SageMath experiments in Differential and Complex Geometry

Daniele Angella

Dipartimento di Matematica e Informatica "Ulisse Dini"
Università di Firenze

February 09, 2017

Summary

Solve research-level problems
in Differential and Complex Geometry
by SageMath.

SageMath

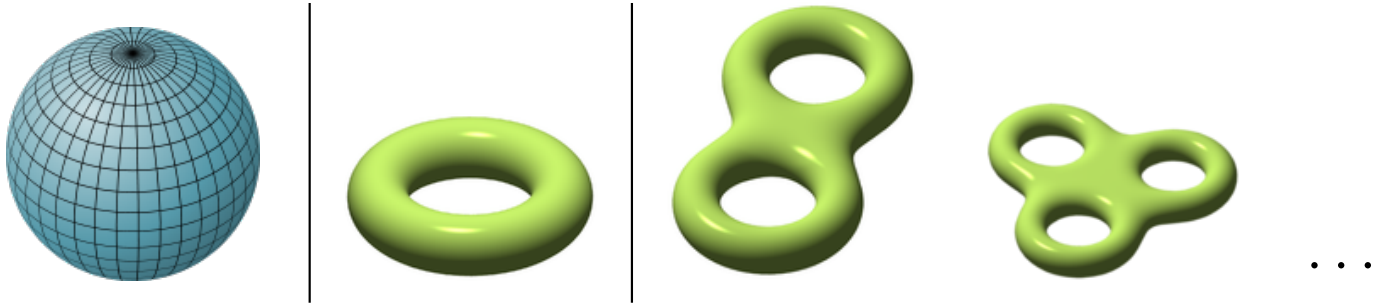
"SageMath is a free open-source mathematics software system licensed under the GPL. It builds on top of many existing open-source packages: NumPy, SciPy, matplotlib, Sympy, Maxima, GAP, FLINT, R and many more. Access their combined power through a common, Python-based language or directly via interfaces or wrappers."

SageMath, the Sage Mathematics Software System (Version 7.3), The Sage Developers, 2016, http://www.sagemath.org.

https://cloud.sagemath.com

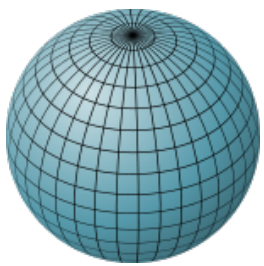
Differential Geometry

Classification
of some class of objects up to some notion of equivalence
by means of some sort of invariants.



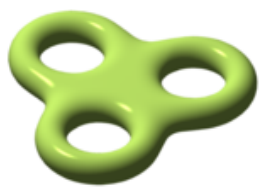
Differential Geometry

Classification
of some class of objects up to some notion of equivalence
by means of some sort of invariants.


genus $= 0$


genus $= 1$

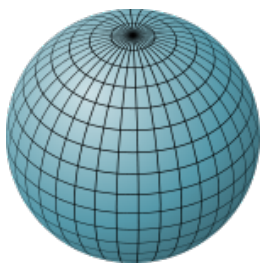

genus $= 2$


genus $= 3$

$\ldots$

Differential Geometry

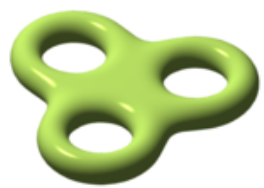
Classification
of some class of objects up to some notion of equivalence
by means of some sort of invariants.



genus $= 0$
csc positive

genus $= 1$
csc zero

genus $= 2$



genus $= 3$

csc negative

$\ldots$

Two problems in Differential Geometry

We focus on a class of objects which have enough symmetries so that their geometry is reduced linearly.


D. Angella, M. G. Franzini, F. A. Rossi, Degree of non-Kählerianity for 6-dimensional nilmanifolds, *Manuscripta Math.* 148 (2015), no. 1-2, 177–211.


A. Latorre, L. Ugarte, R. Villacampa, On the Bott-Chern cohomology and balanced Hermitian nilmanifolds, *Internat. J. Math.* 25 (2014), no. 6, 1450057, 24 pp.


