Abelian categories and linear partial differential equations

Mohamed Barakat

University of Kaiserslautern

Workshop on Computational Commutative Algebra
July 2011, Tehran

Overview

Linear systems of PDEs

Consider the CAUCHY-RIEMANN differential equation:

$$u_x - v_y = 0, \quad u_y + v_x = 0 \tag{CR}$$

Q:

How can we deal with this differential equation algebraically?

Linear systems of PDEs

Consider the CAUCHY-RIEMANN differential equation:

$$u_x - v_y = 0, \quad u_y + v_x = 0 \tag{CR}$$

Q:

How can we deal with this differential equation algebraically?

Computing *modulo* the equations

Idea: Consider differential consequences

- $u_{xx} + u_{yy} = ?$
- $v_{xx} + v_{yy} = ?$
- $u_{xxx} + u_y + v_x + v_{yyy} = ?$
- $u_{xxx} + u_{yy} + v_{xx} + v_{yyy} = ?$
- ...

Q:

How can we consider all differential consequences simultaneously?

Computing *modulo* the equations

Idea: Consider differential consequences

- $u_{xx} + u_{yy} = \partial_x(u_x - v_y) + \partial_y(u_y + v_x) = 0$
- $v_{xx} + v_{yy} = -\partial_y(u_x - v_y) + \partial_x(u_y + v_x) = 0$
- $u_{xxx} + u_y + v_x + v_{yyy} = ?$
- $u_{xxx} + u_{yy} + v_{xx} + v_{yyy} = ?$
- ...

Q:

How can we consider all differential consequences simultaneously?

Computing *modulo* the equations

Idea: Consider differential consequences

- $u_{xx} + u_{yy} = 0$
- $v_{xx} + v_{yy} = 0$
- $u_{xxx} + u_y + v_x + v_{yyy} = ?$
- $u_{xxx} + u_{yy} + v_{xx} + v_{yyy} = ?$
- ...

Q:

How can we consider all differential consequences simultaneously?

Computing *modulo* the equations

Idea: Consider differential consequences

- $u_{xx} + u_{yy} = 0$
- $v_{xx} + v_{yy} = 0$
- $u_{xxx} + u_y + v_x + v_{yyy} = (\partial_x^2 - \partial_y^2)(u_x - v_y) + (\partial_x \partial_y + 1)(u_y + v_x) = 0$
- $u_{xxx} + u_{yy} + v_{xx} + v_{yyy} = ?$
- ...

Q:

How can we consider all differential consequences simultaneously?

Computing *modulo* the equations

Idea: Consider differential consequences

- $u_{xx} + u_{yy} = 0$
- $v_{xx} + v_{yy} = 0$
- $u_{xxx} + u_y + v_x + v_{yyy} = 0$
- $u_{xxx} + u_{yy} + v_{xx} + v_{yyy} = ?$
- ...

Q:

How can we consider all differential consequences simultaneously?

Computing *modulo* the equations

Idea: Consider differential consequences

- $u_{xx} + u_{yy} = 0$
- $v_{xx} + v_{yy} = 0$
- $u_{xxx} + u_y + v_x + v_{yyy} = 0$
- $u_{xxx} + u_{yy} + v_{xx} + v_{yyy} = u_{yy} - v_{yy}$
- ...

Q:

How can we consider all differential consequences simultaneously?

Computing *modulo* the equations

Idea: Consider differential consequences

- $u_{xx} + u_{yy} = 0$
- $v_{xx} + v_{yy} = 0$
- $u_{xxx} + u_y + v_x + v_{yyy} = 0$
- $u_{xxx} + u_{yy} + v_{xx} + v_{yyy} = u_{yy} - v_{yy}$
- ...

Q:

How can we consider all differential consequences simultaneously?

First step: From systems to differential operators

The CAUCHY-RIEMANN differential equation can be rewritten as

$$\underbrace{\begin{pmatrix} \partial_x & -\partial_y \\ \partial_y & \partial_x \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{\psi} = 0,$$

- with $A \in D^{p \times q}$ $\quad (p = q = 2)$
- for $D = \mathbb{R}[\partial_x, \partial_y]$, or more general diff. rings $R[\partial_x, \partial_y]$
- and $\psi \in \mathcal{F}^{2 \times 1}$, for $\mathcal{F} = C^\infty(\mathbb{R}^n)$ $\quad (n = 2)$
- The solution space:

$$\mathrm{Sol}_{\mathcal{F}}(A) := \{\psi \in \mathcal{F}^{q \times 1} \mid A\psi = 0\}$$

First step: From systems to differential operators

The CAUCHY-RIEMANN differential equation can be rewritten as

$$\underbrace{\begin{pmatrix} \partial_x & -\partial_y \\ \partial_y & \partial_x \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{\psi} = 0,$$

- with $A \in D^{p \times q}$ $\quad (p = q = 2)$
- for $D = \mathbb{R}[\partial_x, \partial_y]$, or more general diff. rings $R[\partial_x, \partial_y]$
- and $\psi \in \mathcal{F}^{2 \times 1}$, for $\mathcal{F} = C^\infty(\mathbb{R}^n) \quad (n = 2)$
- The solution space:

$$\mathrm{Sol}_{\mathcal{F}}(A) := \{\psi \in \mathcal{F}^{q \times 1} \mid A\psi = 0\}$$

First step: From systems to differential operators

The CAUCHY-RIEMANN differential equation can be rewritten as

$$\underbrace{\begin{pmatrix} \partial_x & -\partial_y \\ \partial_y & \partial_x \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{\psi} = 0,$$

- with $A \in D^{p \times q}$ $\quad (p = q = 2)$
- for $D = \mathbb{R}[\partial_x, \partial_y]$, or more general diff. rings $R[\partial_x, \partial_y]$
- and $\psi \in \mathcal{F}^{2\times 1}$, for $\mathcal{F} = C^\infty(\mathbb{R}^n)$ $\quad (n = 2)$
- The solution space:

$$\mathrm{Sol}_{\mathcal{F}}(A) := \{\psi \in \mathcal{F}^{q\times 1} \mid A\psi = 0\}$$

First step: From systems to differential operators

The CAUCHY-RIEMANN differential equation can be rewritten as

$$\underbrace{\begin{pmatrix} \partial_x & -\partial_y \\ \partial_y & \partial_x \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{\psi} = 0,$$

- with $A \in D^{p \times q} \qquad (p = q = 2)$
- for $D = \mathbb{R}[\partial_x, \partial_y]$, or more general diff. rings $R[\partial_x, \partial_y]$
- and $\psi \in \mathcal{F}^{2 \times 1}$, for $\mathcal{F} = C^\infty(\mathbb{R}^n) \qquad (n = 2)$
- The solution space:

$$\mathrm{Sol}_{\mathcal{F}}(A) := \{\psi \in \mathcal{F}^{q \times 1} \mid A\psi = 0\}$$

First step: From systems to differential operators

The CAUCHY-RIEMANN differential equation can be rewritten as

$$\underbrace{\begin{pmatrix} \partial_x & -\partial_y \\ \partial_y & \partial_x \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{\psi} = 0,$$

