, x1 \otimes \mathfrak{I} \wedge u, x1 \otimes \mathfrak{I} \\
& \wedge v, x1 \otimes \mathfrak{J} \wedge u, x1 \otimes \mathfrak{J} \wedge v, x1 \otimes u \wedge v, x2 \otimes \mathfrak{I} \wedge \mathfrak{J}, x2 \otimes \mathfrak{I} \wedge u, x2 \otimes \mathfrak{I} \wedge v, \\
& x2 \otimes \mathfrak{J} \wedge u, x2 \otimes \mathfrak{J} \wedge v, x2 \otimes u \wedge v, x3 \otimes \mathfrak{I} \wedge \mathfrak{J}, x3 \otimes \mathfrak{I} \wedge u, x3 \otimes \mathfrak{I} \wedge v, x3 \\
& \otimes \mathfrak{J} \wedge u, x3 \otimes \mathfrak{J} \wedge v, x3 \otimes u \wedge v], [x1 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u, x1 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge v, x1 \otimes \mathfrak{I} \\
& \wedge u \wedge v, x1 \otimes \mathfrak{J} \wedge u \wedge v, x2 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u, x2 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge v, x2 \otimes \mathfrak{I} \wedge u \wedge v, x2 \\
& \otimes \mathfrak{J} \wedge u \wedge v, x3 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u, x3 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge v, x3 \otimes \mathfrak{I} \wedge u \wedge v, x3 \otimes \mathfrak{J} \wedge u \\
& \wedge v], [x1 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u \wedge v, x2 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u \wedge v, x3 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u \wedge v], []]
\end{aligned} \tag{3.18}$$

They produced the same answer.

alg3 > Cohomology (RC) ;

$$[[], [], [], []] \tag{3.19}$$

Now we will look at the nuts and bolts of a Serre Hochschild spectral sequence book.

alg3 > op (P) ;

$$\begin{aligned}
table([-1 = table([(1, 0) = table([PageCohomology = table([Chains = [x1 \otimes u, x1 \otimes v, \tag{3.20} \\
& x2 \otimes u, x2 \otimes v, x3 \otimes u, x3 \otimes v], Forms = [x1 \otimes u, x1 \otimes v, x2 \otimes u, x2 \otimes v, x3 \otimes u, \\
& x3 \otimes v], ChainDerivative = [x1 \otimes \mathfrak{I} \wedge u - x3 \otimes \mathfrak{J} \wedge u, x1 \otimes \mathfrak{I} \wedge v - x3 \otimes \mathfrak{J} \wedge v, \\
& -x1 \otimes \mathfrak{J} \wedge u - x3 \otimes u \wedge v, -x1 \otimes \mathfrak{J} \wedge v - x1 \otimes u \wedge v, x3 \otimes \mathfrak{I} \wedge u, x3 \otimes \mathfrak{I} \\
& \wedge v]]), Exact = table([Chains = [], Forms = [], ChainDerivative = []]), Complement \\
& = table([Chains = [], Forms = [], ChainDerivative = []])], (2, 1) \\
& = table([PageCohomology = table([Chains = [x1 \otimes \mathfrak{I} \wedge u \wedge v, x1 \otimes \mathfrak{J} \wedge u \wedge v, x2 \\
& \otimes \mathfrak{I} \wedge u \wedge v, x2 \otimes \mathfrak{J} \wedge u \wedge v, x3 \otimes \mathfrak{I} \wedge u \wedge v, x3 \otimes \mathfrak{J} \wedge u \wedge v], Forms = [x1 \\
& \otimes \mathfrak{I} \wedge u \wedge v, x1 \otimes \mathfrak{J} \wedge u \wedge v, x2 \otimes \mathfrak{I} \wedge u \wedge v, x2 \otimes \mathfrak{J} \wedge u \wedge v, x3 \otimes \mathfrak{I} \wedge u \\
& \wedge v, x3 \otimes \mathfrak{J} \wedge u \wedge v], ChainDerivative = [x3 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u \wedge v, x1 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u \\
& \wedge v, x1 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u \wedge v, 0 x1 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u \wedge v, 0 x1 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u \wedge v, x3 \otimes \mathfrak{I} \\
& \wedge \mathfrak{J} \wedge u \wedge v]]), Exact = table([Chains = [], Forms = [], ChainDerivative = []]), \\
& Complement = table([Chains = [], Forms = [], ChainDerivative = []])], (0, 2) \\
& = table([PageCohomology = table([Chains = [x1 \otimes \mathfrak{I} \wedge \mathfrak{J}, x2 \otimes \mathfrak{I} \wedge \mathfrak{J}, x3 \otimes \mathfrak{I} \\
& \wedge \mathfrak{J}], Forms = [x1 \otimes \mathfrak{I} \wedge \mathfrak{J}, x2 \otimes \mathfrak{I} \wedge \mathfrak{J}, x3 \otimes \mathfrak{I} \wedge \mathfrak{J}], ChainDerivative = [-2 x1 \\
& \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u - x1 \otimes \mathfrak{J} \wedge u \wedge v, -x1 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge v - 2 x2 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u - x2 \\
& \otimes \mathfrak{J} \wedge u \wedge v + x3 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge v, -2 x3 \otimes \mathfrak{I} \wedge \mathfrak{J} \wedge u - x3 \otimes \mathfrak{J} \wedge u \wedge v]]),
\end{aligned}$$

$Exact = table([Chains = [], Forms = [], ChainDerivative = []]), Complement$
 $= table([Chains = [], Forms = [], ChainDerivative = []]), (0, 1)$
 $= table([PageCohomology = table([Chains = [x1 \otimes \mathfrak{g}, x1 \otimes \mathfrak{g}^2, x2 \otimes \mathfrak{g}, x2 \otimes \mathfrak{g}^2, x3 \otimes \mathfrak{g}, x3 \otimes \mathfrak{g}^2], Forms = [x1 \otimes \mathfrak{g}, x1 \otimes \mathfrak{g}^2, x2 \otimes \mathfrak{g}, x2 \otimes \mathfrak{g}^2, x3 \otimes \mathfrak{g}, x3 \otimes \mathfrak{g}^2],$
 $ChainDerivative = [x1 \otimes \mathfrak{g} \wedge u - x1 \otimes u \wedge v + x3 \otimes \mathfrak{g} \wedge \mathfrak{g}, x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 + x1 \otimes \mathfrak{g} \wedge u \wedge v + x1 \otimes \mathfrak{g}^2 \wedge u, x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 + x1 \otimes \mathfrak{g} \wedge u \wedge v + x2 \otimes \mathfrak{g} \wedge u - x2 \otimes u \wedge v - x3 \otimes \mathfrak{g} \wedge u, x1 \otimes \mathfrak{g}^2 \wedge u + x2 \otimes \mathfrak{g} \wedge v + x2 \otimes \mathfrak{g}^2 \wedge u - x3 \otimes \mathfrak{g}^2 \wedge v, x3 \otimes \mathfrak{g} \wedge u - x3 \otimes u \wedge v, x3 \otimes \mathfrak{g} \wedge \mathfrak{g} + x3 \otimes \mathfrak{g} \wedge v + x3 \otimes \mathfrak{g}^2 \wedge u]])], Exact = table([Chains = [], Forms = [], ChainDerivative = []]), Complement = table([Chains = [], Forms = [], ChainDerivative = []]), (1, 1) = table([PageCohomology = table([Chains = [x1 \otimes \mathfrak{g} \wedge u, x1 \otimes \mathfrak{g} \wedge v, x1 \otimes \mathfrak{g}^2 \wedge u, x1 \otimes \mathfrak{g}^2 \wedge v, x2 \otimes \mathfrak{g} \wedge u, x2 \otimes \mathfrak{g} \wedge v, x2 \otimes \mathfrak{g}^2 \wedge u, x2 \otimes \mathfrak{g}^2 \wedge v, x3 \otimes \mathfrak{g} \wedge u, x3 \otimes \mathfrak{g} \wedge v, x3 \otimes \mathfrak{g}^2 \wedge u, x3 \otimes \mathfrak{g}^2 \wedge v], Forms = [x1 \otimes \mathfrak{g} \wedge u, x1 \otimes \mathfrak{g} \wedge v, x1 \otimes \mathfrak{g}^2 \wedge u, x1 \otimes \mathfrak{g}^2 \wedge v, x2 \otimes \mathfrak{g} \wedge u, x2 \otimes \mathfrak{g} \wedge v, x2 \otimes \mathfrak{g}^2 \wedge u, x2 \otimes \mathfrak{g}^2 \wedge v, x3 \otimes \mathfrak{g} \wedge u, x3 \otimes \mathfrak{g} \wedge v, x3 \otimes \mathfrak{g}^2 \wedge u, x3 \otimes \mathfrak{g}^2 \wedge v],$
 $ChainDerivative = [x3 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u, x1 \otimes \mathfrak{g} \wedge u \wedge v + x3 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge v, x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u - x1 \otimes \mathfrak{g} \wedge u \wedge v, x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge v + x1 \otimes \mathfrak{g}^2 \wedge u \wedge v, x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u + x3 \otimes \mathfrak{g} \wedge u \wedge v, x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge v + x1 \otimes \mathfrak{g} \wedge u \wedge v + x2 \otimes \mathfrak{g} \wedge u \wedge v, -x2 \otimes \mathfrak{g} \wedge u \wedge v + x3 \otimes \mathfrak{g}^2 \wedge u \wedge v, x1 \otimes \mathfrak{g}^2 \wedge u \wedge v + x2 \otimes \mathfrak{g}^2 \wedge u \wedge v, 0 x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u, x3 \otimes \mathfrak{g} \wedge u \wedge v, x3 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u - x3 \otimes \mathfrak{g} \wedge u \wedge v, x3 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge v + x3 \otimes \mathfrak{g}^2 \wedge u \wedge v]])], Exact = table([Chains = [], Forms = [], ChainDerivative = []]), Complement = table([Chains = [], Forms = [], ChainDerivative = []]), (1, 2) = table([PageCohomology = table([Chains = [x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u, x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge v, x2 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u, x2 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge v, x3 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u, x3 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge v], Forms = [x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u, x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge v, x2 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u, x2 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge v, x3 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u, x3 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge v], ChainDerivative = [0 x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u \wedge v, -2 x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u \wedge v, -x3 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u \wedge v, -x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u \wedge v - 2 x2 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u \wedge v, 0 x1 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u \wedge v, -2 x3 \otimes \mathfrak{g} \wedge \mathfrak{g}^2 \wedge u \wedge v]])], Exact = table([Chains = [], Forms = [], ChainDerivative = []]), Complement = table([Chains = [], Forms = [], ChainDerivative = []]), PageInfo = table([Min = -1, NumberOfSquares = 9, PageNumber = -1, Max = 5, X = 1, Vectors = 3, SizeOfSubAlgebra = 2, SizeOfComplementarySpace = 2]), (0, 0)$

$= [x1 \otimes 1, x2 \otimes 1, x3 \otimes 1], \text{Forms} = [x1 \otimes \mathcal{I} - x3 \otimes \mathcal{J}, -x1 \otimes \mathcal{J}, x3 \otimes \mathcal{I}],$
 $\text{ChainDerivative} = [x1 \otimes \mathcal{I} - x3 \otimes \mathcal{J}, -x1 \otimes \mathcal{J} - x1 \otimes u + x3 \otimes v, x3 \otimes \mathcal{I}],$
 $\text{Complement} = \text{table}([\text{Chains} = [x2 \otimes \mathcal{J}, -x1 \otimes \mathcal{J} + x2 \otimes \mathcal{I}], \text{Forms} = [x2 \otimes \mathcal{J},$
 $-x1 \otimes \mathcal{J} + x2 \otimes \mathcal{I}], \text{ChainDerivative} = [x1 \otimes \mathcal{J} \wedge u + x2 \otimes \mathcal{I} \wedge v + x2 \otimes \mathcal{J} \wedge u$
 $- x3 \otimes \mathcal{J} \wedge v, x1 \otimes \mathcal{I} \wedge u - x1 \otimes \mathcal{I} \wedge v - x1 \otimes \mathcal{J} \wedge u + x2 \otimes \mathcal{I} \wedge u - x2 \otimes u$
 $\wedge v - x3 \otimes \mathcal{I} \wedge v]]), (1, 1) = \text{table}([\text{PageCohomology} = \text{table}([\text{Chains} = [$
 $\text{Forms} = [$
 $\text{ChainDerivative} = []]), \text{Exact} = \text{table}([\text{Chains} = [x1 \otimes u, x1 \otimes v, x2$
 $\otimes u, x2 \otimes v, x3 \otimes u, x3 \otimes v, x2 \otimes \mathcal{J}, -x1 \otimes \mathcal{J} + x2 \otimes \mathcal{I}], \text{Forms} = [x1 \otimes \mathcal{I} \wedge u$
 $- x3 \otimes \mathcal{J} \wedge u, x1 \otimes \mathcal{I} \wedge v - x3 \otimes \mathcal{J} \wedge v, -x1 \otimes \mathcal{J} \wedge u, -x1 \otimes \mathcal{J} \wedge v, x3 \otimes \mathcal{I}$
 $\wedge u, x3 \otimes \mathcal{I} \wedge v, -x1 \otimes \mathcal{J} \wedge v + x2 \otimes \mathcal{I} \wedge v + x2 \otimes \mathcal{J} \wedge u, -x1 \otimes \mathcal{J} \wedge u + x2$
 $\otimes \mathcal{I} \wedge u], \text{ChainDerivative} = [x1 \otimes \mathcal{I} \wedge u - x3 \otimes \mathcal{J} \wedge u, x1 \otimes \mathcal{I} \wedge v - x3 \otimes \mathcal{J}$
 $\wedge v, -x1 \otimes \mathcal{J} \wedge u - x3 \otimes u \wedge v, -x1 \otimes \mathcal{J} \wedge v - x1 \otimes u \wedge v, x3 \otimes \mathcal{I} \wedge u, x3$
 $\otimes \mathcal{I} \wedge v, x1 \otimes \mathcal{J} \wedge u + x2 \otimes \mathcal{I} \wedge v + x2 \otimes \mathcal{J} \wedge u - x3 \otimes \mathcal{J} \wedge v, x1 \otimes \mathcal{I} \wedge u$
 $- x1 \otimes \mathcal{I} \wedge v - x1 \otimes \mathcal{J} \wedge u + x2 \otimes \mathcal{I} \wedge u - x2 \otimes u \wedge v - x3 \otimes \mathcal{I} \wedge v]]),$
 $\text{Complement} = \text{table}([\text{Chains} = [x2 \otimes \mathcal{J} \wedge v, x2 \otimes \mathcal{J} \wedge u], \text{Forms} = [x2 \otimes \mathcal{J} \wedge v, x2$
 $\otimes \mathcal{J} \wedge u], \text{ChainDerivative} = [x1 \otimes \mathcal{J} \wedge u \wedge v + x2 \otimes \mathcal{J} \wedge u \wedge v, -x2 \otimes \mathcal{I} \wedge u$
 $\wedge v + x3 \otimes \mathcal{J} \wedge u \wedge v]]), (1, 2) = \text{table}([\text{PageCohomology} = \text{table}([\text{Chains}$
 $= [$
 $\text{Forms} = [$
 $\text{ChainDerivative} = []]), \text{Exact} = \text{table}([\text{Chains} = [x1 \otimes \mathcal{I} \wedge u, x1$
 $\otimes \mathcal{I} \wedge v, x1 \otimes \mathcal{J} \wedge u, x1 \otimes \mathcal{J} \wedge v, x2 \otimes \mathcal{I} \wedge \mathcal{J}], \text{Forms} = [x3 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u, x3$
 $\otimes \mathcal{I} \wedge \mathcal{J} \wedge v, x1 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u, x1 \otimes \mathcal{I} \wedge \mathcal{J} \wedge v, -2x2 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u],$
 $\text{ChainDerivative} = [x3 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u, x1 \otimes \mathcal{I} \wedge u \wedge v + x3 \otimes \mathcal{I} \wedge \mathcal{J} \wedge v, x1 \otimes \mathcal{I}$
 $\wedge \mathcal{J} \wedge u - x1 \otimes \mathcal{I} \wedge u \wedge v, x1 \otimes \mathcal{I} \wedge \mathcal{J} \wedge v + x1 \otimes \mathcal{J} \wedge u \wedge v, x3 \otimes \mathcal{I} \wedge u, x3$
 $\otimes \mathcal{I} \wedge v, -x1 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u - 2x2 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u - x2 \otimes \mathcal{J} \wedge u \wedge v + x3 \otimes \mathcal{I}$
 $\wedge \mathcal{J} \wedge v, x1 \otimes \mathcal{I} \wedge u - x1 \otimes \mathcal{I} \wedge v - x1 \otimes \mathcal{J} \wedge u + x2 \otimes \mathcal{I} \wedge u - x2 \otimes u \wedge v$
 $- x3 \otimes \mathcal{I} \wedge v]]), \text{Complement} = \text{table}([\text{Chains} = [x2 \otimes \mathcal{I} \wedge \mathcal{J}], \text{Forms} = [x2$
 $\otimes \mathcal{I} \wedge \mathcal{J}], \text{ChainDerivative} = [-x1 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u \wedge v - 2x2 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u$
 $\wedge v]]), \text{PageInfo} = \text{table}([\text{Min} = -1, \text{NumberOfSquares} = 9, \text{PageNumber} = 1, \text{Max}$
 $= 5, X = 1, \text{Vectors} = 3, \text{SizeOfSubAlgebra} = 2, \text{SizeOfComplementarySpace} = 2]), (0, 0)$
 $= \text{table}([\text{PageCohomology} = \text{table}([\text{Chains} = [$
 $\text{Forms} = [$
 $\text{ChainDerivative} = []]),$
 $\text{Exact} = \text{table}([\text{Chains} = [$
 $\text{Forms} = [$
 $\text{ChainDerivative} = []]), \text{Complement}$
 $= \text{table}([\text{Chains} = [$
 $\text{Forms} = [$
 $\text{ChainDerivative} = []])], (2, 2)$
 $= \text{table}([\text{PageCohomology} = \text{table}([\text{Chains} = [$
 $\text{Forms} = [$
 $\text{ChainDerivative} = []]),$
 $\text{Exact} = \text{table}([\text{Chains} = [x1 \otimes \mathcal{I} \wedge u \wedge v, x1 \otimes \mathcal{J} \wedge u \wedge v, x2 \otimes \mathcal{I} \wedge \mathcal{J}],$
 $\text{Forms} = [x3 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u \wedge v, x1 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u \wedge v, -2x2 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u \wedge v],$
 $\text{ChainDerivative} = [x3 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u \wedge v, x1 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u \wedge v, x3 \otimes \mathcal{I} \wedge u \wedge v, x1$
 $\otimes \mathcal{I} \wedge \mathcal{J} \wedge v + x1 \otimes \mathcal{J} \wedge u \wedge v, x3 \otimes \mathcal{I} \wedge u, x3 \otimes \mathcal{I} \wedge v, -x1 \otimes \mathcal{I} \wedge \mathcal{J} \wedge u$