D. Angella, G. Bazzoni, M. Parton, Four dimensional locally conformal symplectic Lie algebras, in preparation.

Two problems in Differential Geometry

We focus on a class of objects which have enough symmetries so that their geometry is reduced linearly.

Objects	6-dim nilmanifolds
Invariants	cohomologies

Objects	4-dim Lie algebras
Invariants	locally conformal symplectic struct

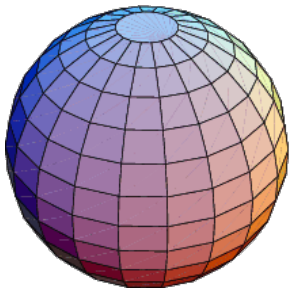

D. Angella, M. G. Franzini, F. A. Rossi, Degree of non-Kählerianity for 6-dimensional nilmanifolds, *Manuscripta Math.* 148 (2015), no. 1-2, 177–211.


A. Latorre, L. Ugarte, R. Villacampa, On the Bott-Chern cohomology and balanced Hermitian nilmanifolds, *Internat. J. Math.* 25 (2014), no. 6, 1450057, 24 pp.


D. Angella, G. Bazzoni, M. Parton, Four dimensional locally conformal symplectic Lie algebras, in preparation.

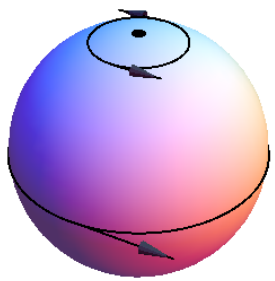
Nilmanifolds and solvmanifolds

- **manifold** = object locally modelled on $\mathbb{R}^n$, so to define a notion of smooth map



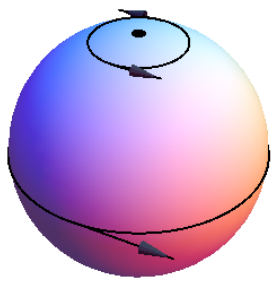
Nilmanifolds and solvmanifolds

- **manifold** = object locally modelled on $\mathbb{R}^n$, so to define a notion of smooth map
- **nil/solvmanifold** = homogeneous manifold of a connected nilpotent/solvable Lie group



Nilmanifolds and solvmanifolds

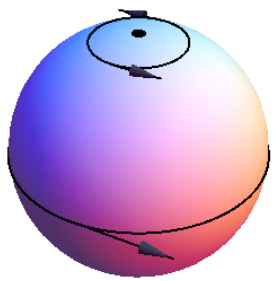
- **manifold** = object locally modelled on $\mathbb{R}^n$, so to define a notion of smooth map
- **nil/solvmanifold** = homogeneous manifold of a connected nilpotent/solvable Lie group



- **invariant objects** for the action of the Lie group give (partial) informations on the geometry

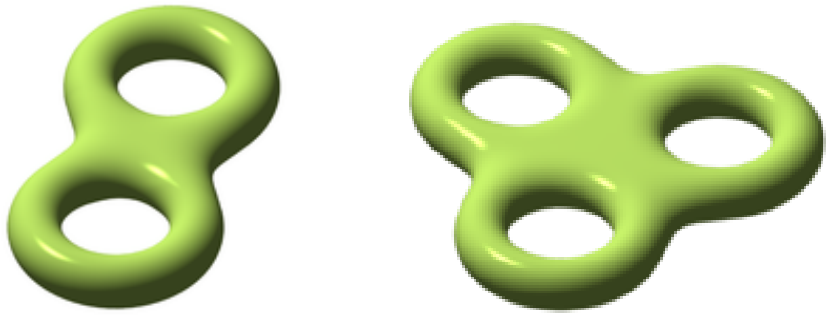
Nilmanifolds and solvmanifolds

- **manifold** = object locally modelled on $\mathbb{R}^n$, so to define a notion of smooth map
- **nil/solvmanifold** = homogeneous manifold of a connected nilpotent/solvable Lie group