- with $A \in D^{p \times q} \quad (p = q = 2)$
- for $D = \mathbb{R}[\partial_x, \partial_y]$, or more general diff. rings $R[\partial_x, \partial_y]$
- and $\psi \in \mathcal{F}^{2 \times 1}$, for $\mathcal{F} = C^\infty(\mathbb{R}^n) \quad (n = 2)$
- The solution space:

$$\mathrm{Sol}_{\mathcal{F}}(A) := \{\psi \in \mathcal{F}^{q \times 1} \mid A\psi = 0\}$$

First step: From systems to differential operators

The Cauchy-Riemann differential equation can be rewritten as

$$\underbrace{\begin{pmatrix} \partial_x & -\partial_y \\ \partial_y & \partial_x \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{\psi} = 0,$$

- with $A \in D^{p \times q} \quad (p = q = 2)$
- for $D = \mathbb{R}[\partial_x, \partial_y]$, or more general diff. rings $R[\partial_x, \partial_y]$
- and $\psi \in \mathcal{F}^{2 \times 1}$, for $\mathcal{F} = C^\infty(\mathbb{R}^n) \quad (n = 2)$
- The solution space:

$$\mathrm{Sol}_{\mathcal{F}}(A) := \{\psi \in \mathcal{F}^{q \times 1} \mid A\psi = 0\}$$

Second step: From differential operators to modules

- $u_{xx} + u_{yy}$
-
-

Second step: From differential operators to modules

- $\partial_x(u_x - v_y) + \partial_y(u_y + v_x)$
-
-

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} \begin{pmatrix} u_x - v_y \\ u_y + v_x \end{pmatrix}$

Second step: From differential operators to modules

- $$\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} \begin{pmatrix} \partial_x & -\partial_y \\ \partial_y & \partial_x \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$
- $v_{xx} + v_{yy}$
-

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$
- $-\partial_y(u_x - v_y) + \partial_x(u_y + v_x)$
-

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$
- $\begin{pmatrix} -\partial_y & \partial_x \end{pmatrix} \begin{pmatrix} \partial_x & -\partial_y \\ \partial_y & \partial_x \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$
-

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$
- $\begin{pmatrix} -\partial_y & \partial_x \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$
- $\begin{pmatrix} -\partial_y & \partial_x \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$
- $u_{xxx} + u_y + v_x + v_{yyy}$

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$
- $\begin{pmatrix} -\partial_y & \partial_x \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$
- $(\partial_x^2 - \partial_y^2)(u_x - v_y) + (\partial_x \partial_y + 1)(u_y + v_x)$

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$
- $\begin{pmatrix} -\partial_y & \partial_x \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$
- $\begin{pmatrix} \partial_x^2 - \partial_y^2 & \partial_x \partial_y + 1 \end{pmatrix} A \begin{pmatrix} u \\ v \end{pmatrix}$

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A$
- $\begin{pmatrix} -\partial_y & \partial_x \end{pmatrix} A$
- $\begin{pmatrix} \partial_x^2 - \partial_y^2 & \partial_x \partial_y + 1 \end{pmatrix} A$

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A \in D^{1\times p} A$
- $\begin{pmatrix} -\partial_y & \partial_x \end{pmatrix} A \in D^{1\times p} A$
- $\begin{pmatrix} \partial_x^2 - \partial_y^2 & \partial_x\partial_y + 1 \end{pmatrix} A \in D^{1\times p} A$

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A \in D^{1\times p} A \leq D^{1\times q}$
- $\begin{pmatrix} -\partial_y & \partial_x \end{pmatrix} A \in D^{1\times p} A \leq D^{1\times q}$
- $\begin{pmatrix} \partial_x^2 - \partial_y^2 & \partial_x\partial_y + 1 \end{pmatrix} A \in D^{1\times p} A \leq D^{1\times q}$

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A \in D^{1\times p} A \leq D^{1\times q}$
- $\begin{pmatrix} -\partial_y & \partial_x \end{pmatrix} A \in D^{1\times p} A \leq D^{1\times q}$
- $\begin{pmatrix} \partial_x^2 - \partial_y^2 & \partial_x\partial_y + 1 \end{pmatrix} A \in D^{1\times p} A \leq D^{1\times q}$

$$\{\text{differential consequences}\} \equiv D^{1\times p} A \leq D^{1\times q}.$$

Second step: From differential operators to modules

- $\begin{pmatrix} \partial_x & \partial_y \end{pmatrix} A \in D^{1\times p} A \leq D^{1\times q}$

- $\begin{pmatrix} -\partial_y & \partial_x \end{pmatrix} A \in D^{1\times p} A \leq D^{1\times q}$

- $\begin{pmatrix} \partial_x^2 - \partial_y^2 & \partial_x \partial_y + 1 \end{pmatrix} A \in D^{1\times p} A \leq D^{1\times q}$

{differential consequences} $\equiv D^{1\times p} A \leq D^{1\times q}$.

computing modulo the equations

$\triangleq$

computing in the left D-module $\boxed{M := D^{1\times q} / D^{1\times p} A}$.

The module M

$M = D^{1\times q}/D^{1\times p}A$ is thus **finitely presented**:

- **Generators**: The q residue classes of standard generators of the free D-module $F_0 = D^{1\times q} = \langle e_1, \ldots, e_q \rangle$.
- **Relations**: The p rows of A.
- The matrix diff. operator A is now the **matrix of relations** of M.

The module M

$M = D^{1\times q}/D^{1\times p}A$ is thus **finitely presented**:

- **Generators**: The q residue classes of standard generators of the free D-module $F_0 = D^{1\times q} = \langle e_1, \ldots, e_q\rangle$.
- **Relations**: The p rows of A.
- The matrix diff. operator A is now the **matrix of relations** of M.

The module M

$M = D^{1\times q}/D^{1\times p}A$ is thus **finitely presented**:

- **Generators**: The q residue classes of standard generators of the free D-module $F_0 = D^{1\times q} = \langle e_1, \ldots, e_q \rangle$.
- **Relations**: The p rows of A.
- The matrix diff. operator A is now the **matrix of relations** of M.

The module M

$M = D^{1\times q}/D^{1\times p}A$ is thus **finitely presented**:

- **Generators**: The q residue classes of standard generators of the free D-module $F_0 = D^{1\times q} = \langle e_1, \dots, e_q \rangle$.
- **Relations**: The p rows of A.
- The matrix diff. operator A is now the **matrix of relations** of M.

Linear systems as modules

Q: How to describe the space of solutions in this language?

Theorem (B. MALGRANGE & E. NOETHER)
Let $\mathcal{F}$ be the D-module where we seek the solutions.

$$\begin{array}{ccc} \mathrm{Hom}_D(M,\mathcal{F}) & \xrightarrow{\cong} & \mathrm{Sol}_{\mathcal{F}}(A) \\ \psi & \mapsto & \psi = (\psi(e_i)) \in \mathcal{F}^{q\times 1} \end{array}$$

is an isomorphism of ABELian groups.

Linear systems as modules

Q: How to describe the space of solutions in this language?