$\wedge u_2 - 2x_2 \otimes \gamma_1 \wedge \gamma_2 \wedge u_1 \wedge u_2, -x_2 \otimes \gamma_1 \wedge u_1 \wedge u_2 + x_3 \otimes \gamma_2 \wedge u_1 \wedge u_2]]$),
Complement = *table*([*Chains* = [], *Forms* = [], *ChainDerivative* = []]), (2, 0)
= *table*([*PageCohomology* = *table*([*Chains* = [], *Forms* = [], *ChainDerivative* = [])],
Exact = *table*([*Chains* = [], *Forms* = [], *ChainDerivative* = [])], *Complement*
= *table*([*Chains* = [], *Forms* = [], *ChainDerivative* = [])]), 2 = *table*((1, 0)
= *table*([*PageCohomology* = *table*([*Chains* = [], *Forms* = [], *ChainDerivative* = [])],
Exact = *table*([*Chains* = [], *Forms* = [], *ChainDerivative* = [])], *Complement*
= *table*([*Chains* = [], *Forms* = [], *ChainDerivative* = [])]), (2, 1)
= *table*([*PageCohomology* = *table*([*Chains* = [], *Forms* = [], *ChainDerivative* = [])],
Exact = *table*([*Chains* = [$x_1 \otimes u_1 \wedge u_2, x_2 \otimes u_1 \wedge u_2, x_3 \otimes u_1 \wedge u_2, x_2 \otimes \gamma_2 \wedge u_2, x_2 \otimes \gamma_2$
 $\wedge u_1$], *Forms* = [$x_1 \otimes \gamma_1 \wedge u_1 \wedge u_2 - x_3 \otimes \gamma_2 \wedge u_1 \wedge u_2, -x_1 \otimes \gamma_2 \wedge u_1 \wedge u_2, x_3 \otimes \gamma_1$
 $\wedge u_1 \wedge u_2, x_2 \otimes \gamma_2 \wedge u_1 \wedge u_2, x_1 \otimes \gamma_2 \wedge u_1 \wedge u_2 - x_2 \otimes \gamma_1 \wedge u_1 \wedge u_2$], *ChainDerivative*
= [$x_1 \otimes \gamma_1 \wedge u_1 \wedge u_2 - x_3 \otimes \gamma_2 \wedge u_1 \wedge u_2, -x_1 \otimes \gamma_2 \wedge u_1 \wedge u_2, x_3 \otimes \gamma_1 \wedge u_1 \wedge u_2, x_1$
 $\otimes \gamma_1 \wedge \gamma_2 \wedge u_2 + x_1 \otimes \gamma_2 \wedge u_1 \wedge u_2, x_3 \otimes \gamma_1 \wedge u_1, x_3 \otimes \gamma_1 \wedge u_2, x_1 \otimes \gamma_2 \wedge u_1 \wedge u_2 + x_2$
 $\otimes \gamma_2 \wedge u_1 \wedge u_2, -x_2 \otimes \gamma_1 \wedge u_1 \wedge u_2 + x_3 \otimes \gamma_2 \wedge u_1 \wedge u_2$]]), *Complement*
= *table*([*Chains* = [], *Forms* = [], *ChainDerivative* = [])]), (0, 2)
= *table*([*PageCohomology* = *table*([*Chains* = [], *Forms* = [], *ChainDerivative* = [])],
Exact = *table*([*Chains* = [$x_1 \otimes \gamma_1, x_1 \otimes \gamma_2$], *Forms* = [$x_3 \otimes \gamma_1 \wedge \gamma_2, x_1 \otimes \gamma_1 \wedge \gamma_2$],
ChainDerivative = [$x_1 \otimes \gamma_1 \wedge u_1 - x_1 \otimes u_1 \wedge u_2 + x_3 \otimes \gamma_1 \wedge \gamma_2, x_1 \otimes \gamma_1 \wedge \gamma_2 + x_1 \otimes \gamma_1$
 $\wedge u_2 + x_1 \otimes \gamma_2 \wedge u_1, x_3 \otimes \gamma_1$]]), *Complement* = *table*([*Chains* = [], *Forms* = [],
ChainDerivative = [])]), (0, 1) = *table*([*PageCohomology* = *table*([*Chains* = [],
Forms = [], *ChainDerivative* = [])], *Exact* = *table*([*Chains* = [$x_1 \otimes 1, x_2 \otimes 1, x_3 \otimes 1$],
Forms = [$x_1 \otimes \gamma_1 - x_3 \otimes \gamma_2, -x_1 \otimes \gamma_2, x_3 \otimes \gamma_1$], *ChainDerivative* = [$x_1 \otimes \gamma_1 - x_3$
 $\otimes \gamma_2, -x_1 \otimes \gamma_2 - x_1 \otimes u_1 + x_3 \otimes u_2, x_3 \otimes \gamma_1$]]), *Complement* = *table*([*Chains* = [],
Forms = [], *ChainDerivative* = [])]), (1, 1) = *table*([*PageCohomology*
= *table*([*Chains* = [], *Forms* = [], *ChainDerivative* = [])], *Exact* = *table*([*Chains*
= [$x_1 \otimes u_1, x_1 \otimes u_2, x_2 \otimes u_1, x_2 \otimes u_2, x_3 \otimes u_1, x_3 \otimes u_2, x_2 \otimes \gamma_2, -x_1 \otimes \gamma_2 + x_2 \otimes \gamma_1$],
Forms = [$x_1 \otimes \gamma_1 \wedge u_1 - x_3 \otimes \gamma_2 \wedge u_1, x_1 \otimes \gamma_1 \wedge u_2 - x_3 \otimes \gamma_2 \wedge u_2, -x_1 \otimes \gamma_2 \wedge u_1,$
 $-x_1 \otimes \gamma_2 \wedge u_2, x_3 \otimes \gamma_1 \wedge u_1, x_3 \otimes \gamma_1 \wedge u_2, -x_1 \otimes \gamma_2 \wedge u_2 + x_2 \otimes \gamma_1 \wedge u_2 + x_2 \otimes \gamma_2$
 $\wedge u_1, -x_1 \otimes \gamma_2 \wedge u_1 + x_2 \otimes \gamma_1 \wedge u_1$], *ChainDerivative* = [$x_1 \otimes \gamma_1 \wedge u_1 - x_3 \otimes \gamma_2 \wedge u_1,$
 $x_1 \otimes \gamma_1 \wedge u_2 - x_3 \otimes \gamma_2 \wedge u_2, -x_1 \otimes \gamma_2 \wedge u_1 - x_3 \otimes u_1 \wedge u_2, -x_1 \otimes \gamma_2 \wedge u_2 - x_1 \otimes u_1$
 $\wedge u_2, x_3 \otimes \gamma_1 \wedge u_1, x_3 \otimes \gamma_1 \wedge u_2, x_1 \otimes \gamma_2 \wedge u_1 + x_2 \otimes \gamma_1 \wedge u_2 + x_2 \otimes \gamma_2 \wedge u_1 - x_3 \otimes \gamma_2$
 $\wedge u_2, x_1 \otimes \gamma_1 \wedge u_1 - x_1 \otimes \gamma_1 \wedge u_2 - x_1 \otimes \gamma_2 \wedge u_1 + x_2 \otimes \gamma_1 \wedge u_1 - x_2 \otimes u_1 \wedge u_2 - x_3$
 $\otimes \gamma_1 \wedge u_2$]]), *Complement* = *table*([*Chains* = [], *Forms* = [], *ChainDerivative* = [])])
), (1, 2) = *table*([*PageCohomology* = *table*([*Chains* = [], *Forms* = [],
ChainDerivative = [])], *Exact* = *table*([*Chains* = [$x_1 \otimes \gamma_1 \wedge u_1, x_1 \otimes \gamma_1 \wedge u_2, x_1 \otimes \gamma_2$

$\wedge u, x1 \otimes \mathcal{P} \wedge u, x2 \otimes \mathcal{P} \wedge \mathcal{P}], Forms = [x3 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u, x3 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u, x1$
 $\otimes \mathcal{P} \wedge \mathcal{P} \wedge u, x1 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u, -2 x2 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u], ChainDerivative = [x3 \otimes \mathcal{P}$
 $\wedge \mathcal{P} \wedge u, x1 \otimes \mathcal{P} \wedge u \wedge u + x3 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u, x1 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u - x1 \otimes \mathcal{P} \wedge u$
 $\wedge u, x1 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u + x1 \otimes \mathcal{P} \wedge u \wedge u, x3 \otimes \mathcal{P} \wedge u, x3 \otimes \mathcal{P} \wedge u, -x1 \otimes \mathcal{P} \wedge \mathcal{P}$
 $\wedge u - 2 x2 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u - x2 \otimes \mathcal{P} \wedge u \wedge u + x3 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u, x1 \otimes \mathcal{P} \wedge u - x1$
 $\otimes \mathcal{P} \wedge u - x1 \otimes \mathcal{P} \wedge u + x2 \otimes \mathcal{P} \wedge u - x2 \otimes u \wedge u - x3 \otimes \mathcal{P} \wedge u]]),$
Complement = table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]), *PageInfo*
= table([*Min* = -1, *NumberOfSquares* = 9, *PageNumber* = 2, *Max* = 5, *X* = 1, *Vectors*
= 3, *SizeOfSubAlgebra* = 2, *SizeOfComplementarySpace* = 2]), (0, 0)
= table([*PageCohomology* = table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]),
Exact = table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]), *Complement*
= table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]), (2, 2)
= table([*PageCohomology* = table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]),
Exact = table([*Chains* = [$x1 \otimes \mathcal{P} \wedge u \wedge u, x1 \otimes \mathcal{P} \wedge u \wedge u, x2 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u$],
Forms = [$x3 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u \wedge u, x1 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u \wedge u, -2 x2 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u \wedge u$],
ChainDerivative = [$x3 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u \wedge u, x1 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u \wedge u, x3 \otimes \mathcal{P} \wedge u \wedge u, x1$
 $\otimes \mathcal{P} \wedge \mathcal{P} \wedge u + x1 \otimes \mathcal{P} \wedge u \wedge u, x3 \otimes \mathcal{P} \wedge u, x3 \otimes \mathcal{P} \wedge u, -x1 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u$
 $\wedge u - 2 x2 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u \wedge u, -x2 \otimes \mathcal{P} \wedge u \wedge u + x3 \otimes \mathcal{P} \wedge u \wedge u$]]]),
Complement = table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]), (2, 0)
= table([*PageCohomology* = table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]),
Exact = table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]), *Complement*
= table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]), 3 = table([(1, 0)
= table([*PageCohomology* = table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]),
Exact = table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]), *Complement*
= table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]), (2, 1)
= table([*PageCohomology* = table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]),
Exact = table([*Chains* = [$x1 \otimes u \wedge u, x2 \otimes u \wedge u, x3 \otimes u \wedge u, x2 \otimes \mathcal{P} \wedge u, x2 \otimes \mathcal{P}$
 $\wedge u$], *Forms* = [$x1 \otimes \mathcal{P} \wedge u \wedge u - x3 \otimes \mathcal{P} \wedge u \wedge u, -x1 \otimes \mathcal{P} \wedge u \wedge u, x3 \otimes \mathcal{P}$
 $\wedge u \wedge u, x2 \otimes \mathcal{P} \wedge u \wedge u, x1 \otimes \mathcal{P} \wedge u \wedge u - x2 \otimes \mathcal{P} \wedge u \wedge u$], *ChainDerivative*
= [$x1 \otimes \mathcal{P} \wedge u \wedge u - x3 \otimes \mathcal{P} \wedge u \wedge u, -x1 \otimes \mathcal{P} \wedge u \wedge u, x3 \otimes \mathcal{P} \wedge u \wedge u, x1$
 $\otimes \mathcal{P} \wedge \mathcal{P} \wedge u + x1 \otimes \mathcal{P} \wedge u \wedge u, x3 \otimes \mathcal{P} \wedge u, x3 \otimes \mathcal{P} \wedge u, x1 \otimes \mathcal{P} \wedge u \wedge u + x2$
 $\otimes \mathcal{P} \wedge u \wedge u, -x2 \otimes \mathcal{P} \wedge u \wedge u + x3 \otimes \mathcal{P} \wedge u \wedge u$]]]), *Complement*
= table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]), (0, 2)
= table([*PageCohomology* = table([*Chains* = [], *Forms* = [], *ChainDerivative* = []]),
Exact = table([*Chains* = [$x1 \otimes \mathcal{P}, x1 \otimes \mathcal{P}$], *Forms* = [$x3 \otimes \mathcal{P} \wedge \mathcal{P}, x1 \otimes \mathcal{P} \wedge \mathcal{P}$],
ChainDerivative = [$x1 \otimes \mathcal{P} \wedge u - x1 \otimes u \wedge u + x3 \otimes \mathcal{P} \wedge \mathcal{P}, x1 \otimes \mathcal{P} \wedge \mathcal{P} + x1 \otimes \mathcal{P}$