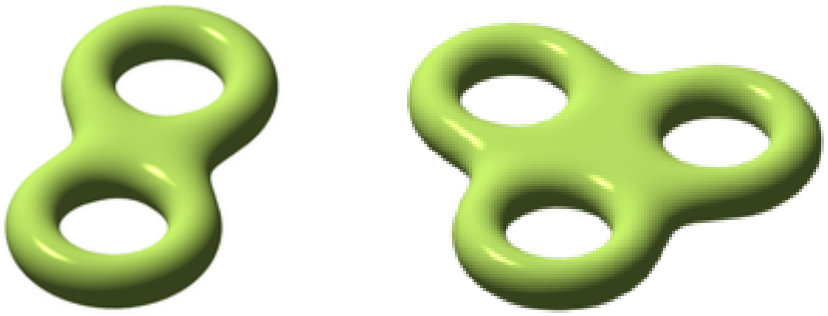


- **invariant objects** for the action of the Lie group give (partial) informations on the geometry
- they are encoded in the **Lie algebra** = vector space $\mathbb{R}^n$ with a differential $d\colon \wedge^k \mathbb{R}^n \to \wedge^{k+1}\mathbb{R}^n$ (Leibniz rule and $d \circ d = 0$)

Cohomologies

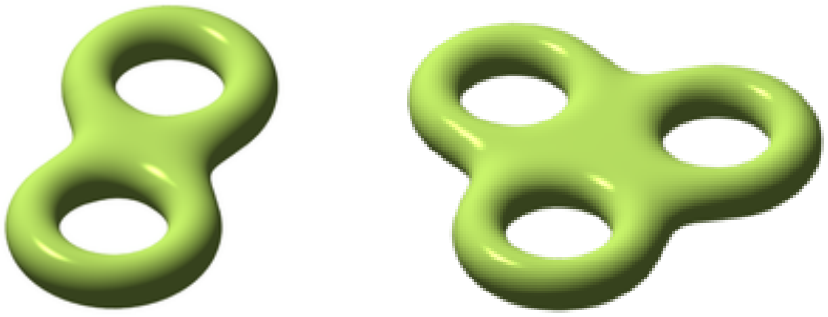


Cohomologies



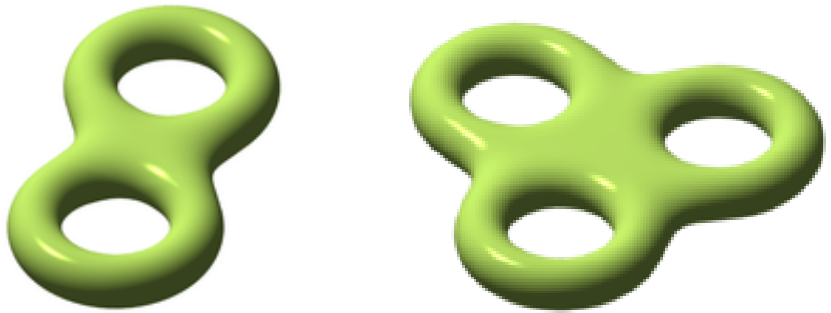
- **cohomology** = assigns a sequence of Abelian groups to a topological space

Cohomologies



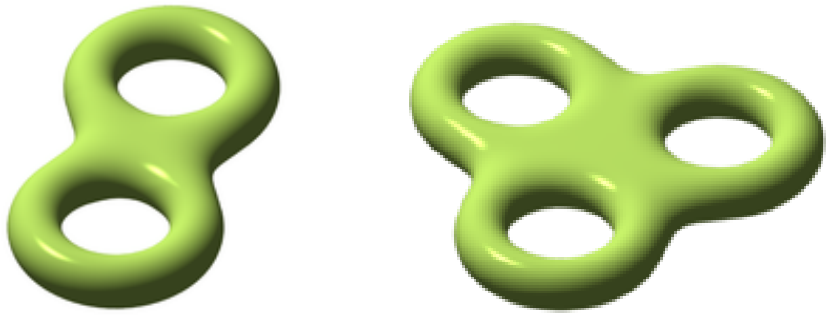
- **cohomology** = assigns a sequence of Abelian groups to a topological space
- *e.g.* **de Rham cohomology**