Theorem (B. MALGRANGE & E. NOETHER)
Let $\mathfrak{F}$ be the D-module where we seek the solutions.

$$\begin{array}{ccc} \mathrm{Hom}_D(M,\mathfrak{F}) & \stackrel{\cong}{\longrightarrow} & \mathrm{Sol}_{\mathfrak{F}}(A) \\ \psi & \mapsto & \psi = (\psi(e_i)) \in \mathfrak{F}^{q\times 1} \end{array}$$

is an isomorphism of ABELian groups.

Linear systems as modules

Q: How to describe the space of solutions in this language?

Theorem (B. MALGRANGE & E. NOETHER)
Let $\mathfrak{F}$ be the D-module where we seek the solutions.

$$\begin{array}{ccc} \mathrm{Hom}_D(M,\mathfrak{F}) & \xrightarrow{\cong} & \mathrm{Sol}_{\mathfrak{F}}(A) \\ \psi & \mapsto & \psi = (\psi(e_i)) \in \mathfrak{F}^{q\times 1} \end{array}$$

is an isomorphism of ABELian groups.

Linear systems as modules

Q: How to describe the space of solutions in this language?

Theorem (B. MALGRANGE & E. NOETHER)
Let $\mathcal{F}$ be the D-module where we seek the solutions.

$$\begin{array}{ccc} \mathrm{Hom}_D(M, \mathcal{F}) & \xrightarrow{\cong} & \mathrm{Sol}_{\mathcal{F}}(A) \\ \psi & \mapsto & \psi = (\psi(e_i)) \in \mathcal{F}^{q \times 1} \end{array}$$

is an isomorphism of ABELian groups.

Linear systems vs. modules

Q: But where is the advantage?

A:

- $\mathrm{Hom}_D(M, \mathcal{F})$ only depends on the *isomorphism type* of M.
- The D-module $\mathcal{F}$ can be altered.

Linear systems vs. modules

Q: But where is the advantage?

A:

- $\mathrm{Hom}_D(M, \mathcal{F})$ only depends on the *isomorphism type* of M.
- The D-module $\mathcal{F}$ can be altered.

Linear systems vs. modules

Q: But where is the advantage?

A:

- $\mathrm{Hom}_D(M, \mathcal{F})$ only depends on the *isomorphism type* of M.
- The D-module $\mathcal{F}$ can be altered.

The module M

Interpret the differential operator A as a map

$$A : D^{1\times p} \to D^{1\times q}.$$

Recover the D-module M as the **cokernel** of A

$$M := D^{1\times q}/D^{1\times p}A = \operatorname{coker}(F_1 \xrightarrow{A} F_0),$$

where $F_1 := D^{1\times p}$ and $F_0 := D^{1\times q}$.

The module M

Interpret the differential operator A as a map

$$A : D^{1\times p} \to D^{1\times q}.$$

Recover the D-module M as the **cokernel** of A

$$M := D^{1\times q}/D^{1\times p}A = \mathrm{coker}(F_1 \xrightarrow{A} F_0),$$

where $F_1 := D^{1\times p}$ and $F_0 := D^{1\times q}$.

Linear systems vs. modules

In analysis we learn that the higher order scalar ODE (in u)

$$u_{xxx} + a(x)u_{xx} + b(x)u_x + c(x)u = 0$$

and the first order system (in u, v, w)

$$u_x = v,\ \ v_x = w, w_x + a(x)w + b(x)v + c(x)u = 0$$

are "equivalent".
So consider the differential algebra $D = K(x)[\partial]$ and the corresponding differential operators

$$\mathrm{M}_{\mathrm{scl}} := \left(\ \partial^3 + a(x)\partial^2 + b(x)\partial + c(x)\ \right) \in D^{1\times 1}$$

and

$$\mathrm{M}_{\mathrm{sys}} := \begin{pmatrix} \partial & -1 & 0 \\ 0 & \partial & -1 \\ c(x) & b(x) & \partial + a(x) \end{pmatrix} \in D^{3\times 3}.$$

Linear systems vs. modules

In analysis we learn that the higher order scalar ODE (in u)

$$u_{xxx} + a(x)u_{xx} + b(x)u_x + c(x)u = 0$$

and the first order system (in u, v, w)

$$u_x = v,\ v_x = w, w_x + a(x)w + b(x)v + c(x)u = 0$$

are “equivalent”.

So consider the differential algebra $D = K(x)[\partial]$ and the corresponding differential operators

$$\mathrm{M}_{\mathrm{scl}} := \left(\partial^3 + a(x)\partial^2 + b(x)\partial + c(x) \right) \in D^{1\times 1}$$

and

$$\mathrm{M}_{\mathrm{sys}} := \begin{pmatrix} \partial & -1 & 0 \\ 0 & \partial & -1 \\ c(x) & b(x) & \partial + a(x) \end{pmatrix} \in D^{3\times 3}.$$

Linear systems vs. modules

In analysis we learn that the higher order scalar ODE (in u)

$$u_{xxx} + a(x)u_{xx} + b(x)u_x + c(x)u = 0$$

and the first order system (in u, v, w)

$$u_x = v,\ v_x = w, w_x + a(x)w + b(x)v + c(x)u = 0$$

are “equivalent”.
So consider the differential algebra $D = K(x)[\partial]$ and the corresponding differential operators

$$\mathrm{M}_{\mathrm{scl}} := \left(\ \partial^3 + a(x)\partial^2 + b(x)\partial + c(x)\ \right) \in D^{1\times 1}$$

and

$$\mathrm{M}_{\mathrm{sys}} := \begin{pmatrix} \partial & -1 & 0 \\ 0 & \partial & -1 \\ c(x) & b(x) & \partial + a(x) \end{pmatrix} \in D^{3\times 3}.$$

Linear systems vs. modules

In analysis we learn that the higher order scalar ODE (in u)

$$u_{xxx} + a(x)u_{xx} + b(x)u_x + c(x)u = 0$$

and the first order system (in u, v, w)

$$u_x = v,\; v_x = w, w_x + a(x)w + b(x)v + c(x)u = 0$$

are “equivalent”.
So consider the differential algebra $D = K(x)[\partial]$ and the corresponding differential operators

$$\mathtt{M}_{\mathrm{scl}} := \begin{pmatrix} \partial^3 + a(x)\partial^2 + b(x)\partial + c(x) \end{pmatrix} \in D^{1\times 1}$$

and

$$\mathtt{M}_{\mathrm{sys}} := \begin{pmatrix} \partial & -1 & 0 \\ 0 & \partial & -1 \\ c(x) & b(x) & \partial + a(x) \end{pmatrix} \in D^{3\times 3}.$$