$\wedge u_2 + x_1 \otimes \mathcal{P} \wedge u_1, x_3 \otimes \mathcal{P}]])$, $Complement = table([Chains = [], Forms = [],$
 $ChainDerivative = []]])$, $(0, 1) = table([PageCohomology = table([Chains = [],$
 $Forms = [], ChainDerivative = []])$, $Exact = table([Chains = [x_1 \otimes 1, x_2 \otimes 1, x_3 \otimes 1],$
 $Forms = [x_1 \otimes \mathcal{P} - x_3 \otimes \mathcal{P}, -x_1 \otimes \mathcal{P}, x_3 \otimes \mathcal{P}], ChainDerivative = [x_1 \otimes \mathcal{P} - x_3$
 $\otimes \mathcal{P}, -x_1 \otimes \mathcal{P} - x_1 \otimes u_1 + x_3 \otimes u_2, x_3 \otimes \mathcal{P}]])$, $Complement = table([Chains = [],$
 $Forms = [], ChainDerivative = []]])$, $(1, 1) = table([PageCohomology$
 $= table([Chains = [], Forms = [], ChainDerivative = []])$, $Exact = table([Chains$
 $= [x_1 \otimes u_1, x_1 \otimes u_2, x_2 \otimes u_1, x_2 \otimes u_2, x_3 \otimes u_1, x_3 \otimes u_2, x_2 \otimes \mathcal{P}, -x_1 \otimes \mathcal{P} + x_2 \otimes \mathcal{P}],$
 $Forms = [x_1 \otimes \mathcal{P} \wedge u_1 - x_3 \otimes \mathcal{P} \wedge u_1, x_1 \otimes \mathcal{P} \wedge u_2 - x_3 \otimes \mathcal{P} \wedge u_2, -x_1 \otimes \mathcal{P} \wedge u_1,$
 $-x_1 \otimes \mathcal{P} \wedge u_2, x_3 \otimes \mathcal{P} \wedge u_1, x_3 \otimes \mathcal{P} \wedge u_2, -x_1 \otimes \mathcal{P} \wedge u_2 + x_2 \otimes \mathcal{P} \wedge u_2 + x_2 \otimes \mathcal{P}$
 $\wedge u_1, -x_1 \otimes \mathcal{P} \wedge u_1 + x_2 \otimes \mathcal{P} \wedge u_1], ChainDerivative = [x_1 \otimes \mathcal{P} \wedge u_1 - x_3 \otimes \mathcal{P} \wedge u_1,$
 $x_1 \otimes \mathcal{P} \wedge u_2 - x_3 \otimes \mathcal{P} \wedge u_2, -x_1 \otimes \mathcal{P} \wedge u_1 - x_3 \otimes u_1 \wedge u_2, -x_1 \otimes \mathcal{P} \wedge u_2 - x_1 \otimes u_1$
 $\wedge u_2, x_3 \otimes \mathcal{P} \wedge u_1, x_3 \otimes \mathcal{P} \wedge u_2, x_1 \otimes \mathcal{P} \wedge u_1 + x_2 \otimes \mathcal{P} \wedge u_2 + x_2 \otimes \mathcal{P} \wedge u_1 - x_3 \otimes \mathcal{P}$
 $\wedge u_2, x_1 \otimes \mathcal{P} \wedge u_1 - x_1 \otimes \mathcal{P} \wedge u_2 - x_1 \otimes \mathcal{P} \wedge u_1 + x_2 \otimes \mathcal{P} \wedge u_1 - x_2 \otimes u_1 \wedge u_2 - x_3$
 $\otimes \mathcal{P} \wedge u_2]])$, $Complement = table([Chains = [], Forms = [], ChainDerivative = []]])$
 $)$, $(1, 2) = table([PageCohomology = table([Chains = [], Forms = [],$
 $ChainDerivative = []]])$, $Exact = table([Chains = [x_1 \otimes \mathcal{P} \wedge u_1, x_1 \otimes \mathcal{P} \wedge u_2, x_1 \otimes \mathcal{P}$
 $\wedge u_1, x_1 \otimes \mathcal{P} \wedge u_2, x_2 \otimes \mathcal{P} \wedge \mathcal{P}], Forms = [x_3 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_1, x_3 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_2, x_1$
 $\otimes \mathcal{P} \wedge \mathcal{P} \wedge u_1, x_1 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_2, -2x_2 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_1], ChainDerivative = [x_3 \otimes \mathcal{P}$
 $\wedge \mathcal{P} \wedge u_1, x_1 \otimes \mathcal{P} \wedge u_1 \wedge u_2 + x_3 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_2, x_1 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_1 - x_1 \otimes \mathcal{P} \wedge u_1$
 $\wedge u_2, x_1 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_2 + x_1 \otimes \mathcal{P} \wedge u_1 \wedge u_2, x_3 \otimes \mathcal{P} \wedge u_1, x_3 \otimes \mathcal{P} \wedge u_2, -x_1 \otimes \mathcal{P} \wedge \mathcal{P}$
 $\wedge u_1 - 2x_2 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_1 - x_2 \otimes \mathcal{P} \wedge u_1 \wedge u_2 + x_3 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_2, x_1 \otimes \mathcal{P} \wedge u_1 - x_1$
 $\otimes \mathcal{P} \wedge u_2 - x_1 \otimes \mathcal{P} \wedge u_1 + x_2 \otimes \mathcal{P} \wedge u_1 - x_2 \otimes u_1 \wedge u_2 - x_3 \otimes \mathcal{P} \wedge u_2]])$,
 $Complement = table([Chains = [], Forms = [], ChainDerivative = []]])$, $PageInfo$
 $= table([Min = -1, NumberOfSquares = 9, PageNumber = 3, Max = 5, X = 1, Vectors$
 $= 3, SizeOfSubAlgebra = 2, SizeOfComplementarySpace = 2])$, $(0, 0)$
 $= table([PageCohomology = table([Chains = [], Forms = [], ChainDerivative = []]])$,
 $Exact = table([Chains = [], Forms = [], ChainDerivative = []]])$, $Complement$
 $= table([Chains = [], Forms = [], ChainDerivative = []]])$, $(2, 2)$
 $= table([PageCohomology = table([Chains = [], Forms = [], ChainDerivative = []]])$,
 $Exact = table([Chains = [x_1 \otimes \mathcal{P} \wedge u_1 \wedge u_2, x_1 \otimes \mathcal{P} \wedge u_1 \wedge u_2, x_2 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_2],$
 $Forms = [x_3 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_1 \wedge u_2, x_1 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_1 \wedge u_2, -2x_2 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_1 \wedge u_2],$
 $ChainDerivative = [x_3 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_1 \wedge u_2, x_1 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_1 \wedge u_2, x_3 \otimes \mathcal{P} \wedge u_1 \wedge u_2, x_1$
 $\otimes \mathcal{P} \wedge \mathcal{P} \wedge u_2 + x_1 \otimes \mathcal{P} \wedge u_1 \wedge u_2, x_3 \otimes \mathcal{P} \wedge u_1, x_3 \otimes \mathcal{P} \wedge u_2, -x_1 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_1$
 $\wedge u_2 - 2x_2 \otimes \mathcal{P} \wedge \mathcal{P} \wedge u_1 \wedge u_2, -x_2 \otimes \mathcal{P} \wedge u_1 \wedge u_2 + x_3 \otimes \mathcal{P} \wedge u_1 \wedge u_2]])$,
 $Complement = table([Chains = [], Forms = [], ChainDerivative = []]])$, $(2, 0)$

$$\begin{aligned} &= \text{table}([\text{PageCohomology} = \text{table}([\text{Chains} = [], \text{Forms} = [], \text{ChainDerivative} = []]), \\ &\text{Exact} = \text{table}([\text{Chains} = [], \text{Forms} = [], \text{ChainDerivative} = []]), \text{Complement} \\ &= \text{table}([\text{Chains} = [], \text{Forms} = [], \text{ChainDerivative} = []])]), \text{PageNumber} = 3) \end{aligned}$$

5.3 *Calculating Multiple Pages at once*

In this section we will be testing the program against a Lie algebra. I will be calculating multiple pages at once. I will also show that you can take the book you've calculated and add more pages when ever you want without having to start from the beginning.

Multiple Pages

Set up

```
alg2 > restart:  
> with(Library);  
      [Browse, DGplot, DGshow, MetricSearch, References, Retrieve, Search] (4.1.1)
```

This will allow us to see the page and square the program is working on. Plus it will give us how many forms are in each square as well.

```
alg1 > infolevel[SHSSLAWC] := 4;  
      infolevelSHSSLAWC := 4 (4.1.2)
```

Setting up the Lie algebra

```
> LD:=Retrieve("Winternitz",1,[6,19],alg1);  
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3] (4.1.3)  
      ]=e6, [e2, e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0,  
      [e4, e5]=0, [e4, e6]=0, [e5, e6]=0
```

```
> DGsetup(LD);  
      Lie algebra: alg1 (4.1.4)
```

```
> MultiplicationTable();
```

alg1	<i>e1</i>	<i>e2</i>	<i>e3</i>	<i>e4</i>	<i>e5</i>	<i>e6</i>
<i>e1</i>	0	<i>e3</i>	<i>e4</i>	<i>e5</i>	<i>e6</i>	0
<i>e2</i>	- <i>e3</i>	0	<i>e6</i>	0	0	0
<i>e3</i>	- <i>e4</i>	- <i>e6</i>	0	0	0	0
<i>e4</i>	- <i>e5</i>	0	0	0	0	0
<i>e5</i>	- <i>e6</i>	0	0	0	0	0
<i>e6</i>	0	0	0	0	0	0

(4.1.5)

Finding the Subalgebra

```
alg1 > Query([e4, e5, e6], "Subalgebra");  
      true (4.1.6)
```

```
alg1 > Query([e4, e5, e6], "Ideal");  
      true (4.1.7)
```

Renaming the algebra into our alpha beta convention

```
alg1 > B:=[e4, e5, e6, e1, e2, e3]  
      B := [e4, e5, e6, e1, e2, e3] (4.1.8)
```

```
alg1 > DGEEnvironment[LieAlgebra](B, vectorlabels=[x1, x2, x3,  
      y1, y2, y3], formlabels=[alpha1, alpha2, alpha3, beta1,  
      beta2, beta3], alg2);  
      Lie algebra: alg2 (4.1.9)
```

```
alg2 > MultiplicationTable();
```

<code>alg2</code>	$x1$	$x2$	$x3$	$y1$	$y2$	$y3$	
$x1$	0	0	0	$-x2$	0	0	
$x2$	0	0	0	$-x3$	0	0	
$x3$	0	0	0	0	0	0	
$y1$	$x2$	$x3$	0	0	$y3$	$x1$	
$y2$	0	0	0	$-y3$	0	$x3$	
$y3$	0	0	0	$-x1$	$-x3$	0	(4.1.10)

Checking to see if the complementary space really is the complementary space.

```
alg2 > Annihilator([x1,x2,x3]);
           [\beta1, \beta2, \beta3]
```

(4.1.11)

```
alg2 > DisplayStructureEquations();
      d \alpha1 = -\beta1 \wedge \beta3
      d \alpha2 = \alpha1 \wedge \beta1
      d \alpha3 = \alpha2 \wedge \beta1 - \beta2 \wedge \beta3
      d \beta1 = 0 \alpha1 \wedge \alpha2
      d \beta2 = 0 \alpha1 \wedge \alpha2
      d \beta3 = -\beta1 \wedge \beta2
```

(4.1.12)

```
alg2 > Alpha:=evalDG([alpha1,alpha2,alpha3]);
           A := [\alpha1, \alpha2, \alpha3]
```

(4.1.13)

Define the Complimentary Space

```
alg2 > Beta1:=ComplementaryBasis(Alpha);
           B1 := [\beta1, \beta2, \beta3]
```

(4.1.14)

▼ Calculating Multiple Pages at once

```
alg2 > Pages := SHSSLAWC2(Alpha,Beta1, NumberOfPages = 3);
Setting up Sequence
Calculating Page 0
           Calculating E[0]^(0 0)
on Previous Page: 0
           Number of Forms
on the Page: 0
           Number of Forms
Forms: 0
           Number of Exact
NonClosed Forms: 0
           Number of
           Calculating E[0]^(0 1)
on Previous Page: 3
           Number of Forms
           Number of Forms
```

on the Page: 3		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[0]^{(0\ 2)}$	Number of Forms
on Previous Page: 3		Number of Forms
on the Page: 3		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[0]^{(0\ 3)}$	Number of Forms
on Previous Page: 1		Number of Forms
on the Page: 1		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[0]^{(1\ 0)}$	Number of Forms
on Previous Page: 3		Number of Forms
on the Page: 3		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[0]^{(1\ 1)}$	Number of Forms
on Previous Page: 9		Number of Forms
on the Page: 9		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[0]^{(1\ 2)}$	Number of Forms
on Previous Page: 9		Number of Forms
on the Page: 9		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[0]^{(1\ 3)}$	Number of Forms
on Previous Page: 3		Number of Forms
on the Page: 3		Number of Exact
Forms: 0		

NonClosed Forms: 0	Calculating $E[0]^{(2\ 0)}$	Number of
on Previous Page: 3		Number of Forms
on the Page: 3		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[0]^{(2\ 1)}$	Number of
on Previous Page: 9		Number of Forms
on the Page: 9		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[0]^{(2\ 2)}$	Number of
on Previous Page: 9		Number of Forms
on the Page: 9		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[0]^{(2\ 3)}$	Number of
on Previous Page: 3		Number of Forms
on the Page: 3		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[0]^{(3\ 0)}$	Number of
on Previous Page: 1		Number of Forms
on the Page: 1		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[0]^{(3\ 1)}$	Number of
on Previous Page: 3		Number of Forms
on the Page: 3		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[0]^{(3\ 2)}$	Number of

on Previous Page: 3		Number of Forms
on the Page: 3		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
	Calculating $E[0]^{(3\ 3)}$	
on Previous Page: 1		Number of Forms
on the Page: 1		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
Calculating Page 1		
	Calculating $E[1]^{(0\ 0)}$	
on Previous Page: 0		Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
	Calculating $E[1]^{(0\ 1)}$	
on Previous Page: 3		Number of Forms
on the Page: 1		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 2		Number of
	Calculating $E[1]^{(0\ 2)}$	
on Previous Page: 3		Number of Forms
on the Page: 1		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 2		Number of
	Calculating $E[1]^{(0\ 3)}$	
on Previous Page: 1		Number of Forms
on the Page: 1		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
	Calculating $E[1]^{(1\ 0)}$	
on Previous Page: 3		Number of Forms

on the Page: 2		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 1		Number of
	Calculating $E[1]^{(1\ 1)}$	
on Previous Page: 9		Number of Forms
on the Page: 3		Number of Forms
Forms: 2		Number of Exact
NonClosed Forms: 4		Number of
	Calculating $E[1]^{(1\ 2)}$	
on Previous Page: 9		Number of Forms
on the Page: 3		Number of Forms
Forms: 2		Number of Exact
NonClosed Forms: 4		Number of
	Calculating $E[1]^{(1\ 3)}$	
on Previous Page: 3		Number of Forms
on the Page: 2		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 1		Number of
	Calculating $E[1]^{(2\ 0)}$	
on Previous Page: 3		Number of Forms
on the Page: 2		Number of Forms
Forms: 1		Number of Exact
NonClosed Forms: 0		Number of
	Calculating $E[1]^{(2\ 1)}$	
on Previous Page: 9		Number of Forms
on the Page: 3		Number of Forms
Forms: 4		Number of Exact
NonClosed Forms: 2		Number of
	Calculating $E[1]^{(2\ 2)}$	
on Previous Page: 9		Number of Forms
on the Page: 3		Number of Forms
		Number of Exact