Cohomologies



- **cohomology** = assigns a sequence of Abelian groups to a topological space
- *e.g.* **de Rham cohomology**
- for *nil/solvmanifolds*: recall that we have a differential $d\colon \wedge^k \mathbb{R}^n \to \wedge^{k+1}\mathbb{R}^n$ with $d \circ d = 0$: **Lie algebra cohomology** $\frac{\ker d}{\operatorname{im} d}$

Cohomologies



- **cohomology** = assigns a sequence of Abelian groups to a topological space
- *e.g.* **de Rham cohomology**
- for *nil/solvmanifolds*: recall that we have a differential $d\colon \wedge^k \mathbb{R}^n \to \wedge^{k+1}\mathbb{R}^n$ with $d \circ d = 0$: **Lie algebra cohomology** $\frac{\ker d}{\operatorname{im} d}$

Thm (Nomizu)

For nilmanifolds, de Rham cohomology = Lie algebra cohomology

Cohomologies with SageMath

X nilmanifold with Lie algebra

$$\mathfrak{h}_8 := (-23, 0, 0, 0, 0, 0)$$

that is, $\mathfrak{h}_8 = \mathrm{span}\{e^0, \dots, e^5\}$ with $de^0 = -e^2 \wedge e^3$, $de^1 = \cdots = de^5 = 0$
and Leibniz rule ($d(\alpha \wedge \beta) = d\alpha \wedge \beta \pm \alpha \wedge d\beta$)

Cohomologies with SageMath

X nilmanifold with Lie algebra

$$\mathfrak{h}_8 := (-23, 0, 0, 0, 0, 0)$$

that is, $\mathfrak{h}_8 = \mathrm{span}\{e^0, \dots, e^5\}$ with $de^0 = -e^2 \wedge e^3$, $de^1 = \cdots = de^5 = 0$
and Leibniz rule ($d(\alpha \wedge \beta) = d\alpha \wedge \beta \pm \alpha \wedge d\beta$)

```
sage: E = ExteriorAlgebra(SR, 'e', 6)
```

Cohomologies with SageMath

X nilmanifold with Lie algebra

$$\mathfrak{h}_8 := (-23, 0, 0, 0, 0, 0)$$

that is, $\mathfrak{h}_8 = \mathrm{span}\{e^0, \dots, e^5\}$ with $de^0 = -e^2 \wedge e^3$, $de^1 = \cdots = de^5 = 0$
and Leibniz rule ($d(\alpha \wedge \beta) = d\alpha \wedge \beta \pm \alpha \wedge d\beta$)

```
sage: E = ExteriorAlgebra(SR, 'e', 6)

sage: str_eq = {(2,3):-E.gens()[0], }
sage: d = E.coboundary(str_eq)
```

Cohomologies with SageMath

X nilmanifold with Lie algebra

$$\mathfrak{h}_8 := (-23, 0, 0, 0, 0, 0)$$

that is, $\mathfrak{h}_8 = \mathrm{span}\{e^0, \dots, e^5\}$ with $de^0 = -e^2 \wedge e^3$, $de^1 = \cdots = de^5 = 0$
and Leibniz rule ($d(\alpha \wedge \beta) = d\alpha \wedge \beta \pm \alpha \wedge d\beta$)

```
sage: E = ExteriorAlgebra(SR, 'e', 6)

sage: str_eq = {(2,3):-E.gens()[0], }
sage: d = E.coboundary(str_eq)

sage: [d(b) for b in E.gens()]
[-e2^e3, 0, 0, 0, 0, 0]
```

Cohomologies with SageMath

X nilmanifold with Lie algebra

$$\mathfrak{h}_8 := (-23, 0, 0, 0, 0, 0)$$

that is, $\mathfrak{h}_8 = \mathrm{span}\{e^0, \dots, e^5\}$ with $de^0 = -e^2 \wedge e^3$, $de^1 = \cdots = de^5 = 0$
and Leibniz rule ($d(\alpha \wedge \beta) = d\alpha \wedge \beta \pm \alpha \wedge d\beta$)

```
sage: E = ExteriorAlgebra(SR, 'e', 6)

sage: str_eq = {(2,3):-E.gens()[0], }
sage: d = E.coboundary(str_eq)

sage: [d(b) for b in E.gens()]
[-e2^e3, 0, 0, 0, 0, 0]

sage: all([d(d(b)) == 0 for b in E.gens()])
True
```