Linear systems vs. modules

In analysis we learn that the higher order scalar ODE (in u)

$$u_{xxx} + a(x)u_{xx} + b(x)u_x + c(x)u = 0$$

and the first order system (in u, v, w)

$$u_x = v,\ v_x = w, w_x + a(x)w + b(x)v + c(x)u = 0$$

are “equivalent”.
So consider the differential algebra $D = K(x)[\partial]$ and the corresponding differential operators

$$\mathtt{M}_{\mathrm{scl}} := \left(\ \partial^3 + a(x)\partial^2 + b(x)\partial + c(x)\ \right) \in D^{1\times 1}$$

and

$$\mathtt{M}_{\mathrm{sys}} := \begin{pmatrix} \partial & -1 & 0 \\ 0 & \partial & -1 \\ c(x) & b(x) & \partial + a(x) \end{pmatrix} \in D^{3\times 3}.$$

Linear systems vs. modules

The corresponding modules are

$$M_{\mathrm{scl}} := \mathrm{coker}(D^{1\times 1} \xrightarrow{\mathtt{M}_{\mathrm{scl}}} D^{1\times 1}) \text{ and}$$
$$M_{\mathrm{sys}} := \mathrm{coker}(D^{1\times 3} \xrightarrow{\mathtt{M}_{\mathrm{sys}}} D^{1\times 3}).$$

The modules M_{scl} and M_{sys} are indeed isomorphic:

$$\alpha : M_{\mathrm{scl}} \xrightarrow{\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}} M_{\mathrm{sys}}, \quad \beta : M_{\mathrm{sys}} \xrightarrow{\begin{pmatrix} 1 \\ \partial \\ \partial^2 \end{pmatrix}} M_{\mathrm{scl}}.$$

We call $M_{\mathrm{scl}} \cong M_{\mathrm{sys}}$ the module of **universal solutions.**

Linear systems vs. modules

The corresponding modules are

$$\begin{aligned} M_{\mathrm{scl}} &:= \operatorname{coker}(D^{1\times 1} \xrightarrow{\mathtt{M}_{\mathrm{scl}}} D^{1\times 1}) \text{ and} \\ M_{\mathrm{sys}} &:= \operatorname{coker}(D^{1\times 3} \xrightarrow{\mathtt{M}_{\mathrm{sys}}} D^{1\times 3}). \end{aligned}$$

The modules M_{scl} and M_{sys} are indeed isomorphic:

$$\alpha : M_{\mathrm{scl}} \xrightarrow{\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}} M_{\mathrm{sys}}, \quad \beta : M_{\mathrm{sys}} \xrightarrow{\begin{pmatrix} 1 \\ \partial \\ \partial^2 \end{pmatrix}} M_{\mathrm{scl}}.$$

We call $M_{\mathrm{scl}} \cong M_{\mathrm{sys}}$ the module of **universal solutions.**

Linear systems vs. modules

The corresponding modules are

$$\begin{aligned} M_{\mathrm{scl}} &:= \operatorname{coker}(D^{1\times 1} \xrightarrow{\mathtt{M}_{\mathrm{scl}}} D^{1\times 1}) \text{ and} \\ M_{\mathrm{sys}} &:= \operatorname{coker}(D^{1\times 3} \xrightarrow{\mathtt{M}_{\mathrm{sys}}} D^{1\times 3}). \end{aligned}$$

The modules M_{scl} and M_{sys} are indeed isomorphic:

$$\alpha : M_{\mathrm{scl}} \xrightarrow{\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}} M_{\mathrm{sys}}, \quad \beta : M_{\mathrm{sys}} \xrightarrow{\begin{pmatrix} 1 \\ \partial \\ \partial^2 \end{pmatrix}} M_{\mathrm{scl}}.$$

We call $M_{\mathrm{scl}} \cong M_{\mathrm{sys}}$ the module of **universal solutions.**

Linear systems vs. modules

The corresponding modules are

$$M_{\mathrm{scl}} := \mathrm{coker}\left(D^{1\times 1} \xrightarrow{\mathtt{M}_{\mathrm{scl}}} D^{1\times 1}\right) \text{ and}$$
$$M_{\mathrm{sys}} := \mathrm{coker}\left(D^{1\times 3} \xrightarrow{\mathtt{M}_{\mathrm{sys}}} D^{1\times 3}\right).$$

The modules M_{scl} and M_{sys} are indeed isomorphic:

$$\alpha : M_{\mathrm{scl}} \xrightarrow{\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}} M_{\mathrm{sys}}, \quad \beta : M_{\mathrm{sys}} \xrightarrow{\begin{pmatrix} 1 \\ \partial \\ \partial^2 \end{pmatrix}} M_{\mathrm{scl}}.$$

We call $M_{\mathrm{scl}} \cong M_{\mathrm{sys}}$ the module of **universal solutions.**

Linear systems vs. modules

The corresponding modules are

$$\begin{aligned} M_{\mathrm{scl}} &:= \operatorname{coker}(D^{1\times 1} \xrightarrow{\mathtt{M}_{\mathrm{scl}}} D^{1\times 1}) \text{ and} \\ M_{\mathrm{sys}} &:= \operatorname{coker}(D^{1\times 3} \xrightarrow{\mathtt{M}_{\mathrm{sys}}} D^{1\times 3}). \end{aligned}$$

The modules M_{scl} and M_{sys} are indeed isomorphic:

$$\alpha : M_{\mathrm{scl}} \xrightarrow{\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}} M_{\mathrm{sys}}, \quad \beta : M_{\mathrm{sys}} \xrightarrow{\begin{pmatrix} 1 \\ \partial \\ \partial^2 \end{pmatrix}} M_{\mathrm{scl}}.$$

We call $M_{\mathrm{scl}} \cong M_{\mathrm{sys}}$ the module of **universal solutions.**

Linear systems vs. modules

The corresponding modules are

$$\begin{aligned} M_{\mathrm{scl}} &:= \operatorname{coker}(D^{1\times 1} \xrightarrow{\mathtt{M}_{\mathrm{scl}}} D^{1\times 1}) \text{ and} \\ M_{\mathrm{sys}} &:= \operatorname{coker}(D^{1\times 3} \xrightarrow{\mathtt{M}_{\mathrm{sys}}} D^{1\times 3}). \end{aligned}$$

The modules M_{scl} and M_{sys} are indeed isomorphic:

$$\alpha : M_{\mathrm{scl}} \xrightarrow{\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}} M_{\mathrm{sys}}, \quad \beta : M_{\mathrm{sys}} \xrightarrow{\begin{pmatrix} 1 \\ \partial \\ \partial^2 \end{pmatrix}} M_{\mathrm{scl}}.$$

We call $M_{\mathrm{scl}} \cong M_{\mathrm{sys}}$ the module of **universal solutions.**

Triangular systems and cascade integration

Any filtration of M, i.e., a chain of submodules

$$0 \leq L \leq M$$

leads to a triangular presentation matrix

$$\mathtt{M} = \left(\begin{array}{c|c} \mathtt{N} & \eta \\ \hline 0 & \mathtt{L} \end{array}\right)$$

with $L = \operatorname{coker}(\mathtt{L})$ and $N = M/L \cong \operatorname{coker}(\mathtt{N})$.

Solving a triangular system $\rightsquigarrow$ cascade integration.