Forms: 4		Number of
NonClosed Forms: 2	Calculating $E[1]^{(2\ 3)}$	Number of Forms
on Previous Page: 3		Number of Forms
on the Page: 2		Number of Exact
Forms: 1		Number of
NonClosed Forms: 0	Calculating $E[1]^{(3\ 0)}$	Number of Forms
on Previous Page: 1		Number of Forms
on the Page: 1		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[1]^{(3\ 1)}$	Number of Forms
on Previous Page: 3		Number of Forms
on the Page: 1		Number of Exact
Forms: 2		Number of
NonClosed Forms: 0	Calculating $E[1]^{(3\ 2)}$	Number of Forms
on Previous Page: 3		Number of Forms
on the Page: 1		Number of Exact
Forms: 2		Number of
NonClosed Forms: 0	Calculating $E[1]^{(3\ 3)}$	Number of Forms
on Previous Page: 1		Number of Forms
on the Page: 1		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[2]^{(0\ 0)}$	Number of Forms
Calculating Page 2		Number of Forms
on Previous Page: 0		Number of Exact
on the Page: 0		Number of
Forms: 0		Number of

NonClosed Forms: 0	Calculating $E[2]^{(0\ 1)}$	Number of Forms
on Previous Page: 1		Number of Forms
on the Page: 0		Number of Exact
Forms: 0		Number of
NonClosed Forms: 1	Calculating $E[2]^{(0\ 2)}$	Number of Forms
on Previous Page: 1		Number of Forms
on the Page: 0		Number of Exact
Forms: 0		Number of
NonClosed Forms: 1	Calculating $E[2]^{(0\ 3)}$	Number of Forms
on Previous Page: 1		Number of Forms
on the Page: 0		Number of Exact
Forms: 0		Number of
NonClosed Forms: 1	Calculating $E[2]^{(1\ 0)}$	Number of Forms
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[2]^{(1\ 1)}$	Number of Forms
on Previous Page: 3		Number of Forms
on the Page: 2		Number of Exact
Forms: 0		Number of
NonClosed Forms: 1	Calculating $E[2]^{(1\ 2)}$	Number of Forms
on Previous Page: 3		Number of Forms
on the Page: 2		Number of Exact
Forms: 0		Number of
NonClosed Forms: 1	Calculating $E[2]^{(1\ 3)}$	Number of Forms

on Previous Page: 2		Number of Forms
on the Page: 1		Number of Exact
Forms: 0		Number of
NonClosed Forms: 1	Calculating $E[2]^{(2\ 0)}$	Number of Forms
on Previous Page: 2		Number of Forms
on the Page: 1		Number of Exact
Forms: 1		Number of
NonClosed Forms: 0	Calculating $E[2]^{(2\ 1)}$	Number of Forms
on Previous Page: 3		Number of Forms
on the Page: 2		Number of Exact
Forms: 1		Number of
NonClosed Forms: 0	Calculating $E[2]^{(2\ 2)}$	Number of Forms
on Previous Page: 3		Number of Forms
on the Page: 2		Number of Exact
Forms: 1		Number of
NonClosed Forms: 0	Calculating $E[2]^{(2\ 3)}$	Number of Forms
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[2]^{(3\ 0)}$	Number of Forms
on Previous Page: 1		Number of Forms
on the Page: 0		Number of Exact
Forms: 1		Number of
NonClosed Forms: 0	Calculating $E[2]^{(3\ 1)}$	Number of Forms
on Previous Page: 1		Number of Forms
on the Page: 0		

Forms: 1		Number of Exact
NonClosed Forms: 0		Number of
on Previous Page: 1	Calculating $E[2]^{(3\ 2)}$	Number of Forms
on the Page: 0		Number of Forms
Forms: 1		Number of Exact
NonClosed Forms: 0		Number of
on Previous Page: 1	Calculating $E[2]^{(3\ 3)}$	Number of Forms
on the Page: 1		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
Calculating Page 3		
on Previous Page: 0	Calculating $E[3]^{(0\ 0)}$	Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
on Previous Page: 0	Calculating $E[3]^{(0\ 1)}$	Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
on Previous Page: 0	Calculating $E[3]^{(0\ 2)}$	Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
on Previous Page: 0	Calculating $E[3]^{(0\ 3)}$	Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact

NonClosed Forms: 0	Calculating $E[3]^{(1\ 0)}$	Number of
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[3]^{(1\ 1)}$	Number of
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[3]^{(1\ 2)}$	Number of
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[3]^{(1\ 3)}$	Number of
on Previous Page: 1		Number of Forms
on the Page: 1		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[3]^{(2\ 0)}$	Number of
on Previous Page: 1		Number of Forms
on the Page: 1		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[3]^{(2\ 1)}$	Number of
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[3]^{(2\ 2)}$	Number of

on Previous Page: 2	Number of Forms
on the Page: 2	Number of Forms
Forms: 0	Number of Exact
NonClosed Forms: 0	Number of
Calculating $E[3]^{(2\ 3)}$	
on Previous Page: 2	Number of Forms
on the Page: 2	Number of Forms
Forms: 0	Number of Exact
NonClosed Forms: 0	Number of
Calculating $E[3]^{(3\ 0)}$	
on Previous Page: 0	Number of Forms
on the Page: 0	Number of Forms
Forms: 0	Number of Exact
NonClosed Forms: 0	Number of
Calculating $E[3]^{(3\ 1)}$	
on Previous Page: 0	Number of Forms
on the Page: 0	Number of Forms
Forms: 0	Number of Exact
NonClosed Forms: 0	Number of
Calculating $E[3]^{(3\ 2)}$	
on Previous Page: 0	Number of Forms
on the Page: 0	Number of Forms
Forms: 0	Number of Exact
NonClosed Forms: 0	Number of
Calculating $E[3]^{(3\ 3)}$	
on Previous Page: 1	Number of Forms
on the Page: 1	Number of Forms
Forms: 0	Number of Exact
NonClosed Forms: 0	Number of

Pages := Book

(4.2.1)

Taking the answer and calculating more pages

alg2 > Page2 := SHSSLAWC2(Pages, NumberOfPages=2);

Calculating Page 4	Calculating $E[4]^{(0\ 0)}$	Number of Forms
on Previous Page: 0		Number of Forms
on the Page: 0		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[4]^{(0\ 1)}$	Number of Forms
on Previous Page: 0		Number of Forms
on the Page: 0		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[4]^{(0\ 2)}$	Number of Forms
on Previous Page: 0		Number of Forms
on the Page: 0		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[4]^{(0\ 3)}$	Number of Forms
on Previous Page: 0		Number of Forms
on the Page: 0		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[4]^{(1\ 0)}$	Number of Forms
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[4]^{(1\ 1)}$	Number of Forms
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[4]^{(1\ 2)}$	Number of Forms

on Previous Page: 2		Number of Forms
on the Page: 2		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[4]^{(1\ 3)}$	Number of Forms
on Previous Page: 1		Number of Forms
on the Page: 1		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[4]^{(2\ 0)}$	Number of Forms
on Previous Page: 1		Number of Forms
on the Page: 1		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[4]^{(2\ 1)}$	Number of Forms
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[4]^{(2\ 2)}$	Number of Forms
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[4]^{(2\ 3)}$	Number of Forms
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Exact
Forms: 0		Number of
NonClosed Forms: 0	Calculating $E[4]^{(3\ 0)}$	Number of Forms
on Previous Page: 0		Number of Forms
on the Page: 0		

Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
on Previous Page: 0	Calculating $E[4]^{(3\ 1)}$	Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
on Previous Page: 0	Calculating $E[4]^{(3\ 2)}$	Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
on Previous Page: 1	Calculating $E[4]^{(3\ 3)}$	Number of Forms
on the Page: 1		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
Calculating Page 5		
on Previous Page: 0	Calculating $E[5]^{(0\ 0)}$	Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
on Previous Page: 0	Calculating $E[5]^{(0\ 1)}$	Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
on Previous Page: 0	Calculating $E[5]^{(0\ 2)}$	Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact

NonClosed Forms: 0	Calculating $E[5]^{(0\ 3)}$	Number of
on Previous Page: 0		Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[5]^{(1\ 0)}$	Number of
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[5]^{(1\ 1)}$	Number of
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[5]^{(1\ 2)}$	Number of
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[5]^{(1\ 3)}$	Number of
on Previous Page: 1		Number of Forms
on the Page: 1		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[5]^{(2\ 0)}$	Number of
on Previous Page: 1		Number of Forms
on the Page: 1		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0	Calculating $E[5]^{(2\ 1)}$	Number of

on Previous Page: 2		Number of Forms
on the Page: 2		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
	Calculating $E[5]^{(2\ 2)}$	
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
	Calculating $E[5]^{(2\ 3)}$	
on Previous Page: 2		Number of Forms
on the Page: 2		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
	Calculating $E[5]^{(3\ 0)}$	
on Previous Page: 0		Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
	Calculating $E[5]^{(3\ 1)}$	
on Previous Page: 0		Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
	Calculating $E[5]^{(3\ 2)}$	
on Previous Page: 0		Number of Forms
on the Page: 0		Number of Forms
Forms: 0		Number of Exact
NonClosed Forms: 0		Number of
	Calculating $E[5]^{(3\ 3)}$	
on Previous Page: 1		Number of Forms
		Number of Forms

on the Page: 1

Forms: 0

NonClosed Forms: 0

Number of Exact

Number of

$Page2 := Book$

(4.2.2)

`alg2 > eval(Page2) :`

Checking Answer

Computer's answer

`alg2 > RC := RelativeChains([]) :`

`alg2 > C := Cohomology(RC) ;`

$C := [[\beta_2, \beta_1], [\alpha_2 \wedge \beta_1 + \beta_2 \wedge \beta_3, -\alpha_1 \wedge \beta_3 + \alpha_2 \wedge \beta_2, \alpha_1 \wedge \beta_2 + \alpha_3 \wedge \beta_1], [\alpha_1 \wedge \alpha_2 \wedge \beta_1 - \alpha_3 \wedge \beta_1 \wedge \beta_3, \alpha_1 \wedge \beta_2 \wedge \beta_3 + \alpha_3 \wedge \beta_1 \wedge \beta_3, -\alpha_1 \wedge \alpha_2 \wedge \beta_2 + \alpha_2 \wedge \alpha_3 \wedge \beta_1, -\alpha_1 \wedge \alpha_2 \wedge \beta_2 + \alpha_3 \wedge \beta_2 \wedge \beta_3], [\alpha_1 \wedge \alpha_2 \wedge \beta_2 \wedge \beta_3 - \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_3, \alpha_1 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_3 - \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2, -\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 + \alpha_1 \wedge \alpha_3 \wedge \beta_2 \wedge \beta_3], [\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_3, \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_2 \wedge \beta_3], [\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3]]$ (4.3.1)

My answer

`alg2 > Answer := ReadCohomology(Pages[4]) ;`

$Answer := table([0 = [], 1 = [\beta_1, \beta_2], 2 = [\alpha_1 \wedge \beta_2 + \alpha_3 \wedge \beta_1, \alpha_1 \wedge \beta_3 - \alpha_2 \wedge \beta_2, \beta_2 \wedge \beta_3], 3 = [\alpha_1 \wedge \alpha_2 \wedge \beta_2 - \alpha_3 \wedge \beta_2 \wedge \beta_3, \alpha_2 \wedge \alpha_3 \wedge \beta_1 - \alpha_3 \wedge \beta_2 \wedge \beta_3, \alpha_3 \wedge \beta_1 \wedge \beta_3, \alpha_1 \wedge \beta_2 \wedge \beta_3], 4 = [\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 - \alpha_1 \wedge \alpha_3 \wedge \beta_2 \wedge \beta_3, \alpha_1 \wedge \alpha_2 \wedge \beta_2 \wedge \beta_3, \alpha_1 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_3], 5 = [\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_2 \wedge \beta_3, \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_3], 6 = [\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3], PageInfo = table([NumberOfSquares = 16, PageNumber = 4, SizeOfSubAlgebra = 3, SizeOfComplementarySpace = 3])])$ (4.3.2)

Let's test if the 2-forms cohomology class

`alg2 > Answer[2] ;`

$[\alpha_1 \wedge \beta_2 + \alpha_3 \wedge \beta_1, \alpha_1 \wedge \beta_3 - \alpha_2 \wedge \beta_2, \beta_2 \wedge \beta_3]$ (4.3.3)

`alg2 > TCA(Answer[2], C[2]) ;`

"true" (4.3.4)

The Dimensions are the same

`alg2 > map(nops, C) ;`

$[2, 3, 4, 3, 2, 1]$ (4.3.5)

`alg2 > map(nops, ReadCohomology(Pages[2])) ;`

$table([0 = 0, 1 = 2, 2 = 3, 3 = 4, 4 = 3, 5 = 2, 6 = 1, PageInfo = 2])$ (4.3.6)

5.4 *Relative Chains*

This section will demonstrate the program calculating the cohomology relative to elements of the Lie algebra.

Relative Chains

```
alg2 > restart:
> with(Library);
    [Browse, DGplot, DGshow, MetricSearch, References, Retrieve, Search] (5.1.1)
```

```
alg1 > infolevel[SHSSLAWC] := 0;
                                infolevelSHSSLAWC := 0 (5.1.2)
```

Setting up the Lie algebra

```
> LD:=Retrieve("Winternitz",1,[6,19],alg1);
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3] (5.1.3)
      ]=e6, [e2, e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0,
      [e4, e5]=0, [e4, e6]=0, [e5, e6]=0
```

```
> DGsetup(LD);
                                Lie algebra: alg1 (5.1.4)
```

```
> MultiplicationTable();
```

alg1	<i>e1</i>	<i>e2</i>	<i>e3</i>	<i>e4</i>	<i>e5</i>	<i>e6</i>
<i>e1</i>	0	<i>e3</i>	<i>e4</i>	<i>e5</i>	<i>e6</i>	0
<i>e2</i>	- <i>e3</i>	0	<i>e6</i>	0	0	0
<i>e3</i>	- <i>e4</i>	- <i>e6</i>	0	0	0	0
<i>e4</i>	- <i>e5</i>	0	0	0	0	0
<i>e5</i>	- <i>e6</i>	0	0	0	0	0
<i>e6</i>	0	0	0	0	0	0

Finding the Subalgebra

```
alg1 > Query([e4,e5,e6],"Subalgebra");
                                true (5.1.6)
```

```
alg1 > Query([e4,e5,e6],"Ideal");
                                true (5.1.7)
```

Renaming the algebra into our alpha beta convention

```
alg1 > B:=[e4,e5,e6,e1,e2,e3]
                                B := [e4, e5, e6, e1, e2, e3] (5.1.8)
```

```
alg1 > DGEEnvironment[LieAlgebra](B,vectorlabels=[x1,x2,x3,
y1,y2,y3],formlabels=[alpha1,alpha2,alpha3,beta1,
beta2,beta3],alg2);
                                Lie algebra: alg2 (5.1.9)
```

```
alg2 > MultiplicationTable();
```

5.5 *Testing Form Range*

This section will test the form range function of the Serre Hochschild spectral sequence. In Form range will calculate the sequence with respect to a range of forms. It is useful if you aren't interested in calculating the complete cohomology of a Lie algebra but just the cohomology of Λ^k .