Cohomologies with SageMath

```
sage: cplx_dR = d.chain_complex()
```

Cohomologies with SageMath

```
sage: cplx_dR = d.chain_complex()

sage: cplx_dR
Chain complex with at most 7 nonzero terms over Symbolic Ring
```

Cohomologies with SageMath

```
sage: cplx_dR = d.chain_complex()

sage: cplx_dR
Chain complex with at most 7 nonzero terms over Symbolic Ring

sage: cplx_dR.homology()
{0: Vector space of dimension 1 over Symbolic Ring,
    1: Vector space of dimension 5 over Symbolic Ring,
    2: Vector space of dimension 11 over Symbolic Ring,
    3: Vector space of dimension 14 over Symbolic Ring,
    4: Vector space of dimension 11 over Symbolic Ring,
    5: Vector space of dimension 5 over Symbolic Ring,
    6: Vector space of dimension 1 over Symbolic Ring}
```

$$b_0(X) = b_6(X) = 1, \qquad b_1(X) = b_5(X) = 5,$$
$$b_2(X) = b_4(X) = 11, \qquad b_3(X) = 14.$$

Cohomologies with SageMath

```
sage: t = cputime()
sage: E = ExteriorAlgebra(SR,'e',6)
sage: for algebra in alg_nilp_6.keys():
          d = E.coboundary(alg_nilp_6[algebra])
          H = d.chain_complex().homology()
          Poincare_poly = sum([H[j].dimension() * x^j \
              for j in range(len(H))])
          print algebra, "\n\t", Poincare_poly
sage: cputime(t)
```

Cohomologies with SageMath

```
sage: t = cputime()
sage: E = ExteriorAlgebra(SR,'e',6)
sage: for algebra in alg_nilp_6.keys():
          d = E.coboundary(alg_nilp_6[algebra])
          H = d.chain_complex().homology()
          Poincare_poly = sum([H[j].dimension() * x^j \
              for j in range(len(H))])
          print algebra, "\n\t", Poincare_poly
sage: cputime(t)

\mathfrak{g}_{6.N18}^{-1} :
        x^6 + 2*x^5 + 4*x^4 + 6*x^3 + 4*x^2 + 2*x + 1
6\mathfrak{g}_{1} :
        x^6 + 6*x^5 + 15*x^4 + 20*x^3 + 15*x^2 + 6*x + 1
\mathfrak{g}_{3.1}\oplus 3g_{1} :
        x^6 + 5*x^5 + 11*x^4 + 14*x^3 + 11*x^2 + 5*x + 1
...
0.7200000000000024
```

Cohomologies with SageMath

Introducing a complex structure:

```
sage: mat_J = matrix(6,6,[[ 0,-1, 0, 0, 0, 0], \
                          [ 1, 0, 0, 0, 0, 0], \
                          [ 0, 0, 0,-1, 0, 0], \
                          [ 0, 0, 1, 0, 0, 0], \
                          [ 0, 0, 0, 0, 0,-1], \
                          [ 0, 0, 0, 0, 1, 0]])
sage: J = E.lift_morphism(mat_J)
```

Cohomologies with SageMath

Introducing a complex structure:

```
sage: mat_J = matrix(6,6,[[ 0,-1, 0, 0, 0, 0], \
                          [ 1, 0, 0, 0, 0, 0], \
                          [ 0, 0, 0,-1, 0, 0], \
                          [ 0, 0, 1, 0, 0, 0], \
                          [ 0, 0, 0, 0, 0,-1], \
                          [ 0, 0, 0, 0, 1, 0]])
sage: J = E.lift_morphism(mat_J)

sage: for j in range(3):
          varphi[j] = E.gens()[2*j] - I * J(E.gens()[2*j])
          print varphi[j], "|-->", J(varphi[j])
e0 - I*e1 |--> I*e0 + e1
e2 - I*e3 |--> I*e2 + e3
e4 - I*e5 |--> I*e4 + e5
```