Triangular systems and cascade integration

Any filtration of M, i.e., a chain of submodules

$$0 \leq L \leq M$$

leads to a triangular presentation matrix

$$\mathtt{M} = \left(\begin{array}{c|c} \mathtt{N} & \eta \\ \hline 0 & \mathtt{L} \end{array}\right)$$

with $L = \operatorname{coker}(\mathtt{L})$ and $N = M/L \cong \operatorname{coker}(\mathtt{N})$.

Solving a triangular system $\rightsquigarrow$ cascade integration.

Overview

The evaluation map revisited

Recall, the evaluation map

$$\varepsilon : \begin{cases} M & \to & M^{**} \\ m & \mapsto & m \mapsto (\varphi \mapsto \varphi(m)) \end{cases} .$$

It is in general neither surjective nor injective. Its kernel

$$\mathrm{t}_{-1}(M) = \mathrm{t}(M) := \ker \varepsilon \qquad \text{(torsion submodule)}$$

yields the 2-step filtration

$$0 \leq \mathrm{t}(M) \leq M.$$

$M/\mathrm{t}(M)$	$\rightsquigarrow$	the underdetermined part of the system
$\mathrm{t}(M)$	$\rightsquigarrow$	the overdetermined part of the system

The evaluation map revisited

Recall, the evaluation map

$$\varepsilon : \begin{cases} M & \to & M^{**} \\ m & \mapsto & m \mapsto (\varphi \mapsto \varphi(m)) \end{cases} .$$

It is in general neither surjective nor injective. Its kernel

$$\mathrm{t}_{-1}(M) = \mathrm{t}(M) := \ker \varepsilon \qquad \text{(torsion submodule)}$$

yields the 2-step filtration

$$0 \leq \mathrm{t}(M) \leq M.$$

$M/\mathrm{t}(M)$	$\leftrightsquigarrow$	the underdetermined part of the system
$\mathrm{t}(M)$	$\leftrightsquigarrow$	the overdetermined part of the system

The evaluation map revisited

Recall, the evaluation map

$$\varepsilon : \begin{cases} M & \to & M^{**} \\ m & \mapsto & m \mapsto (\varphi \mapsto \varphi(m)) \end{cases} .$$

It is in general neither surjective nor injective. Its kernel

$$\mathrm{t}_{-1}(M) = \mathrm{t}(M) := \ker \varepsilon \qquad \text{(torsion submodule)}$$

yields the 2-step filtration

$$0 \leq \mathrm{t}(M) \leq M.$$

$M/\mathrm{t}(M)$	$\rightsquigarrow$	the underdetermined part of the system
$\mathrm{t}(M)$	$\rightsquigarrow$	the overdetermined part of the system

The evaluation map revisited

Recall, the evaluation map

$$\varepsilon : \begin{cases} M & \to & M^{**} \\ m & \mapsto & m \mapsto (\varphi \mapsto \varphi(m)) \end{cases} .$$

It is in general neither surjective nor injective. Its kernel

$$\mathrm{t}_{-1}(M) = \mathrm{t}(M) := \ker \varepsilon \qquad \text{(torsion submodule)}$$

yields the 2-step filtration

$$0 \leq \mathrm{t}(M) \leq M.$$

$M/\mathrm{t}(M)$ $\leadsto$ the **under**determined part of the system
$\mathrm{t}(M)$ $\leadsto$ the **over**determined part of the system

Purity filtration

Purity filtration

In fact, there is a finer filtration

$$\cdots \leq \mathrm{t}_{-(c+1)}(M) \leq \mathrm{t}_{-c}(M) \leq \cdots \leq \mathrm{t}_{-1}(M) \leq \mathrm{t}_0(M) := M$$

called the purity filtration. The graded part

$$M_c := \mathrm{t}_{-c}(M) / \mathrm{t}_{-(c+1)}(M)$$

is called pure of grade c.

Is there a way to recover the complete purity filtration?

Purity filtration

Purity filtration

In fact, there is a finer filtration

$$\cdots \leq \mathrm{t}_{-(c+1)}(M) \leq \mathrm{t}_{-c}(M) \leq \cdots \leq \mathrm{t}_{-1}(M) \leq \mathrm{t}_0(M) := M$$

called the purity filtration. The graded part

$$M_c := \mathrm{t}_{-c}(M) / \mathrm{t}_{-(c+1)}(M)$$

is called pure of grade c.

Is there a way to recover the complete purity filtration?

Purity filtration

Purity filtration

In fact, there is a finer filtration

$$\cdots \leq \mathrm{t}_{-(c+1)}(M) \leq \mathrm{t}_{-c}(M) \leq \cdots \leq \mathrm{t}_{-1}(M) \leq \mathrm{t}_0(M) := M$$

called the purity filtration. The graded part

$$M_c := \mathrm{t}_{-c}(M) / \mathrm{t}_{-(c+1)}(M)$$

is called pure of grade c.

Is there a way to recover the complete purity filtration?

Functors prefer complexes with adapted objects

Part 1 of the answer: The derived category formalism

The target of the evaluation map $\varepsilon : M \to M^{**}$ involves applying two functors to the module M:

$$M^{**} := F(G(M)), \text{ with}$$

$$F := \mathrm{Hom}_D(-, D_D), G := \mathrm{Hom}_D(-, {}_DD)$$

- Since VERDIER's thesis we know that it is better to apply functors to complexes with adapted objects rather than to modules.

Functors prefer complexes with adapted objects

Part 1 of the answer: The derived category formalism

The target of the evaluation map $\varepsilon : M \to M^{**}$ involves applying two functors to the module M:

$$M^{**} := F(G(M)), \text{ with}$$

$$F := \operatorname{Hom}_D(-, D_D), G := \operatorname{Hom}_D(-, {}_D D)$$

- Since VERDIER's thesis we know that it is better to apply functors to complexes with adapted objects rather than to modules.

A free resolution

Let $D = R[\partial_x, \partial_y, \partial_z]$ and $M = D/\langle \partial_x, \partial_y, \partial_z \rangle$ with resolution $P_\bullet$:

$$
\begin{array}{ccccccccc}
D^{1\times 1} & \xleftarrow{\overbrace{\begin{pmatrix} \partial_x \\ \partial_y \\ \partial_z \end{pmatrix}}^{\mathrm{grad}}} & D^{1\times 3} & \xleftarrow{\overbrace{\begin{pmatrix} 0 & \partial_z & -\partial_y \\ -\partial_z & 0 & \partial_x \\ \partial_y & -\partial_x & 0 \end{pmatrix}}^{\mathrm{rot}}} & D^{1\times 3} & \xleftarrow{\overbrace{\begin{pmatrix} \partial_x & \partial_y & \partial_z \end{pmatrix}}^{\mathrm{div}}} & D^{1\times 1} & \longleftarrow & 0 \\
\downarrow & & & & & & & & \\
M & & & & & & & & \\
\downarrow & & & & & & & & \\
0 & & & & & & & &
\end{array}
$$