Form Range

[alg2 > restart:

```
> with(Library);
    [Browse, DGplot, DGshow, MetricSearch, References, Retrieve, Search] (5.2.1.1)
```

```
> LD:=Retrieve("Winternitz",1,[6,19],alg1);
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3] (5.2.1.2)
      ]=e6, [e2, e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]
      ]=0, [e4, e5]=0, [e4, e6]=0, [e5, e6]=0
```

```
> DGsetup(LD);
                                Lie algebra: alg1 (5.2.1.3)
```

```
> MultiplicationTable();
```

alg1	e1	e2	e3	e4	e5	e6
e1	0	e3	e4	e5	e6	0
e2	-e3	0	e6	0	0	0
e3	-e4	-e6	0	0	0	0
e4	-e5	0	0	0	0	0
e5	-e6	0	0	0	0	0
e6	0	0	0	0	0	0

(5.2.1.4)

Rearranging and relabeling the elements of the Lie algebra.

The subalgebra is x1, x2, x3

The complementary space is y1, y2, y3

The dual to the subalgebra is alpha1, alpha2, alpha3

The dual to the complementary space is beta1, beta2, beta3

```
alg1 > B:=[e4,e5,e6,e1,e2,e3]:DGEnvironment[LieAlgebra]
      (B,vectorlabels=[x1,x2,x3,y1,y2,y3],formlabels=
      [alpha1,alpha2,alpha3,beta1,beta2,beta3],alg2);
                                Lie algebra: alg2 (5.2.1.5)
```

```
alg2 > MultiplicationTable();
```

(5.2.1.6)

alg2	x1	x2	x3	y1	y2	y3
x1	0	0	0	$-x2$	0	0
x2	0	0	0	$-x3$	0	0
x3	0	0	0	0	0	0
y1	$x2$	$x3$	0	0	$y3$	$x1$
y2	0	0	0	$-y3$	0	$x3$
y3	0	0	0	$-x1$	$-x3$	0

(5.2.1.6)

```
alg2 > Annihilator([x1,x2,x3]);
           [beta1, beta2, beta3]
```

(5.2.1.7)

```
alg2 > DisplayStructureEquations();
d alpha1 = -beta1 ^ beta3
d alpha2 = alpha1 ^ beta1
d alpha3 = alpha2 ^ beta1 - beta2 ^ beta3
d beta1 = 0 alpha1 ^ alpha2
d beta2 = 0 alpha1 ^ alpha2
d beta3 = -beta1 ^ beta2
```

(5.2.1.8)

The dual to the subalgebra

```
alg2 > Alpha:=evalDG([alpha1,alpha2,alpha3]);
           A := [alpha1, alpha2, alpha3]
```

(5.2.1)

The dual to the complementary space

```
alg2 > Beta1:=ComplementaryBasis(Alpha);
           B1 := [beta1, beta2, beta3]
```

(5.2.2)

Calculating the spectral sequence from 1-forms to 3-forms. This will give us the cohomology of the 1-forms and 2-forms.

```
alg2 > Test2:=SHSSLAWC2(Alpha,Beta1,FormRange=[1,3],
           NumberOfPages=3);
           Test2 := Book
```

(5.2.3)

The answer we got

```
alg2 > RC := ReadCohomology(Test2[3]);
RC := table([1=[beta1, beta2], 2=[alpha1 ^ beta2 + alpha3 ^ beta1, alpha1 ^ beta3 - alpha2 ^ beta2, beta2 ^ beta3], 3
           = [alpha1 ^ alpha2 ^ alpha3, alpha1 ^ alpha2 ^ beta2, alpha1 ^ alpha2 ^ beta3, alpha1 ^ alpha3 ^ beta2, alpha1 ^ alpha3 ^ beta3, alpha2
           ^ alpha3 ^ beta1, alpha2 ^ alpha3 ^ beta2, alpha2 ^ alpha3 ^ beta3, alpha1 ^ beta2 ^ beta3, alpha2 ^ beta2 ^ beta3, alpha3 ^ beta1
           ^ beta3, alpha3 ^ beta2 ^ beta3], PageInfo=table([SizeOfComplementarySpace=3,
           SizeOfSubAlgebra=3, NumberOfSquares=16, PageNumber=3])])
```

(5.2.4)

Testing our answer

```
alg2 > RC2 := RelativeChains([]):  
alg2 > Coho2 := Cohomology(RC2):
```

Our answer for 1-forms is the same

```
alg2 > Coho2[1];  
[ $\beta_2, \beta_1$ ] (5.2.5)
```

Our answer for the 2-forms is close but not obviously the same. We will not check to see if they are equivalent

```
alg2 > Coho2[2];  
[ $\alpha_2 \wedge \beta_1 + \beta_2 \wedge \beta_3, -\alpha_1 \wedge \beta_3 + \alpha_2 \wedge \beta_2, \alpha_1 \wedge \beta_2 + \alpha_3 \wedge \beta_1$ ] (5.2.6)
```

```
alg2 > RC[2];  
[ $\alpha_1 \wedge \beta_2 + \alpha_3 \wedge \beta_1, \alpha_1 \wedge \beta_3 - \alpha_2 \wedge \beta_2, \beta_2 \wedge \beta_3$ ] (5.2.7)
```

The are equivalent

```
alg2 > TCA(RC[2],Coho2[2]);  
"true" (5.2.8)
```


5.6 *Toolkit testing*

Here we will test several external functions for the spectral sequence.

Tool Kit

```
alg2 > restart:
> with(Library);
    [Browse, DGplot, DGshow, MetricSearch, References, Retrieve, Search] (6.1.1)
```

```
alg1 > infolevel[SHSSLAWC] := 2;
        infolevelSHSSLAWC := 2 (6.1.2)
```

```
> LD:=Retrieve("Winternitz",1,[6,19],alg1);
LD := [e1, e2]=e3, [e1, e3]=e4, [e1, e4]=e5, [e1, e5]=e6, [e1, e6]=0, [e2, e3] (6.1.3)
      ]=e6, [e2, e4]=0, [e2, e5]=0, [e2, e6]=0, [e3, e4]=0, [e3, e5]=0, [e3, e6]=0,
      [e4, e5]=0, [e4, e6]=0, [e5, e6]=0
```

```
> DGsetup(LD);
        Lie algebra: alg1 (6.1.4)
```

```
> MultiplicationTable();
```

alg1	<i>e1</i>	<i>e2</i>	<i>e3</i>	<i>e4</i>	<i>e5</i>	<i>e6</i>
<i>e1</i>	0	<i>e3</i>	<i>e4</i>	<i>e5</i>	<i>e6</i>	0
<i>e2</i>	- <i>e3</i>	0	<i>e6</i>	0	0	0
<i>e3</i>	- <i>e4</i>	- <i>e6</i>	0	0	0	0
<i>e4</i>	- <i>e5</i>	0	0	0	0	0
<i>e5</i>	- <i>e6</i>	0	0	0	0	0
<i>e6</i>	0	0	0	0	0	0

(6.1.5)

Finding the Subalgebra

```
alg1 > Query([e4, e5, e6], "Subalgebra");
        true (6.1.6)
```

```
alg1 > Query([e4, e5, e6], "Ideal");
        true (6.1.7)
```

Renaming the algebra into our alpha beta convention

```
alg1 > B:=[e4, e5, e6, e1, e2, e3]
        B := [e4, e5, e6, e1, e2, e3] (6.1.8)
```

```
alg1 > DGEEnvironment[LieAlgebra](B, vectorlabels=[x1, x2, x3,
        y1, y2, y3], formlabels=[alpha1, alpha2, alpha3, beta1,
        beta2, beta3], alg2);
        Lie algebra: alg2 (6.1.9)
```

```
alg2 > MultiplicationTable();
```

alg2	x1	x2	x3	y1	y2	y3	
x1	0	0	0	$-x2$	0	0	
x2	0	0	0	$-x3$	0	0	
x3	0	0	0	0	0	0	
y1	x2	x3	0	0	y3	x1	
y2	0	0	0	$-y3$	0	x3	
y3	0	0	0	$-x1$	$-x3$	0	(6.1.10)

Checking to see if the complementary space really is the complementary space.

```
alg2 > Annihilator([x1,x2,x3]);
           [beta1, beta2, beta3]
```

(6.1.11)

```
alg2 > DisplayStructureEquations();
      d alpha1 = -beta1 ^ beta3
      d alpha2 = alpha1 ^ beta1
      d alpha3 = alpha2 ^ beta1 - beta2 ^ beta3
      d beta1 = 0 alpha1 ^ alpha2
      d beta2 = 0 alpha1 ^ alpha2
      d beta3 = -beta1 ^ beta2
```

(6.1.12)

```
alg2 > Alpha:=evalDG([alpha1,alpha2,alpha3]);
           A := [alpha1, alpha2, alpha3]
```

(6.1.13)

Define the Complimentary Space

```
alg2 > Beta1:=ComplementaryBasis(Alpha);
           B1 := [beta1, beta2, beta3]
```

(6.1.14)

```
alg2 > Example3 := SHSSLAWC2(Alpha, Beta1, NumberOfPages=
3);
Setting up Sequence
Calculating Page 0
Calculating Page 1
Calculating Page 2
Calculating Page 3
           Example3 := Book
```

(6.1.15)

Calculating chains and projections externally

```
alg2 > testforms := eval(Example3[2][1,2][PageCohomology]
[Forms]);
           testforms := [alpha1 ^ alpha2 ^ beta2, alpha2 ^ alpha3 ^ beta1]
```

(6.1.16)

```
alg2 > Chains := 'Chains';
           Chains := Chains
```

(6.1.17)

```
alg2 > Chain := eval(Example3[2][1,2][PageCohomology]
[Chains]);
```

$$\text{Chain} := [\alpha_1 \wedge \alpha_2 \wedge \beta_2 - \alpha_3 \wedge \beta_2 \wedge \beta_3, \alpha_2 \wedge \alpha_3 \wedge \beta_1 - \alpha_3 \wedge \beta_2 \wedge \beta_3] \quad (6.1.18)$$

$$\begin{aligned} \text{alg2} > \text{Gamma} &:= \text{evalDG}(2 * \text{testforms}[1] - \text{testforms}[2]); \\ \Gamma &:= 2 \alpha_1 \wedge \alpha_2 \wedge \beta_2 - \alpha_2 \wedge \alpha_3 \wedge \beta_1 \end{aligned} \quad (6.1.19)$$

$$\begin{aligned} \text{alg2} > \text{GammaZ} &:= \text{ZigZag}(\text{Gamma}, \text{Example3}[2]); \\ \text{GammaZ} &:= 2 \alpha_1 \wedge \alpha_2 \wedge \beta_2 - \alpha_2 \wedge \alpha_3 \wedge \beta_1 - \alpha_3 \wedge \beta_2 \wedge \beta_3 \end{aligned} \quad (6.1.20)$$

$$\begin{aligned} \text{alg2} > \text{a} &:= \text{SSProjection}(\text{GammaZ}, 1, 2, 3); \\ \text{a} &:= 2 \alpha_1 \wedge \alpha_2 \wedge \beta_2 - \alpha_2 \wedge \alpha_3 \wedge \beta_1 \end{aligned} \quad (6.1.21)$$

$$\begin{aligned} \text{alg2} > \text{b} &:= \text{SSProjection}(\text{GammaZ}, 2, 1, 3); \\ \text{b} &:= -\alpha_3 \wedge \beta_2 \wedge \beta_3 \end{aligned} \quad (6.1.22)$$

$$\begin{aligned} \text{alg2} > \text{ExteriorDerivative}(\text{a}); \\ &\alpha_2 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3 \end{aligned} \quad (6.1.23)$$

$$\begin{aligned} \text{alg2} > \text{ExteriorDerivative}(\text{b}); \\ &-\alpha_2 \wedge \beta_1 \wedge \beta_2 \wedge \beta_3 \end{aligned} \quad (6.1.24)$$

$$\begin{aligned} \text{alg2} > \text{ExteriorDerivative}(\text{Chain}[2]); \\ &0 \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \beta_1 \end{aligned} \quad (6.1.25)$$

5.7 Conclusion

These programs have allowed me to calculate the cohomology of various Lie algebras. I have been able to improve the speed of the current software which calculates the cohomology directly.

```

The Lie algebra so42
so32 > LD2:=SimpleLieAlgebraData("so(4,2)",so42):
Initializing Lie Algebra
so32 > DGsetup(LD2):
Calculating the Relative Chains
so32 > C2:=RelativeChains(so42):
Timing how long my program takes to calculate the Cohomology in seconds
so42 > st:=time():JacCohomology(C2[3]):time()-st;
                                     141.297
(3.1)
Timing how long it takes the maple software to calculate cohomology in seconds
so42 > st:=time():Cohomology(C2[3..5]):time()-st;
                                     1991.688
(3.2)
Almost 14 times faster
so42 > 1991.688/141.297;
                                     14.09575575
(3.3)
The number of minutes each took
so42 > "Maple Code" = evalf(1991.688/60) ; "My code" = 141.297/60 ;
                                     "Maple Code" = 33.19480000
                                     "My code" = 2.354950000
(3.4)

```

By creating the Serre–Hochschild spectral sequence software, I have created a structure in which more speed improvements can be made. These speed improvements can lead to future research on Lie algebras previously considered to large to work with.

BIBLIOGRAPHY

- [1] R. BISHOP AND S. GOLDBERG, *Tensor Analysis On Manifolds*, Dover Publications, Inc., New York, 1980.
- [2] D. FUKS, *Cohomology Of Infinite-Dimensional Lie Algebras*, Springer, New York, 1986.
- [3] S. GOLDBERG, *Curvature and Homology Revised Edition*, Dover Publications, Inc, New York, 1998.
- [4] G. HOCHSCHILD AND J.-P. SERRE, *Cohomology of lie algebras*, Annals Of Mathematics, 57 (1953), pp. 591–603.
- [5] D. LOVELOCK AND H. RUND, *Tensors, Differential Forms, and Variational Principles*, Dover Publications, Inc, New York, 1989.
- [6] J. MCCLEARY, *User's Guide To Spectral Sequences*, Publish or Perish Inc, Delaware, 1985.
- [7] D. SATTINGER AND O. WEAVER, *Lie Groups And Algebras with Applications To Physics, Geometry, And Mechanics*, Springer-Verlag New York Inc, New York, 1986.