Cohomologies with SageMath

```
sage: E.<varphi0, varphi1, varphi2, \
          barvarphi0, barvarphi1, barvarphi2> = \
          ExteriorAlgebra(SR)
sage: str_eq = {(1,4): I/2 * varphi0 + I/2 * barvarphi0}
sage: delbar = E.coboundary(str_eq)
```

Cohomologies with SageMath

```
sage: E.<varphi0, varphi1, varphi2, \
          barvarphi0, barvarphi1, barvarphi2> = \
          ExteriorAlgebra(SR)
sage: str_eq = {(1,4): I/2 * varphi0 + I/2 * barvarphi0}
sage: delbar = E.coboundary(str_eq)

sage: HDol = delbar.chain_complex().homology() ; HDol
{0: Vector space of dimension 1 over Symbolic Ring,
    1: Vector space of dimension 5 over Symbolic Ring,
    2: Vector space of dimension 11 over Symbolic Ring,
    3: Vector space of dimension 14 over Symbolic Ring,
    4: Vector space of dimension 11 over Symbolic Ring,
    5: Vector space of dimension 5 over Symbolic Ring,
    6: Vector space of dimension 1 over Symbolic Ring}
```

Cohomologies with SageMath

- Ceballos-Otal-Ugarte-Villacampa classify invariant complex structures on 6-dimensional nilmanifolds in infinite families
- other cohomologies: Bott-Chern and Aeppli

```
sage: hDol = {}
sage: for j in range(len(E.gens())+1):
          V = VectorSpace(SR,len(E.basis(j)))
          Z = V.subspace(d_mat[j].transpose().kernel())
          B = Z.subspace([V([d_mat[j-1][h,k] \
              for h in range(d_mat[j-1].nrows())]) \
              for k in range(d_mat[j-1].ncols())])
          H = Z.quotient(B)
          hDol[j] = dim(H)
sage: hDol
{0: 1, 1: 3, 2: 5, 3: 6, 4: 5, 5: 3, 6: 1}
```

lcs structures

- **symplectic** = non-degenerate 2-form ω such that

$$d\omega = 0$$

- **locally conformally symplectic (lcs)** = locally, up to conformal change, of the type above:

$$d\omega = \vartheta \wedge \omega \quad \text{with} \quad d\vartheta = 0$$

lcs structures

classify lcs structure on 4-dimensional Lie algebra

$$\mathfrak{r}_4 = (03 + 13, 13 + 23, 23, 0)$$

lcs structures

classify lcs structure on 4-dimensional Lie algebra

$$\mathfrak{r}_4 = (03 + 13, 13 + 23, 23, 0)$$

```
sage: E = ExteriorAlgebra(SR, 'e', 4)
sage: str_eq = {
          (0,3) : E.gens()[0],
          (1,3) : E.gens()[0] + E.gens()[1],
          (2,3) : E.gens()[1] + E.gens()[2],
      }
sage: d = E.coboundary(str_eq)
```

lcs structures

classify lcs structure on 4-dimensional Lie algebra

$$\mathfrak{r}_4 = (03 + 13, 13 + 23, 23, 0)$$

```
sage: E = ExteriorAlgebra(SR, 'e', 4)
sage: str_eq = {
          (0,3) : E.gens()[0],
          (1,3) : E.gens()[0] + E.gens()[1],
          (2,3) : E.gens()[1] + E.gens()[2],
      }
sage: d = E.coboundary(str_eq)

sage: print([d(b) for b in E.gens()])
[e0^e3 + e1^e3, e1^e3 + e2^e3, e2^e3, 0]
```

lcs structures

$$d\vartheta = 0 \qquad \text{and} \qquad d\omega = \vartheta \wedge \omega$$

```
sage: thetacoeff = var(["theta%d" % j \
          for j in range(len(E.basis(1)))])
sage: theta = sum([thetacoeff[j] * E.gens()[j] \
          for j in range(len(E.basis(1)))])
```

lcs structures

$$d\vartheta = 0 \qquad \text{and} \qquad d\omega = \vartheta \wedge \omega$$

```
sage: thetacoeff = var(["theta%d" % j \
          for j in range(len(E.basis(1)))])
sage: theta = sum([thetacoeff[j] * E.gens()[j] \
          for j in range(len(E.basis(1)))])

sage: d(theta)
```