- $P_\bullet$: adapted objects but complicated morphisms
- M: complicated object

A free resolution

Let $D = R[\partial_x, \partial_y, \partial_z]$ and $M = D/\langle \partial_x, \partial_y, \partial_z \rangle$ with resolution $P_\bullet$:

$$
\begin{array}{ccccccccccc}
0 & \longleftarrow & D^{1\times 1} & \xleftarrow{\overbrace{\begin{pmatrix} \partial_x \\ \partial_y \\ \partial_z \end{pmatrix}}^{\text{grad}}} & D^{1\times 3} & \xleftarrow{\overbrace{\begin{pmatrix} 0 & \partial_z & -\partial_y \\ -\partial_z & 0 & \partial_x \\ \partial_y & -\partial_x & 0 \end{pmatrix}}^{\text{rot}}} & D^{1\times 3} & \xleftarrow{\overbrace{\begin{pmatrix} \partial_x & \partial_y & \partial_z \end{pmatrix}}^{\text{div}}} & D^{1\times 1} & \longleftarrow & 0 \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
0 & \longleftarrow & M & \longleftarrow & 0 & \longleftarrow & 0 & \longleftarrow & 0 & \longleftarrow & 0
\end{array}
$$

- $P_\bullet$: adapted objects but complicated morphisms
- M: complicated object

The module vs. its resolution

Let us demonstrate VERDIER's statement on the dualizing functors $G := \mathrm{Hom}_D(-, {}_D D)$ and $F := \mathrm{Hom}_D(-, D_D)$:

$$M = D/\langle \partial_x, \partial_y, \partial_z \rangle \neq 0$$

$$P_\bullet \quad : \qquad 0 \longleftarrow D^{1\times 1} \xleftarrow{\begin{pmatrix} \partial_x \\ \partial_y \\ \partial_z \end{pmatrix}} D^{1\times 3} \longleftarrow D^{1\times 3} \longleftarrow D^{1\times 1} \longleftarrow 0$$

$M^{**} = 0$ while $P^{**} \cong P.$

The module vs. its resolution

Let us demonstrate VERDIER's statement on the dualizing functors $G := \mathrm{Hom}_D(-, {}_D D)$ and $F := \mathrm{Hom}_D(-, D_D)$:

$$M^* = \mathrm{Hom}(M, D) = 0$$

$$P_\bullet^* \quad : \qquad 0 \longrightarrow D^{1\times 1} \xrightarrow{\begin{pmatrix} \partial_x \\ \partial_y \\ \partial_z \end{pmatrix}} D^{3\times 1} \longrightarrow D^{3\times 1} \longrightarrow D^{1\times 1} \longrightarrow 0$$

$M^{**} = 0$ while $P^{**} \cong P$.

The module vs. its resolution

Let us demonstrate VERDIER's statement on the dualizing functors $G := \mathrm{Hom}_D(-, {}_D D)$ and $F := \mathrm{Hom}_D(-, D_D)$:

$$M^{**} = \mathrm{Hom}(\mathrm{Hom}(M, D), D) = 0$$

$$P_\bullet^{**}: \quad 0 \longleftarrow D^{1\times 1} \xleftarrow{\begin{pmatrix}\partial_x \\ \partial_y \\ \partial_z\end{pmatrix}} D^{1\times 3} \longleftarrow D^{1\times 3} \longleftarrow D^{1\times 1} \longleftarrow 0$$

$$M^{**} = 0 \quad \text{while} \quad P^{**} \cong P.$$

The module vs. its resolution

Let us demonstrate VERDIER's statement on the dualizing functors $G := \mathrm{Hom}_D(-, {}_D D)$ and $F := \mathrm{Hom}_D(-, D_D)$:

$$P_\bullet^{**} : \qquad 0 \longleftarrow D^{1\times 1} \xleftarrow{\begin{pmatrix}\partial_x\\ \partial_y\\ \partial_z\end{pmatrix}} D^{1\times 3} \longleftarrow D^{1\times 3} \longleftarrow D^{1\times 1} \longleftarrow 0$$

$$M^{**} = 0 \quad \text{while} \quad P^{**} \cong P.$$

Quasi-isomorhpisms

- Any projective resolution $P_\bullet : 0 \leftarrow P_0 \leftarrow P_1 \leftarrow P_2 \leftarrow \cdots$ of a module M is such a complex with adapted objects.
- The natural epimorphism $M \overset{\nu}{\twoheadleftarrow} P_0$ induces a quasi-isomorphism of complexes, where M is now regarded as a complex concentrated in degree 0:

$$\begin{array}{ccccccccccc} P_\bullet & : & 0 & \longleftarrow & P_0 & \longleftarrow & P_1 & \longleftarrow & P_2 & \longleftarrow & \cdots \\ \nu \downarrow \cong & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \\ M & : & 0 & \longleftarrow & M & \longleftarrow & 0 & \longleftarrow & 0 & \longleftarrow & \cdots \end{array}$$

Quasi-isomorhpisms

- Any projective resolution $P_\bullet : 0 \leftarrow P_0 \leftarrow P_1 \leftarrow P_2 \leftarrow \cdots$ of a module M is such a complex with adapted objects.
- The natural epimorphism $M \overset{\nu}{\twoheadleftarrow} P_0$ induces a quasi-isomorphism of complexes, where M is now regarded as a complex concentrated in degree 0:

$$\begin{array}{ccccccccccc}
P_\bullet & : & 0 & \longleftarrow & P_0 & \longleftarrow & P_1 & \longleftarrow & P_2 & \longleftarrow & \cdots \\
\nu \downarrow \cong & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \\
M & : & 0 & \longleftarrow & M & \longleftarrow & 0 & \longleftarrow & 0 & \longleftarrow & \cdots
\end{array}$$

Functors prefer complexes with adapted objects

Recall, we are interested in a substitute for the evaluation map

$$\varepsilon : M \to M^{**} = F(G(M))$$

which yields the complete purity filtration.

Answer:

- Instead of the module $G(M)$ consider the complex $G(P_\bullet)$.
- Instead of applying F to the complex $G(P_\bullet)$ apply it to a resolution $T^\bullet$ of $G(P_\bullet)$.
- Fortunately there is a resolution $T^\bullet$ of $G(P_\bullet)$ which is a bifiltered complex.
- It is the **total complex** of the so-called Cartan-Eilenberg **bicomplex**.

Functors prefer complexes with adapted objects

Recall, we are interested in a substitute for the evaluation map

$$\varepsilon : M \to M^{**} = F(G(M))$$

which yields the complete purity filtration.

Answer:

- Instead of the module $G(M)$ consider the complex $G(P_\bullet)$.
- Instead of applying F to the complex $G(P_\bullet)$ apply it to a resolution $T^\bullet$ of $G(P_\bullet)$.
- Fortunately there is a resolution $T^\bullet$ of $G(P_\bullet)$ which is a bifiltered complex.
- It is the **total complex** of the so-called Cartan-Eilenberg **bicomplex**.

Functors prefer complexes with adapted objects

Recall, we are interested in a substitute for the evaluation map

$$\varepsilon : M \to M^{**} = F(G(M))$$

which yields the complete purity filtration.