APPENDIX

A. SOURCE CODE

A.1 Conversion Programs

A.1.1 FormToList

```
FormToList := proc(X, dimV)
    local n, location, i, j, r;
        n := nops(X);
        if n = 1 then
            return X[1]
        end if;
        location := 0 ;
        r := 0;
        location:= location + binomial( dimV, n)-binomial(dimV- X[1] + 1, n);
        for i from 2 to n - 1 do
            location := location + binomial(dimV - X[i - 1], n - i + 1) - binomial(dimV - X[i] + 1, n - i + 1);
        end do;
        location := location + X[n] - X[n - 1];
    end;
```


A.1.2 *TensorToList*

```

TensorToList := proc(X)
    local Vectors;
    Vectors := map(TensorToList2, X);
end:

TensorToList2:=proc(X)
    local FR, FDim, TensorDegree, VectorPlaceHolder, i, j, Inside, TensorList;
    FR := DGinformation(X, "ObjectFrame");
    FDim := DGinformation(FR, "FrameBaseDimension");
    # This if statement is to rectify the problem with the brackets
    # in the internal makeup of the tensor object.
    # A tensor made up of multiple base elements has one more bracket
    # than a tensor made up of only one base element.
    if nops(op(X)[2]) = 1 then
        TensorList := op(X)[2];
    else
        TensorList := op(op(X)[2]);
    end if;
    TensorDegree := nops(TensorList[1][1]);
    VectorPlaceHolder := Vector(FDim ^ TensorDegree);
    for j by 1 to nops(op(X)[2]) do
        if TensorDegree = 1 then
            VectorPlaceHolder[TensorList[j][1]] := TensorList[j][2];
        else
            VectorPlaceHolder[TensorToList0(TensorList[j][1], FDim)] := TensorList[j][2];
        end if;
    end do;
end:

```

```
    VectorPlaceHolder;
end:

TensorToList0 := proc(X , TensorSize)
    local dim , i , position;
    dim := nops(X);
    position := 0;
    for i by 1 to dim do
        position := position + TensorSize ^ (dim - i) * (X[i] - 1);
    end do;
    position := position + 1;
end:
```

A.1.3 *TensorValuedFormToList*

```

TensorValuedFormToList := proc(X)

  local Vectors;

  Vectors := map(TLACV2, X);

end:

TLACV2 := proc(X)

  local FR, FDim, FormDegree, Size, TensorSize, position, i, j, m, NumberOfForms,
  location, FormPart, PositionList, BaseVector, SizeOfVectorSpace, FormList, RSN;

  FR := DGinformation(X, "ObjectFrame");
  RSN := DGframeData[FR] :- RepresentationSpaceName;
  SizeOfVectorSpace := DGinformation(RSN, "FrameBaseDimension");
  FDim := DGinformation(FR, "FrameBaseDimension");
  FormDegree := DGinformation(X, "FormDegree");

  # op(X)[2] is a list of numbers representing the forms and their coefficients that make up X.
  #An example of this is :[[1, 2], 1], [[3, 4], 12]]
  # This example X consists of two forms added together.
  #The first form, [1,2] has a coefficient of 1 and the second form [3,4] has a coefficient of 12.

  FormList := op(X)[2];
  Size := nops(FormList[1][1]);
  NumberOfForms := nops(FormList);
  TensorSize := Size - FormDegree;
  BaseVector := Vector(SizeOfVectorSpace ^ (TensorSize) * binomial(FDim, FormDegree));

  #The conversion for 0 forms

```

```

if FormDegree = 0 then
  for i by 1 to NumberOfForms do
    position := 0;
    for j by 1 to TensorSize do
      position := position + SizeOfVectorSpace(TensorSize - j) * (FormList[i][1][j] - 1);
    end do;
    BaseVector[position + 1] := FormList[i][2];
  end do;
end if;

# Conversion for 1 forms

if FormDegree = 1 then
  for i by 1 to NumberOfForms do
    position := 0;
    for j by 1 to TensorSize do
      position := position + SizeOfVectorSpace(TensorSize - j) * (FormList[i][1][j] - 1);
    end do;
    location := 0;
    location := location + FormList[i][1][TensorSize + 1];
    position := binomial(FDim, FormDegree) * (position);
    position := position + location;
    BaseVector[position] := FormList[i][2];
  end do;
end if;

# Conversion of k-forms where k>1

```

```
if FormDegree > 1 then
  for i by 1 to NumberOfForms do
    position := 0;
    PositionList := FormList[i][1];
    FormPart := PositionList[TensorSize + 1 .. Size]
    for j by 1 to TensorSize do
      position := position + SizeOfVectorSpace(TensorSize - j) * (PositionList[j] - 1 );
    end do;
    location := 0;
    position := binomial(FDim, FormDegree) * (position );
    position := position + FormToList0(FormPart, FDim);
    BaseVector[position] := FormList[i][2];
  end do;
end if;
BaseVector;
end;
```

A.1.4 Complementary Basis

```

#JacComp

#Input X is a list of vectors. B is basis of 1-forms.

#Output is a list forms corresponding to the complementary basis.

JacComp := proc(X)

    local n , matrixC , i , j , m ,matrixsize , MatrixG , CompSet , genset ,
    EndSet,MatrixF,FR,FDim,FormDegree ;

    FR := DGinformation(X[1], "ObjectFrame");
    FDim := DGinformation(FR, "FrameBaseDimension");
    FormDegree := DGinformation(X[1], "FormDegree");
    n := nops(X) ;
    matrixsize := binomial(FDim , FormDegree);
    CompSet := ;
    MatrixF := FormToMatrix(X);
    MatrixG := LinearAlgebra[GaussianElimination](MatrixF+);
    i := 1;

    for m by 1 to matrixsize do
        if MatrixG[i,m] = 0
            then CompSet := [op(CompSet) , m]
            else i := i + 1;
        end if
    end do;

    EndSet := Array(1..nops(CompSet));

```

```

for i by 1 to nops(CompSet) do
    EndSet[i] := ListToForm0(CompSet[i],FormDegree,FR);
end do;

convert(EndSet,list);
end:

JacComp2 := proc(X , B)

    local n , matrixC , i , j , m ,matrixsize , MatrixG , CompSet , genset , EndSet,
    MatrixF,FR,FDim,FormDegree,XVector,CompareSet, NewBasis, FTL, NewBasisMatrix;

    FR := DGinformation(X[1], "ObjectFrame");
    FDim := DGinformation(FR, "FrameBaseDimension");
    FormDegree := DGinformation(X[1], "FormDegree");
    n := nops(X) ;
    CompSet := ;
    NewBasis := ;
    matrixsize := nops(B);
    MatrixF:=convert(FormToList(B), Matrix);
    FTL := FormToList(X);

    for i by 1 to n do
        NewBasis:= [op(NewBasis), LinearAlgebra:-LinearSolve(MatrixF,FTL[i])];
    end do;

    NewBasisMatrix := Matrix(1 .. nops(B));

```

```

for i by 1 to nops(NewBasis) do
    NewBasisMatrix[i] := NewBasis[i];
end do;

MatrixG := LinearAlgebra[GaussianElimination](NewBasisMatrix^+);
i := 1;

for m by 1 to matrixsize do
    if MatrixG[i,m] = 0
        then CompSet := [op(CompSet) , m]
        else i := i + 1;
    end if
end do;

EndSet := Array(1..nops(CompSet));

for i by 1 to nops(CompSet) do
    EndSet[i] := B[CompSet[i]];
end do;

convert(EndSet,list);
end:

JacCompInternal := proc(X , B)

    local n , matrixC , i , j , m ,matrixsize , MatrixG , CompSet , genset , EndSet,
    MatrixF,FR,FDim,FormDegree,XVector,CompareSet, NewBasis, FTL, NewBasisMatrix;

```

```

FR := DGinformation(X[1], "ObjectFrame");
FDim := DGinformation(FR, "FrameBaseDimension");
FormDegree := DGinformation(X[1], "FormDegree");
n := nops(X) ;
CompSet := ;
NewBasis := ;
matrixsize := nops(B);
MatrixF:=convert(FormToList(B), Matrix);
FTL := FormToList(X);

for i by 1 to n do
    NewBasis:= [op(NewBasis), LinearAlgebra:-LinearSolve(MatrixF,FTL[i])];
end do;

NewBasisMatrix := Matrix(1 .. nops(B));

for i by 1 to nops(NewBasis) do
    NewBasisMatrix[i] := NewBasis[i];
end do;

MatrixG := LinearAlgebra[GaussianElimination](NewBasisMatrix);
i := 1;

for m by 1 to matrixsize do
    if MatrixG[i,m] = 0
        then CompSet := [op(CompSet) , m]
        else i := i + 1;
    end if

```

end do;

EndSet := Array(1..nops(CompSet));

for i by 1 to nops(CompSet) do

 EndSet[i] := B[CompSet[i]];

end do;

convert(EndSet,list);

end:

A.2 Cohomology Programs

A.2.1 Closed Forms

```

Closed := proc(X)

  local Fset, FR, FDim, FDegree, i, j, N, NullSet, k, ClosedSet,
  n, DList, DMatrix, NullSetSize, ExtDer, Value;

  FR := DGinformation(X[1], "ObjectFrame");
  FDim := DGinformation(FR, "FrameBaseDimension");
  FDegree := DGinformation(X[1], "FormDegree");
  N := nops(X);
  ExtDer:=ExteriorDerivative(X);
  DList:= FormToList(ExtDer);
  DMatrix := convert(convert(DList, list), Matrix);
  NullSet := LinearAlgebra[NullSpace](DMatrix);
  NullSetSize := nops(NullSet);
  ClosedSet := Array(1..NullSetSize);
  Value:=0;

  for i by 1 to NullSetSize do
    for j by 1 to N do
      Value:=evalDG(Value + NullSet[i][j]*X[j]);
    end do;
    ClosedSet[i]:=Value;
    Value:=0;
  end do;

```

```
convert(ClosedSet,list);
```

```
end;
```

A.2.2 *Exact Forms*

```

Exact := proc(X)

    local FR, FDim, FormDegree, i, j, N, ColumnSet, k, ExactSet, n, DList, DMatrix,
    ColumnSetSize, ExtDer, Value, VectorSize;

    FR := DGinformation(X[1], "ObjectFrame");
    FDim := DGinformation(FR, "FrameBaseDimension");
    FormDegree := DGinformation(X[1], "FormDegree");
    N := nops(X);
    VectorSize:=binomial(FDim,FormDegree+1);
    ExtDer:=ExteriorDerivative(X);
    DList:= FormToList(ExtDer);
    DMatrix := convert(convert(DList, list), Matrix);
    ColumnSet := LinearAlgebra[ColumnSpace](DMatrix);

    if nops(ColumnSet) > 0
        then
            ExactSet:=ListToForm(ColumnSet , FormDegree+1, FR);
        else
            ExactSet:=[];
        end if;

    ExactSet;

end:

```

A.2.3 Calculating Cohomology

Omega1 is a list of k-forms and Omega2 is a list of k+1 forms.

```
JacCohomology2 := proc(Omega1, Omega2)
```

```
  local B, H, C, W;
```

```
  #B is a basis for the exact forms in k + 1 form coming from Omega1.
```

```
  B := Exact(Omega1);
```

```
  #C is the complement of B in Omega2.
```

```
  C := JacComp2(B, Omega2);
```

```
  #H is the Cohomology in Omega2
```

```
  H := Closed(C);
```

```
  #W
```

```
  W := JacComp2([op(B), op(H)], Omega2);
```

```
  [H, B, W];
```

```
end:
```

```
JacCohomology := proc(Omega1)
```

```
  local B, H, C, W;
```

```
  B := Exact(Omega1);
```

```
  C := JacComp(B);
```

```
if nops(C) = 0
  then
    H := [];
  else
    H := Closed(C);
end if;

W := JacComp([op(B), op(H)]);
[H, B, W];

end;
```

A.3 Serre Hofschild Spectral Sequence Programs

A.3.1 SHSSLAWC2

```

SHSSLAWC2 := proc(SubAlgebra, ComplementarySpace, Page,
NumberOfPages := 1 , VectorSpace := [], FormRange := [], RelativeChain := [] )

    local i, RepNumber, ComplementarySpace1, Book;

    if VectorSpace = [] then
        RepNumber := 0;
    else
        RepNumber :=1;
    end if;

    if _params['Page'] = NULL then
        if _params['SubAlgebra'] = NULL then
            return "error enter a SubAlgebra or a Page";
        else
            if op(SubAlgebra[1])[1][1]= "form" then
                if _params['ComplementarySpace'] = NULL then
                    ComplementarySpace1 := ComplementaryBasis(SubAlgebra);
                    userinfo(2, SHSSLAWC, NoName, "Setting up Sequence");
                    Book := table([ -1 = SpectralSequenceTableLAWC2(SubAlgebra, ComplementarySpace1,
                    RepNumber, VectorSpaceInternal = VectorSpace, FormRangeInternal = FormRange,
                    RelativeChainInternal = RelativeChain )]);
                    for i from 0 by 1 to NumberOfPages do
                        userinfo( 2, SHSSLAWC, NoName, "Calculating Page", i );
                        Book[i] := NextPageTableLAWC(Book[i-1]);
                    end do;
                end if;
            end if;
        end if;
    end if;
end proc;

```

```

        end do;
        Book[PageNumber] := NumberOfPages;
        return Book;
    else
        userinfo(2, SHSSLAWC, NoName, "Setting up Sequence");
        Book := table([-1 =SpectralSequenceTableLAWC2(SubAlgebra, ComplementarySpace,
        RepNumber, VectorSpaceInternal = VectorSpace, FormRangeInternal = FormRange,
        RelativeChainInternal = RelativeChain)]);
        for i from 0 by 1 to NumberOfPages do
            userinfo( 2, SHSSLAWC,NoName, "Calculating Page", i );
            Book[i] := NextPageTableLAWC(Book[i-1]);
        end do;
        Book[PageNumber] := NumberOfPages;
        return Book;
    end if;
else
    Book := SubAlgebra;
    for i from SubAlgebra[PageNumber] + 1 to SubAlgebra[PageNumber] + NumberOfPages do
        userinfo( 2, SHSSLAWC,NoName, "Calculating Page", i );
        Book[i] := NextPageTableLAWC(Book[i-1]);
    end do;
    Book[PageNumber] := SubAlgebra[PageNumber] + NumberOfPages;
    return Book;
end if;
end if;
else
    Book := Page;
    for i from Page[PageNumber] + 1 to Page[PageNumber] + NumberOfPages do

```

```
        userinfo( 2, SHSSLAWC,NoName, "Calculating Page", i );  
        Book[i] := NextPageTableLAWC(Book[i-1]);  
    end do;  
    Book[PageNumber] := Page[PageNumber] + NumberOfPages;  
    return Book;  
end if;  
end;
```

A.3.2 *SequenceTableLAWC2*

```

# Serre Hofschild Spectral Sequence

#PageInfo := [[Page Number, # of Boxes, SubalgebraSize,
ComplementarySpaceSize], [[ [p,q], [Still in Play], [Exact], [Complementary]] ] ]

SpectralSequenceTableLAWC2 := proc(Alpha, Beta, Rep,
VectorSpaceInternal := [], FormRangeInternal := [], RelativeChainInternal := [] )

local i, j, k, m, SizeOfAlpha, SizeOfBeta, SizeOfSquare, SquareIndex, SquareForms,
Diff, MinDegree, MaxDegree, NumberOfSquaresVariable, Page, FormList, ChainList,
ExteriorDerivativeList, LastForm, RC, KFormAmounts, pq;

SizeOfAlpha := nops(Alpha);
SizeOfBeta := nops(Beta);
NumberOfSquaresVariable := (SizeOfAlpha + 1 ) * (SizeOfBeta + 1 );
#Page := table([PageInfo = []]);

if FormRangeInternal = [] then
    MinDegree := -1;
    MaxDegree := SizeOfAlpha + SizeOfBeta + 1;
else
    MinDegree := FormRangeInternal[1] - 1 ;
    MaxDegree := FormRangeInternal[2] + 1 ;
end if;

for i from 0 to SizeOfBeta do
    for j from 0 to SizeOfAlpha do

```

```

if i + j > MinDegree and i + j < MaxDegree then
  Page[i,j] := table([PageCohomology = [], Exact = [], Complement = [] ] );
  Page[i,j][PageCohomology] := table([Forms = [], Chains = [], ChainDerivative = [] ] );
  Page[i,j][Exact] := table([Forms = [], Chains = [], ChainDerivative = [] ] );
  Page[i,j][Complement] := table([Forms = [], Chains = [], ChainDerivative = [] ] );
end if;
end do;
end do;

Page[PageInfo] := table([PageNumber = -1, NumberOfSquares = (SizeOfAlpha + 1 ) * (SizeOfBeta + 1),
SizeOfSubAlgebra = SizeOfAlpha, SizeOfComplementarySpace = SizeOfBeta,
Vectors = nops(VectorSpaceInternal), X = Rep , Min = MinDegree, Max = MaxDegree]);

RC := RelativeChains(RelativeChainInternal);
KFormAmounts := map(nops, RC);

for i by 1 to nops(RC) do
  if i > MinDegree -1 and i < MaxDegree + 1 then
    if i ≤ SizeOfAlpha + SizeOfBeta + 1 then
      for jby 1 to KFormAmounts[i] do
        pq := PQ(RC[i][j], SizeOfAlpha);
        Page[pq[1],pq[2]][PageCohomology][Forms] := [
op (Page[pq[1],pq[2]][PageCohomology][Forms]), RC[i][j]];
        Page[pq[1],pq[2]][PageCohomology][Chains] := [
op (Page[pq[1],pq[2]][PageCohomology][Chains]), RC[i][j]];
        Page[pq[1],pq[2]][PageCohomology][ChainDerivative] := [
op (Page[pq[1],pq[2]][PageCohomology][ChainDerivative]), ExteriorDerivative(RC[i][j]));
      end do;
    end if;
  end if;
end do;