$$\vartheta_0 e^0 \wedge e^3 + (\vartheta_0 + \vartheta_1) e^1 \wedge e^3 + (\vartheta_1 + \vartheta_2) e^2 \wedge e^3$$

lcs structures

$$d\vartheta = 0 \qquad \text{and} \qquad d\omega = \vartheta \wedge \omega$$

```
sage: theta = theta3 * e3
sage: Omegacoeff = var(["omega%d%d" % (i,j) \
          for i in range(len(E.basis(1))) \
          for j in range(i+1,len(E.basis(1)))])
sage: Omega = sum([Omegacoeff[j] * list(E.basis(2))[j] \
          for j in range(len(E.basis(2)))])
```

lcs structures

$$d\vartheta = 0 \qquad \text{and} \qquad d\omega = \vartheta \wedge \omega$$

```
sage: theta = theta3 * e3
sage: Omegacoeff = var(["omega%d%d" % (i,j) \
          for i in range(len(E.basis(1))) \
          for j in range(i+1,len(E.basis(1)))])
sage: Omega = sum([Omegacoeff[j] * list(E.basis(2))[j] \
          for j in range(len(E.basis(2)))])

sage: d(Omega) - theta * Omega
```

$$(-\omega_{01}\vartheta_3 - 2\,\omega_{01})\, e_0\wedge e_1\wedge e_3 + (-\omega_{02}\vartheta_3 - \omega_{01} - 2\,\omega_{02})\, e_0\wedge e_2\wedge e_3$$
$$+ (-\omega_{12}\vartheta_3 - \omega_{02} - 2\,\omega_{12})\, e_1\wedge e_2\wedge e_3$$

lcs structures

- if $\vartheta_3 \neq -2$, then $\omega_{01} = \omega_{02} = \omega_{12} = 0$;
  then $\Omega^2 = 0$, so Ω is degenerate
- if $\vartheta_3 = -2$, then we get the lcs structures

$$\begin{aligned} \vartheta &= -2 \cdot e^3 \\ \Omega &= \omega_{03} \cdot e^0 \wedge e^3 + \omega_{12} \cdot e^1 \wedge e^2 + \omega_{13} \cdot e^1 \wedge e^3 + \omega_{23} \cdot e^2 \wedge e^3 \end{aligned}$$

where $\omega_{23}, \omega_{13}, \omega_{12}, \omega_{03} \in \mathbb{R}$ satisfy $\omega_{12} \cdot \omega_{03} \neq 0$

lcs structures

Are they equivalent?

lcs structures

Are they equivalent?

Equivalences ψ of the Lie algebra are induced by invertible matrices

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix}$$

such that

$$\psi(d(\_)) = d(\psi(\_))$$

lcs structures

$$\begin{pmatrix} \frac{1}{\omega_{03}} & 0 & 0 & 0 \\ 0 & \frac{1}{\omega_{03}} & 0 & 0 \\ 0 & 0 & \frac{1}{\omega_{03}} & 0 \\ 0 & \frac{\omega_{01}}{\omega_{02}} & -\frac{\omega_{12}}{\omega_{02}} & 1 \end{pmatrix}$$

```
sage: mat1 = matrix(4, 4, [ ... ])
sage: morphism1 = E.lift_morphism(mat1)
```

lcs structures

$$\begin{pmatrix} \frac{1}{\omega_{03}} & 0 & 0 & 0 \\ 0 & \frac{1}{\omega_{03}} & 0 & 0 \\ 0 & 0 & \frac{1}{\omega_{03}} & 0 \\ 0 & \frac{\omega_{01}}{\omega_{02}} & -\frac{\omega_{12}}{\omega_{02}} & 1 \end{pmatrix}$$

```
sage: mat1 = matrix(4, 4, [ ... ])
sage: morphism1 = E.lift_morphism(mat1)

sage: all([morphism1(d(b)) == d(morphism1(b)) \
           for b in E.gens()])
True
```