Answer:

- Instead of the module $G(M)$ consider the complex $G(P_\bullet)$.
- Instead of applying F to the complex $G(P_\bullet)$ apply it to a resolution $T^\bullet$ of $G(P_\bullet)$.
- Fortunately there is a resolution $T^\bullet$ of $G(P_\bullet)$ which is a bifiltered complex.
- It is the **total complex** of the so-called **Cartan-Eilenberg bicomplex**.

Functors prefer complexes with adapted objects

Recall, we are interested in a substitute for the evaluation map

$$\varepsilon : M \to M^{**} = F(G(M))$$

which yields the complete purity filtration.

Answer:

- Instead of the module $G(M)$ consider the complex $G(P_\bullet)$.
- Instead of applying F to the complex $G(P_\bullet)$ apply it to a resolution $T^\bullet$ of $G(P_\bullet)$.
- Fortunately there is a resolution $T^\bullet$ of $G(P_\bullet)$ which is a bifiltered complex.
- It is the **total complex** of the so-called Cartan-Eilenberg **bicomplex**.

Functors prefer complexes with adapted objects

Recall, we are interested in a substitute for the evaluation map

$$\varepsilon : M \to M^{**} = F(G(M))$$

which yields the complete purity filtration.

Answer:

- Instead of the module $G(M)$ consider the complex $G(P_\bullet)$.
- Instead of applying F to the complex $G(P_\bullet)$ apply it to a resolution $T^\bullet$ of $G(P_\bullet)$.
- Fortunately there is a resolution $T^\bullet$ of $G(P_\bullet)$ which is a bifiltered complex.
- It is the **total complex** of the so-called **Cartan-Eilenberg bicomplex**.

The CARTAN-EILENBERG resolution

$$G(P_\bullet): \qquad G(P_0) \longrightarrow G(P_1) \longrightarrow G(P_2) \longrightarrow \cdots$$

The CARTAN-EILENBERG resolution

$$\begin{array}{llllllll}
G(P_\bullet): & G(P_0) & \longrightarrow & G(P_1) & \longrightarrow & G(P_2) & \longrightarrow & \cdots \\
 & \uparrow & & \uparrow & & \uparrow & & \\
B^{\bullet 0}: & B^{0,0} & \longrightarrow & B^{1,0} & \longrightarrow & B^{2,0} & \longrightarrow & \cdots
\end{array}$$

The CARTAN-EILENBERG resolution

$$
\begin{array}{llllllll}
G(P_\bullet): & G(P_0) & \longrightarrow & G(P_1) & \longrightarrow & G(P_2) & \longrightarrow & \cdots \\
 & & & & & & & \\
B^{\bullet\bullet}: & B^{0,0} & \longrightarrow & B^{1,0} & \longrightarrow & B^{2,0} & \longrightarrow & \cdots \\
 & \uparrow & & \uparrow & & \uparrow & & \\
 & B^{0,-1} & \longrightarrow & B^{1,-1} & \longrightarrow & B^{2,-1} & \longrightarrow & \cdots \\
 & \uparrow & & \uparrow & & \uparrow & & \\
 & B^{0,-2} & \longrightarrow & B^{1,-2} & \longrightarrow & B^{2,-2} & \longrightarrow & \cdots \\
 & \uparrow & & \uparrow & & \uparrow & & \\
 & \vdots & & \vdots & & \vdots & &
\end{array}
$$

The CARTAN-EILENBERG resolution

$$G(P_\bullet): \qquad G(P_0) \longrightarrow G(P_1) \longrightarrow G(P_2) \longrightarrow \cdots$$

$$
\begin{array}{llccccccc}
B^{\bullet\bullet}: & & B^{0,0} & \longrightarrow & B^{1,0} & \longrightarrow & B^{2,0} & \longrightarrow & \cdots \\
& & \uparrow & 0 & \uparrow & 1 & \uparrow & 2 & \\
& & B^{0,-1} & \longrightarrow & B^{1,-1} & \longrightarrow & B^{2,-1} & \longrightarrow & \cdots \\
& & \uparrow & -1 & \uparrow & 0 & \uparrow & 1 & \\
T^\bullet := \mathrm{Tot}(B^{\bullet\bullet}): & & B^{0,-2} & \longrightarrow & B^{1,-2} & \longrightarrow & B^{2,-2} & \longrightarrow & \cdots \\
& & \uparrow & -2 & \uparrow & -1 & \uparrow & 0 & \\
& & \vdots & & \vdots & & \vdots & & \ddots
\end{array}
$$

The CARTAN-EILENBERG resolution

$$\begin{array}{ccccccccccccccc}
G(P_\bullet): & \cdots & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & G(P_0) & \rightarrow & G(P_1) & \rightarrow & G(P_2) & \rightarrow & \cdots \\
\cong\uparrow & & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \\
T^\bullet: & \cdots & \rightarrow & T^{-2} & \rightarrow & T^{-1} & \longrightarrow & T^0 & \longrightarrow & T^1 & \longrightarrow & T^2 & \longrightarrow & \cdots
\end{array}$$

- The total complex $T^\bullet$ of the CARTAN-EILENBERG bicomplex $B^{\bullet\bullet}$ is bifiltered in a natural way.

The Cartan-Eilenberg resolution

$$
\begin{array}{ccccccccccccc}
G(P_\bullet): & \cdots & \longrightarrow & 0 & \longrightarrow & 0 & \longrightarrow & G(P_0) & \longrightarrow & G(P_1) & \longrightarrow & G(P_2) & \longrightarrow \cdots \\
\cong\uparrow & & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & \\
T^\bullet: & \cdots & \longrightarrow & T^{-2} & \longrightarrow & T^{-1} & \longrightarrow & T^0 & \longrightarrow & T^1 & \longrightarrow & T^2 & \longrightarrow \cdots
\end{array}
$$

- The total complex $T^\bullet$ of the Cartan-Eilenberg bicomplex $B^{\bullet\bullet}$ is bifiltered in a natural way.

The Cartan-Eilenberg resolution

Induced filtrations

- M is the only homology of the bifiltered complex $F(T^\bullet)$. In other words there is a quasi-isomorphism

$$\varepsilon_\bullet : M \longrightarrow F(T^\bullet) .$$

  In particular, M is a subfactor of the bifiltered module $F(T^0)$.
- Any filtration of an object in an Abelian category induced a filtration on any of its subfactor objects.
- Hence, any filtration on a complex induces a filtration on its (co)homology objects.

The CARTAN-EILENBERG resolution

Induced filtrations

- M is the only homology of the bifiltered complex $F(T^\bullet)$. In other words there is a quasi-isomorphism
$$\varepsilon_\bullet : M \longrightarrow F(T^\bullet) .$$
In particular, M is a subfactor of the bifiltered module $F(T^0)$.
- Any filtration of an object in an ABELian category induced a filtration on any of its subfactor objects.
- Hence, any filtration on a complex induces a filtration on its (co)homology objects.

The CARTAN-EILENBERG resolution

Induced filtrations

- M is the only homology of the bifiltered complex $F(T^\bullet)$. In other words there is a quasi-isomorphism

$$\varepsilon_\bullet : M \longrightarrow F(T^\bullet).$$

  In particular, M is a subfactor of the bifiltered module $F(T^0)$.
- Any filtration of an object in an ABELian category induced a filtration on any of its subfactor objects.
- Hence, any filtration on a complex induces a filtration on its (co)homology objects.