```

```
else
  for j by 1 to KFormAmounts[i] do
    Page[SizeOfBeta,SizeOfAlpha][PageCohomology][Forms] := [
      op( Page[SizeOfBeta,SizeOfAlpha][PageCohomology][Forms]), RC[i][j]);
    Page[SizeOfBeta,SizeOfAlpha][PageCohomology][Chains] := [
      op(Page[SizeOfBeta,SizeOfAlpha][PageCohomology][Chains]), RC[i][j]);
    Page[SizeOfBeta,SizeOfAlpha][PageCohomology][ChainDerivative] := [
      op(Page[SizeOfBeta,SizeOfAlpha][PageCohomology][ChainDerivative]), 0] ;
    end do;
  end if;
end if;
end do;

op(Page);

end:
```

A.3.3 NextPageTableLAWC

q refers to the subalgebra(alpha) p refers to the Complementary space(beta) of the subalgebra

Next Page takes a page and finds the next page in the sequence.

```
NextPageTableLAWC := proc(LastPage1)
```

```

local i,j,k,m, NewP, NewQ, DList, ProjectionList, Omega, FormDecomp, FormDecompList,
FormDecompList2, SHSSChain, SHSSDer, DerivativeList, Place, ConvertList, DMatrix, NullSet,
NullSetSize, ClosedSet, Value, PageSet, ExactSet, CohomologyList, CohoPlaceholderList, CohoChainList,
CohoPlaceHoder, ExterDer,ChainAdd, CompList, CompPlaceholderList, CompPaceholderChainDerList,
ReceivingExactDerList, ExactForm, Chain, CompPlaceHolder, ReceivingExactChainDerList,
ReceivingExactChainList, ReceivingExactList, CompPlaceholderChainList, CohoPlaceHolder,
CohoChainDerList, LastPage,Page,CompPlaceholderChainDerList, ExactTable, ComplementTable,
PageNum, Output;
```

```
#####
```

```
LastPage := LastPage1;
```

```
# Creating the skeleton of the new page
```

```
Page := table([PageInfo = []]);
```

```
ExactTable := table([]);
```

```
ComplementTable := table([]);
```

```
# Initializing the new page object.
```

```

Page[PageInfo] := table([PageNumber = LastPage[PageInfo][PageNumber] + 1,
NumberOfSquares = LastPage[PageInfo][NumberOfSquares],
SizeOfSubAlgebra = LastPage[PageInfo][SizeOfSubAlgebra],
SizeOfComplementarySpace = LastPage[PageInfo][SizeOfComplementarySpace],
Vectors = LastPage[PageInfo][Vectors], X = LastPage[PageInfo][X] ,
Max = LastPage[PageInfo][Max], Min = LastPage[PageInfo][Min]]);

for i from 0 to LastPage[PageInfo][SizeOfComplementarySpace] do
  for j from 0 to LastPage[PageInfo][SizeOfSubAlgebra] do
    if i + j > LastPage[PageInfo][Min] and i + j < LastPage[PageInfo][Max] then
      Page[i,j] := table([PageCohomology = [], Exact = [], Complement = [] ] );
      Page[i,j][PageCohomology] := table([Forms = [], Chains = [], ChainDerivative = [] ] );
      Page[i,j][Exact] := table([Forms = LastPage[i,j][Exact][Forms],
Chains = LastPage[i,j][Exact][Chains],
ChainDerivative = LastPage[i,j][Exact][ChainDerivative] ] );
      Page[i,j][Complement] := table([Forms = [], Chains = [], ChainDerivative = [] ] );
      ExactTable[i,j] := 0;
      ComplementTable[i,j] := 0;
    end if;
  end do;
end do;

```

```
#####
```

```

for i from 0 to LastPage[PageInfo][SizeOfComplementarySpace] do
  for j from 0 to LastPage[PageInfo][SizeOfSubAlgebra] do
    if i + j > LastPage[PageInfo][Min] and i + j < LastPage[PageInfo][Max] then

```

```

NewP := i + Page[PageInfo][PageNumber];
NewQ := j + 1 - Page[PageInfo][PageNumber];
PageNum := (Page[PageInfo][PageNumber]);
Output := E[PageNum]^(i,j);
userinfo( 3, SHSSLAWC,NoName, " Calculating", Output );
userinfo( 4, SHSSLAWC,NoName," Number of Forms on Previous Page:",
nops(LastPage[i,j][PageCohomology][Forms]) );
DList :=Array(1 .. nops(LastPage[i,j][PageCohomology][Forms]));

# Testing if the projection is on the page

if NewP > -1 and NewP ≤ LastPage[PageInfo][SizeOfComplementarySpace] + 1
and NewQ > -1 and NewQ < LastPage[PageInfo][SizeOfSubAlgebra] + 1 and
NewP + NewQ < LastPage[PageInfo][Max] then

#####
# Now this branch deals with the differential projection on the page
# Calculates the projection

ProjectionList := SSProjectionInternalLAWC(
LastPage[i,j][PageCohomology][ChainDerivative],
NewP, NewQ, Page[PageInfo][SizeOfSubAlgebra], LastPage[PageInfo][X]);

# Now calculating the differential

for k by 1 to nops(LastPage[i,j][PageCohomology][Forms]) do
  if ProjectionList[k] = [] then
    DList[k] := Vector(1 .. nops(LastPage[NewP,
```

```

NewQ][PageCohomology][Forms]));
else
if ProjectionList[k][1][2] = 0 then
    DList[k] := Vector(1 .. nops(LastPage[NewP,
NewQ][PageCohomology][Forms]));
else
if LastPage[NewP, NewQ][PageCohomology][Forms] = [] then
    Omega := _DG([op(LastPage[NewP,
NewQ][Exact][Forms][1])[1], ProjectionList[k]));
else
    Omega := _DG([op(LastPage[NewP,
NewQ][PageCohomology][Forms][1])[1], ProjectionList[k]));
end if;
FormDecomp := SHSSDecompLAWC(Omega , LastPage[NewP,
NewQ][PageCohomology][Forms],nops(LastPage[NewP,
NewQ][PageCohomology][Forms]),
LastPage[NewP, NewQ][Exact][Forms],
nops(LastPage[NewP, NewQ][Exact][Forms]),
LastPage[NewP, NewQ][Complement][Forms],
nops(LastPage[NewP, NewQ][Complement][Forms]),
LastPage[PageInfo][Vectors]) ;
FormDecompList := convert(FormDecomp[1],list);
FormDecompList2 := convert(FormDecomp[2], list);
if ZeroVector(FormDecompList, nops(LastPage[NewP,
NewQ][PageCohomology][Forms])) = true then
    SHSSChain := LastPage[i, j][PageCohomology][Chains][k];
    SHSSDer := LastPage[i, j][PageCohomology][ChainDerivative][k];
    for m by 1 to nops(LastPage[NewP, NewQ][Exact][Forms]) do

```

```

SHSSChain := evalDG( SHSSChain - FormDecompList2[m]
* LastPage[NewP, NewQ][Exact][Chains][m]);
SHSSDer:= evalDG(SHSSDer - FormDecompList2[m]
* LastPage[NewP, NewQ][Exact][ChainDerivative][m]);
end do;
LastPage[i,j][PageCohomology][Chains][k] := SHSSChain;
LastPage[i,j][PageCohomology][ChainDerivative][k] := SHSSDer;
end if;
DList[k] := FormDecomp[1];
end if;
end if;
end do; # k loop

# First I want to save what each dr of each basis form was.
# That way I don't have to recalculate the dr of the complementary space.

DerivativeList := Array(1 .. nops(LastPage[i,j][PageCohomology][Forms]));

for k by 1 to nops(LastPage[i,j][PageCohomology][Forms]) do
Place := 0;
for m by 1 to nops(LastPage[NewP, NewQ][PageCohomology][Forms]) do
LastPage[NewP, NewQ][PageCohomology][Forms][m];
Place:= evalDG( Place + DList[k][m] * LastPage[NewP,
NewQ][PageCohomology][Forms][m]);
end do;
DerivativeList[k] := Place;
end do;
end do;

```

```

# Now that I have the d(omega) I can calculate the closed forms by
# calculating the null space

```

```

ConvertList := convert(DList,list);
DMatrix := convert(ConvertList, Matrix);
NullSet := LinearAlgebra[NullSpace](DMatrix);
NullSetSize := nops(NullSet);
ClosedSet := Array(1..NullSetSize);
Value:=0;

```

```

for k by 1 to NullSetSize do
    for m by 1 to nops(LastPage[i,j][PageCohomology][Forms]) do
        Value:=evalDG(Value + NullSet[k][m] *LastPage[i,
            j][PageCohomology][Forms][m]);
    end do;
    ClosedSet[k]:=Value;
    Value:=0;
end do;

```

```

ClosedSet := convert(ClosedSet,list);

```

```

#####

```

```

else

```

```

# The differential is off the page so the entire box is closed

```

```

ClosedSet := LastPage[i,j][PageCohomology][Forms];

```

```

end if;

#####
# end the on page or not if statement
#####

# Now that I have the list of closed forms I can calculate the cohomology.
# Which is the complementary basis of the exact forms in the space of closed.

PageSet := LastPage[i,j][PageCohomology][Forms];
ExactSet := Page[i,j][Exact][Forms];

if ClosedSet = [] then
    CohomologyList := [];
else
    CohomologyList := ComplementaryBasis(ExactSet,ClosedSet);
end if;

CohoPlaceholderList := Array(1 .. nops (CohomologyList));
CohoChainList := Array(1 .. nops(CohomologyList));
CohoChainDerList := Array(1 .. nops(CohomologyList));

for k by 1 to nops(CohomologyList) do
    CohoPlaceholder := SHSSDecompLAWC(CohomologyList[k],
    PageSet, nops(PageSet), [],0,[],0, LastPage[PageInfo][Vectors]);
    ExterDer := 0;
    ChainAdd := 0;
    for m by 1 to nops(PageSet) do

```

```

ChainAdd := evalDG(ChainAdd +
LastPage[i,j][PageCohomology][Chains][m] * CohoPlaceholder[1][m] );
ExterDer := evalDG(ExterDer +
LastPage[i,j][PageCohomology][ChainDerivative][m] * CohoPlaceholder[1][m]);
end do;

CohoPlaceholderList[k] := CohomologyList[k];
CohoChainList[k] := ChainAdd;
CohoChainDerList[k] := ExterDer;
end do;

Page[i,j][PageCohomology][Forms] := convert(CohoPlaceholderList, list);
Page[i,j][PageCohomology][Chains] := convert(CohoChainList,list);
Page[i,j][PageCohomology][ChainDerivative] := convert(CohoChainDerList, list);
# The complementary basis which I will use to calculate the exact forms
in the box it goes into.

if NewP > -1 and NewP ≤ LastPage[PageInfo][SizeOfComplementarySpace] + 1
and NewQ > -1 and NewQ < LastPage[PageInfo][SizeOfSubAlgebra] + 1
and NewQ + NewP < LastPage[PageInfo][Max] then
  CompList:= ComplementaryBasis(ClosedSet, PageSet);
  CompPlaceholderList := Array(1 .. nops (CompList));
  CompPlaceholderChainList := Array(1 .. nops(CompList));
  CompPlaceholderChainDerList := Array(1 .. nops(CompList));
  ReceivingExactList :=Array(1 .. nops(CompList));
  ReceivingExactChainList := Array(1 .. nops(CompList));
  ReceivingExactChainDerList := Array( 1.. nops (CompList));
  for k by 1 to nops(CompList) do

```

```

CompPlaceholder := SHSSDecompLAWC(CompList[k], PageSet,
nops(PageSet), [],0,[],0, LastPage[PageInfo][Vectors]);
ExterDer := 0;
Chain := 0;
ExactForm := 0;
for m by 1 to nops(PageSet) do
    Chain := evalDG(Chain+LastPage[i,j][PageCohomology][Chains][m] *
    CompPlaceholder[1][m]);
    ExterDer := evalDG(ExterDer + LastPage[i,
j][PageCohomology][ChainDerivative][m] * CompPlaceholder[1][m]);
    ExactForm := evalDG(ExactForm + CompPlaceholder[1][m] *
    DerivativeList[m]);
end do;
# end m loop
CompPlaceholderList[k] := CompList[k];
CompPlaceholderChainList[k] := Chain;
CompPlaceholderChainDerList[k] := ExterDer;

# Need to find out what the Exterior Derivative of the CompList is now.
ReceivingExactList[k] := ExactForm;
ReceivingExactChainList[k] := Chain;
ReceivingExactDerList[k] := ExterDer;
end do;
# end k loop
Page[i,j][Complement][Forms] := convert(CompPlaceholderList,list);
Page[i,j][Complement][Chains] := convert(CompPlaceholderChainList, list);
Page[i,j][Complement][ChainDerivative] := convert(CompPlaceholderChainDerList,
list);

```

```

ComplementTable[i,j] := nops(convert(CompPlaceholderList,list));
if nops(CompList) > 0 then
    Page[NewP, NewQ][Exact][Forms] := [op(LastPage[NewP,
    NewQ][Exact][Forms]), op(convert(ReceivingExactList, list))];
    Page[NewP, NewQ][Exact][Chains] :=[op(LastPage[NewP,
    NewQ][Exact][Chains]), op( convert(ReceivingExactChainList, list))];
    Page[NewP, NewQ][Exact][ChainDerivative] := [op(LastPage[NewP,
    NewQ][Exact][ChainDerivative]),op(convert(ReceivingExactDerList, list))];
    ExactTable[NewP,NewQ] := nops(convert(ReceivingExactList, list));
    end if;
end if;
end if;

userinfo( 4, SHSSLAWC,NoName, " Number of Forms on the Page:",
nops(Page[i,j][PageCohomology][Forms]) );
userinfo( 4, SHSSLAWC,NoName, " Number of Exact Forms:",
ExactTable[i,j] );
userinfo( 4, SHSSLAWC,NoName, " Number of NonClosed Forms:",
ComplementTable[i,j] );
end do;
# j loop;
end do; # i loop;
op(Page);

end:

```

A.4 Serre Hofschild Spectral Sequence Toolkit

A.4.1 Read Table

```

ReadTable := proc(Page)

    local i,j,T;
    T := table([PageInfo = []]);
    T[PageInfo] := table([PageNumber = Page[PageInfo][PageNumber],
    NumberOfSquares = Page[PageInfo][NumberOfSquares],
    SizeOfSubAlgebra = Page[PageInfo][SizeOfSubAlgebra],
    SizeOfComplementarySpace = Page[PageInfo][SizeOfComplementarySpace] ]);

    for i from 0 by 1 to Page[PageInfo][SizeOfComplementarySpace] do
        for j from 0 by 1 to Page[PageInfo][SizeOfSubAlgebra] do
            if i + j > Page[PageInfo][Min] and i + j < Page[PageInfo][Max] then
                T[i,j] := Page[i,j][PageCohomology][Forms];
            end if;
        end do;
    end do;

    op(T);

end:

```


A.4.2 Read Exact

```

ReadExact := proc(Page)

    local i,j, T;
    T := table([PageInfo = []]);
    T[PageInfo] := table([PageNumber = Page[PageInfo][PageNumber],
    NumberOfSquares = Page[PageInfo][NumberOfSquares],
    SizeOfSubAlgebra = Page[PageInfo][SizeOfSubAlgebra],
    SizeOfComplementarySpace = Page[PageInfo][SizeOfComplementarySpace] ]);

    for i from 0 by 1 to Page[PageInfo][SizeOfComplementarySpace] do
        for j from 0 by 1 to Page[PageInfo][SizeOfSubAlgebra] do
            if i + j > Page[PageInfo][Min] and i + j < Page[PageInfo][Max] then
                T[i,j] := Page[i,j][Exact][Forms];
            end if;
        end do;
    end do;

    op(T);

end:

```

A.4.3 Read Complement

```

ReadComplement := proc(Page)

    local i,j, T;
    T := table([PageInfo = []]);
    T[PageInfo] := table([PageNumber = Page[PageInfo][PageNumber],
    NumberOfSquares = Page[PageInfo][NumberOfSquares],
    SizeOfSubAlgebra = Page[PageInfo][SizeOfSubAlgebra],
    SizeOfComplementarySpace = Page[PageInfo][SizeOfComplementarySpace] ]);

    for i from 0 by 1 to Page[PageInfo][SizeOfComplementarySpace] do
        for j from 0 by 1 to Page[PageInfo][SizeOfSubAlgebra] do
            if i + j > Page[PageInfo][Min] and i + j < Page[PageInfo][Max] then
                T[i,j] := Page[i,j][Complement][Forms];
            end if;
        end do;
    end do;

    op(T);

end:

```

A.4.4 Read Cohomology

```

ReadCohomology := proc(Page)

    local i,j,T, CohoArray;
    T := table([PageInfo = []]);
    T[PageInfo] := table([PageNumber = Page[PageInfo][PageNumber],
    NumberOfSquares = Page[PageInfo][NumberOfSquares],
    SizeOfSubAlgebra = Page[PageInfo][SizeOfSubAlgebra],
    SizeOfComplementarySpace = Page[PageInfo][SizeOfComplementarySpace] ]);
    CohoArray := [ seq([], i = 0 .. (Page[PageInfo][SizeOfComplementarySpace] +
    Page[PageInfo][SizeOfSubAlgebra]))];

    for i from 0 by 1 to Page[PageInfo][SizeOfComplementarySpace] do
        for j from 0 by 1 to Page[PageInfo][SizeOfSubAlgebra] do
            CohoArray[i+j + 1] :=[op(CohoArray[i+j + 1]),op( Page[i,j][PageCohomology][Chains]) ];
        end do;
    end do;

    for i from 0 by 1 to (Page[PageInfo][SizeOfComplementarySpace] + Page[PageInfo][SizeOfSubAlgebra]) do
        if i > Page[PageInfo][Min] and i < Page[PageInfo][Max] then
            T[i] := CohoArray[i + 1];
        end if;
    end do;

    op(T);

end:

```

A.4.5 ZigZag

```

ZigZag := proc(Sigma, Page)

  local i, pq, FormDecomp, Chain, PageSet, ChainSet;
  pq := PQ(Sigma, Page[PageInfo][SizeOfSubAlgebra]);
  PageSet := Page[pq[1],pq[2]][PageCohomology][Forms];
  ChainSet := Page[pq[1], pq[2]][PageCohomology][Chains];
  FormDecomp := SHSSDecomp(Sigma, PageSet, nops(PageSet), [],0,[],0);
  Chain := 0;

  for i by 1 to nops(PageSet) do
    Chain := evalDG( Chain + FormDecomp[1][i] * ChainSet[i]);
  end do;

  Chain[1];

end:

```

A.4.6 SSprojection

```
SSProjection := proc(Sigma, P,Q , SizeOfSubAlgebra)

    local i, q, m, pq, NewQ, NewP, EdZigzag;
    NewQ := Q;
    NewP := P;
    m := SSProjectionInternal([Sigma],NewP,NewQ, SizeOfSubAlgebra);

    if op(m) = [] then
        return [];
    else
        q := _DG([op(Sigma)[1],op(m)]);
    end if;

    q;

end;
```

A.4.7 TCA

```
#Testing Cohomology Answers TCA
#JAnswer is the answer given from JacCohomology
#CAnswer is the answer given from Cohomology
TCA := proc(JAnswer, CAnswer)

    local Alpha , Beta, answer, sizeOfJAnswer, i;
    sizeOfJAnswer := nops(JAnswer);
    for i by 1 to sizeOfJAnswer do
        Alpha, Beta := CohomologyDecomposition(JAnswer[i], CAnswer);
        answer := evalDG(JAnswer[i] - (Alpha + ExteriorDerivative(Beta)));
        if op(op(answer)[2])[2] = 0 then
            else
                return "false"
            end if;
        end do;
        return "true"
    end;
```

A.4.8 SSExtDer

Maple Code

```

SSExtDer:= proc(Sigma, Page)

    local i, pq, Zig, proj, NewQ, NewP, Proj, FormDecomp, FormValue;

    pq := PQ(Sigma, Page[PageInfo][SizeOfSubAlgebra]);
    NewQ := pq[2] + 1 - Page[PageInfo][PageNumber] - 1 ;
    NewP := pq[1] + Page[PageInfo][PageNumber] + 1;
    Zig := ZigZag(Sigma, Page);

    if NewP > -1 and NewP < Page[PageInfo][SizeOfComplementarySpace] + 1 and
    NewQ > -1 and NewQ < Page[PageInfo][SizeOfSubAlgebra] + 1 then
        Proj := SSProjection(Sigma, Zig, Page);
        if Proj = [] then
            return [];
        else
            FormDecomp := SHSSDecomp(Proj, Page[NewP, NewQ][PageCohomology][Forms],
            nops(Page[NewP, NewQ][PageCohomology][Forms]),
            Page[NewP, NewQ][Exact][Forms], nops(Page[NewP, NewQ][Exact][Forms]),
            Page[NewP, NewQ][Complement][Forms], nops(Page[NewP, NewQ][Complement][Forms])) ;
            FormValue := 0;
            for i by 1 to nops(Page[NewP, NewQ][PageCohomology][Forms]) do
                FormValue := evalDG(FormValue + FormDecomp[1][i] *
                Page[NewP, NewQ][PageCohomology][Forms][i]);
            end do;
        end if;
    else

```

```
        return [];  
    end if;  
  
    FormValue;  
  
end:
```


A.5 Auxiliary Functions

A.5.1 SSProjectionInternal

$$\#d_r = E_r^{p,q} - > E_r^{p+r,q-r+1}$$

```
SSProjectionInternal:=proc(ExteriorDerivativeList,NewP,NewQ,SizeOfAlpha)
```

```
  local ListSize, i, j,ModuloList,counter,EndList;
```

```
  ListSize := nops(ExteriorDerivativeList);
```

```
  if NewQ < NewQ + NewP then
```

```
    if NewQ < 1 then
```

```
      for i by 1 to ListSize do
```

```
        ModuloList := [];
```

```
        for j by 1 to nops(op(ExteriorDerivativeList[i])[2]) do
```

```
          if op(ExteriorDerivativeList[i])[2][j][1][1] < SizeOfAlpha then
```

```
            ModuloList := [op(ModuloList), op(ExteriorDerivativeList[i])[2][j]];

```

```
          end if;
```

```
        end do;
```

```
        EndList[i] := ModuloList;
```

```
      end do;
```

```
    else
```

```
      for i by 1 to ListSize do
```

```
        ModuloList := [];
```

```
        for j by 1 to nops(op(ExteriorDerivativeList[i])[2]) do
```

```
          if op(ExteriorDerivativeList[i])[2][j][1][NewQ] < SizeOfAlpha + 1
```

```

        then
            if op(ExteriorDerivativeList[i])[2][j][1][NewQ + 1 ] > SizeOfAlpha
                then
                    ModuloList:=[op(ModuloList), op(ExteriorDerivativeList[i])[2][j]];
                end if;
            end if;
        end do;
        EndList[i] := ModuloList;
    end do;
end if;
else
    for i by 1 to ListSize do
        ModuloList := [];
        for j by 1 to nops(op(ExteriorDerivativeList[i])[2]) do
            if op(ExteriorDerivativeList[i])[2][j][1][NewQ] > SizeOfAlpha + 1 then
                ModuloList:=[op(ModuloList), op(ExteriorDerivativeList[i])[2][j]];
            end if;
        end do;
        EndList[i] := ModuloList;
    end do;
end if;

convert(EndList, list);

end:

```

A.5.2 *SSProjectionInternalLAWC*

$$\#d_r = E_r^{p,q} - > E_r^{p+r,q-r+1}$$

#X is the Representation. 1 for Rep, 0 for just a Lie algebra

SSProjectionInternalLAWC:=proc(ExteriorDerivativeList,NewP,NewQ,SizeOfAlpha, X)

```

local ListSize, i, j,ModuloList,counter,EndList;

ListSize := nops(ExteriorDerivativeList);
EndList := Array[1.. ListSize];
if NewQ < NewQ + NewP then
  if NewQ < 1 then
    for i by 1 to ListSize do
      ModuloList := [];
      for j by 1 to nops(op(ExteriorDerivativeList[i])[2]) do
        if op(ExteriorDerivativeList[i])[2][j][1][1 + X] > SizeOfAlpha then
          ModuloList := [op(ModuloList), op(ExteriorDerivativeList[i])[2][j]];
        end if;
      end do;
      EndList[i] := ModuloList;
    end do;
  else
    for i by 1 to ListSize do
      ModuloList := [];
      for j by 1 to nops(op(ExteriorDerivativeList[i])[2]) do
        if op(ExteriorDerivativeList[i])[2][j][1][NewQ + X] < SizeOfAlpha + 1 then
          if op(ExteriorDerivativeList[i])[2][j][1][NewQ + 1 + X] > SizeOfAlpha then
            ModuloList:= [op(ModuloList), op(ExteriorDerivativeList[i])[2][j]];
          end if;
        end if;
      end do;
    end do;
  end if;
end if;
end proc;

```

```
        end if;
    end if;
end do;
    EndList[i] := ModuloList;
end do;
end if;
else
for i by 1 to ListSize do
    ModuloList := [];
    for j by 1 to nops(op(ExteriorDerivativeList[i])[2]) do
        if op(ExteriorDerivativeList[i])[2][j][1][NewQ + X] < SizeOfAlpha + 1 then
            ModuloList:= [op(ModuloList), op(ExteriorDerivativeList[i])[2][j]];
        end if;
    end do;
    EndList[i] := ModuloList;
end do;
end if;

convert(EndList, list);

end;
```

A.5.3 *CohoDecompSHSS*

```
#Serre Hoffschild Spectral Sequence Decomposition
```

```
SHSSDecomp := proc(Omega, Cohomology, CohoSize, ExactForms, ExactSize, CompSpace, CompSize)
```

```
local FTL , List, A , FTL2, List2, B, ans1, ans2, i, Soln, ans3,
```

```
EList, CompList, CohoList, MatrixList, VectorList;
```

```
CohoList := [seq( Cohomology[i],i=1 .. CohoSize)];
```

```
EList := [seq( ExactForms[i], i = 1 .. ExactSize)];
```

```
CompList := [ seq (CompSpace[i], i = 1 .. CompSize)];
```

```
A := FormToMatrix([op(CohoList),op(EList), op(CompList)]);
```

```
#List := convert(FTL, list);
```

```
#A := convert(List, Matrix);
```

```
FTL2 := FormToList([Omega]);
```

```
List2 := convert( FTL2, list);
```

```
B := convert( List2, Matrix);
```

```
Soln := LinearAlgebra:-LinearSolve(A, B);
```

```
if CohoSize = 0 then
```

```
    ans1 := [];
```

```
else
```

```
    ans1 := Soln[1.. CohoSize];
```

```
end if;
```

```
if ExactSize =0 then
```

```
    ans2 := [];
```

```
else
    ans2 := Soln[ (CohoSize + 1 ) .. CohoSize + ExactSize];
end if;

if CompSize = 0 then
    ans3 := [];
else
    ans3 := Soln[ (CohoSize + ExactSize +1) .. (ExactSize + CohoSize + CompSize)];
end if;

[ans1,ans2,ans3];

end;
```

A.5.4 *CohoDecompSHSSLAWC*

#Serre Hoffschild Spectral Sequence Decomposition

```
SHSSDecompLAWC := proc(Omega, Cohomology, CohoSize, ExactForms,
ExactSize, CompSpace, CompSize, VectorSize)
```

```
local FTL , List, A , FTL2, List2, B, ans1, ans2, i, Soln, ans3, EList,
CompList, CohoList, MatrixList, VectorList;
CohoList := [seq( Cohomology[i],i=1 .. CohoSize)];
EList := [seq( ExactForms[i], i = 1 .. ExactSize)];
CompList := [ seq (CompSpace[i], i = 1 .. CompSize)];
```

```
if CohoSize = 0 then
```

```
  if op(EList[1])[1][1] = "form" then
```

```
    A := FormToMatrix([op(CohoList),op(EList), op(CompList)]);
```

```
    FTL2 := FormToList([Omega]);
```

```
    List2 := convert( FTL2, list);
```

```
    B := convert( List2, Vector);
```

```
  else
```

```
    MatrixList := [op(CohoList),op(EList), op(CompList)];
```

```
    VectorList := [];
```

```
    for i by 1 to nops(MatrixList) do
```

```
      VectorList := [op(VectorList), TLACV(MatrixList[i], VectorSize)];
```

```
    end do;
```

```
    A := convert (VectorList, Matrix);
```

```
    B := TLACV(Omega, VectorSize);
```

```
  end if;
```

```
else
```

```

if op(CohoList[1])[1][1] = "form" then
    A := FormToMatrix([op(CohoList),op(EList), op(CompList)]);

    #List := convert(FTL, list);
    #A := convert(List, Matrix);

    FTL2 := FormToList([Omega]);
    List2 := convert( FTL2, list);
    B := convert( List2, Vector);
else
    MatrixList := [op(CohoList),op(EList), op(CompList)];
    VectorList := [];
    for i by 1 to nops(MatrixList) do
        VectorList := [op(VectorList), TLACV(MatrixList[i], VectorSize)];
    end do;
    A := convert (VectorList, Matrix);
    B := TLACV(Omega, VectorSize);
end if;
end if;

Soln := LinearAlgebra:-LinearSolve(A, B);

if CohoSize = 0 then
    ans1 := [];
else
    ans1 := Soln[1.. CohoSize];
end if;

```

```
if ExactSize =0 then
  ans2 := [];
else
  ans2 := Soln[ (CohoSize + 1 ) .. CohoSize + ExactSize];
end if;

if CompSize = 0 then
  ans3 := [];
else
  ans3 := Soln[ (CohoSize + ExactSize +1) .. (ExactSize + CohoSize + CompSize)];
end if;

[ans1,ans2,ans3];

end;
```

A.5.5 PQ

Maple Code

#PQ Function Calculates the P and Q by figuring out how many betas and alphas are in a monomial

```
PQ := proc(Sigma,SubAlgebraDimension)
```

```
    local Size, FormElementList, FormDegree,x,AlphaList,p,q;
```

```
    if op(Sigma)[1][1] = "form" then
```

```
        FormElementList := op(Sigma)[2][1][1];
```

```
        FormDegree := nops(FormElementList);
```

```
        AlphaList := select(x->is(x[SubAlgebraDimension + 1]),FormElementList);
```

```
        q:=nops(AlphaList);
```

```
        p:= FormDegree-q;
```

```
        return [p,q];
```

```
    else
```

```
        FormElementList := op(Sigma)[2][1][1];
```

```
        FormDegree := nops(FormElementList) - 1;
```

```
        FormElementList := subsop( 1 = NULL, FormElementList);
```

```
        AlphaList := select(x->is(x[SubAlgebraDimension + 1]),FormElementList);
```

```
        q:=nops(AlphaList);
```

```
        p:= FormDegree-q;
```

```
        return [p,q];
```

```
    end if;
```

```
end;
```


APPENDIX