lcs structures

$$\begin{pmatrix} \frac{1}{\omega_{03}} & 0 & 0 & 0 \\ 0 & \frac{1}{\omega_{03}} & 0 & 0 \\ 0 & 0 & \frac{1}{\omega_{03}} & 0 \\ 0 & \frac{\omega_{01}}{\omega_{02}} & -\frac{\omega_{12}}{\omega_{02}} & 1 \end{pmatrix}$$

```
sage: mat1 = matrix(4, 4, [ ... ])
sage: morphism1 = E.lift_morphism(mat1)

sage: all([morphism1(d(b)) == d(morphism1(b)) \
           for b in E.gens()])
True

sage: mat1.determinant()
omega03^(-3)
```

lcs structures

```
sage: theta = morphism1(theta)
sage: Omega = morphism1(Omega)
```

lcs structures

```
sage: theta = morphism1(theta)
sage: Omega = morphism1(Omega)

sage: theta ; Omega
-2*e3
e0^e3 + r4/r5^2*e1^e2
```

lcs structures

```
sage: theta = morphism1(theta)
sage: Omega = morphism1(Omega)

sage: theta ; Omega
-2*e3
e0^e3 + r4/r5^2*e1^e2
```

$$\begin{cases} \vartheta &= -2 \cdot e^3 \\ \Omega &= e^0 \wedge e^3 + \sigma \cdot e^1 \wedge e^3, \quad \text{for } \sigma \neq 0 \end{cases}$$

lcs structures

Are they different?

lcs structures

Are they different?

```
sage: theta = -2 * E.gens()[3]
sage: var_Omega = var("sigma1 sigma2")
sage: Omega1 = ... + sigma1 * E.gens()[1] * E.gens()[2]
sage: Omega2 = ... + sigma2 * E.gens()[1] * E.gens()[2]
sage: var_mat = var(["a%d%d" % (i,j) for ...])
sage: mat = matrix(len(E.gens()), len(E.gens()), list(var_mat))
sage: morphism = E.lift_morphism(mat)
```

lcs structures

Are they different?

```
sage: theta = -2 * E.gens()[3]
sage: var_Omega = var("sigma1 sigma2")
sage: Omega1 = ... + sigma1 * E.gens()[1] * E.gens()[2]
sage: Omega2 = ... + sigma2 * E.gens()[1] * E.gens()[2]
sage: var_mat = var(["a%d%d" % (i,j) for ...])
sage: mat = matrix(len(E.gens()), len(E.gens()), list(var_mat))
sage: morphism = E.lift_morphism(mat)

sage: Anello = QQ[var_mat + var_Omega]
sage: ideal0 =[(d(morphism(b)) - morphism(d(b))).interior_product(c) \
          for c in E.basis(2) for b in E.basis(1)]
sage: ideal1 = [(morphism(theta) - theta).interior_product(c) \
          for c in E.basis(1)]
sage: ideal1 += [(morphism(Omega1) - Omega2).interior_product(c) \
          for c in E.basis(2)]
```

lcs structures

Are they different?

```
sage: theta = -2 * E.gens()[3]
sage: var_Omega = var("sigma1 sigma2")
sage: Omega1 = ... + sigma1 * E.gens()[1] * E.gens()[2]
sage: Omega2 = ... + sigma2 * E.gens()[1] * E.gens()[2]
sage: var_mat = var(["a%d%d" % (i,j) for ...])
sage: mat = matrix(len(E.gens()), len(E.gens()), list(var_mat))
sage: morphism = E.lift_morphism(mat)

sage: Anello = QQ[var_mat + var_Omega]
sage: ideal0 =[(d(morphism(b)) - morphism(d(b))).interior_product(c) \
          for c in E.basis(2) for b in E.basis(1)]
sage: ideal1 = [(morphism(theta) - theta).interior_product(c) \
          for c in E.basis(1)]
sage: ideal1 += [(morphism(Omega1) - Omega2).interior_product(c) \
          for c in E.basis(2)]

sage: B = Anello.ideal(ideal0 + ideal1).groebner_basis() ; B
[a21^2 + a31*sigma2 - a20, ..., sigma1 - sigma2]
```

Thanks for the attention

GeCo GeDi project:

`http://gecogedi.dimai.unifi.it`