The CARTAN-EILENBERG resolution

Induced filtrations

- M is the only homology of the bifiltered complex $F(T^\bullet)$. In other words there is a quasi-isomorphism
$$\varepsilon_\bullet : M \longrightarrow F(T^\bullet).$$
In particular, M is a subfactor of the bifiltered module $F(T^0)$.
- Any filtration of an object in an ABELian category induced a filtration on any of its subfactor objects.
- Hence, any filtration on a complex induces a filtration on its (co)homology objects.

The purity filtration

Solution

The isomorphism

$$\varepsilon_\bullet : M \longrightarrow F(T^\bullet)$$

provides the solution:
The first filtration on $F(T^\bullet)$ induces the trivial filtration on its homology M, while the second filtration induces the **purity filtration** on M.

This filtration can be computed directly or in terms of spectral sequences:

$$E^2_{pq} = \mathrm{Ext}^{-p}(\mathrm{Ext}^q(M, D), D)) \Longrightarrow M \text{ for } p + q = 0.$$

The purity filtration

Solution

The isomorphism

$$\varepsilon_\bullet : M \longrightarrow F(T^\bullet)$$

provides the solution:
The first filtration on $F(T^\bullet)$ induces the trivial filtration on its homology M, while the second filtration induces the **purity filtration** on M.

This filtration can be computed directly or in terms of spectral sequences:

$$E^2_{pq} = \mathrm{Ext}^{-p}(\mathrm{Ext}^q(M, D), D)) \Longrightarrow M \text{ for } p + q = 0.$$

The purity filtration

Solution

The isomorphism

$$\varepsilon_\bullet : M \longrightarrow F(T^\bullet)$$

provides the solution:
The first filtration on $F(T^\bullet)$ induces the trivial filtration on its homology M, while the second filtration induces the **purity filtration** on M.

This filtration can be computed directly or in terms of spectral sequences:

$$E^2_{pq} = \mathrm{Ext}^{-p}(\mathrm{Ext}^q(M,D),D)) \Longrightarrow M \text{ for } p+q=0.$$

The purity filtration

Solution

The isomorphism

$$\varepsilon_\bullet : M \longrightarrow F(T^\bullet)$$

provides the solution:
The first filtration on $F(T^\bullet)$ induces the trivial filtration on its homology M, while the second filtration induces the **purity filtration** on M.

This filtration can be computed directly or in terms of spectral sequences:

$$E^2_{pq} = \mathrm{Ext}^{-p}(\mathrm{Ext}^q(M, D), D)) \Longrightarrow M \text{ for } p + q = 0.$$

Computability

All the above can be made computable in any ABELian category, in which all existential quantifiers appearing in the axioms can be turned into constructive ones [Bar]. The `homalg` project [hpa11] provides an implementation in this generality.

Definitions

- We call such an ABELian category computable.
- We call a ring D computable, if one can algorithmically solve (in)homogeneous linear systems over D.

[BLH11, Theorem 3.4]

The category of finitely presented modules over a computable ring D is ABELian and as such computable.

Computability

All the above can be made computable in any ABELian category, in which all existential quantifiers appearing in the axioms can be turned into constructive ones [Bar]. The `homalg` project [hpa11] provides an implementation in this generality.

Definitions

- We call such an ABELian category computable.
- We call a ring D computable, if one can algorithmically solve (in)homogeneous linear systems over D.

[BLH11, Theorem 3.4]

The category of finitely presented modules over a computable ring D is ABELian and as such computable.

Computability

All the above can be made computable in any ABELian category, in which all existential quantifiers appearing in the axioms can be turned into constructive ones [Bar]. The `homalg` project [hpa11] provides an implementation in this generality.

Definitions

- We call such an ABELian category computable.
- We call a ring D computable, if one can algorithmically solve (in)homogeneous linear systems over D.

[BLH11, Theorem 3.4]

The category of finitely presented modules over a computable ring D is ABELian and as such computable.

Computability

All the above can be made computable in any ABELian category, in which all existential quantifiers appearing in the axioms can be turned into constructive ones [Bar]. The `homalg` project [hpa11] provides an implementation in this generality.

Definitions

- We call such an ABELian category computable.
- We call a ring D computable, if one can algorithmically solve (in)homogeneous linear systems over D.

[BLH11, Theorem 3.4]

The category of finitely presented modules over a computable ring D is ABELian and as such computable.

Computability

All the above can be made computable in any ABELian category, in which all existential quantifiers appearing in the axioms can be turned into constructive ones [Bar]. The `homalg` project [hpa11] provides an implementation in this generality.

Definitions

- We call such an ABELian category computable.
- We call a ring D computable, if one can algorithmically solve (in)homogeneous linear systems over D.

[BLH11, Theorem 3.4]

The category of finitely presented modules over a computable ring D is ABELian and as such computable.

An example over the WEYL algebra

Example: A module M over $D = k[x,y,z][\partial_x,\partial_y,\partial_z]$

The purity filtration of the module

$$M := \operatorname{coker} \begin{pmatrix} \partial_y\partial_z - \frac{1}{3}\partial_z^2 + \frac{1}{3}\partial_x + \partial_y - \frac{1}{3}\partial_z & \partial_y\partial_z - \frac{1}{3}\partial_z^2 \\ \partial_x\partial_z + \partial_z^2 + \partial_z & \partial_x\partial_z + \partial_z^2 \\ \partial_z^2 - \partial_x + \partial_z & 3\partial_x\partial_y + \partial_z^2 \\ \partial_x\partial_y & 0 \\ \partial_z^2 - \partial_x + \partial_z & -3\partial_x^2 + \partial_z^2 \\ \partial_x^2 & 0 \\ x\partial_z^2 - x\partial_x + \frac{3}{2}\partial_x + x\partial_z + \frac{3}{2}\partial_z + \frac{3}{2} & x\partial_z^2 + \frac{3}{2}\partial_x + \frac{3}{2}\partial_z \\ \partial_z^3 + 2\partial_z^2 + \partial_z & \partial_z^3 + \partial_x\partial_z + \partial_z^2 \end{pmatrix}$$

An example over the WEYL algebra

Example: A module M over $D = k[x, y, z][\partial_x, \partial_y, \partial_z]$

is given by an isomorphism onto

$$H_0(\mathrm{Tot}_\bullet B_{\bullet\bullet}) = \operatorname{coker} \begin{pmatrix} \partial_x & 1 & \cdot \\ \cdot & \partial_x & 1 \\ \cdot & \partial_y & 1 \\ \cdot & \cdot & \partial_x \\ \cdot & \cdot & \partial_y \\ \cdot & \cdot & \partial_z \end{pmatrix}.$$

An example over the WEYL algebra

Example: A module M over $D = k[x,y,z][\partial_x,\partial_y,\partial_z]$

$$\operatorname{coker}\begin{pmatrix} \boxed{\partial_x} & 1 & \cdot \\ \cdot & \boxed{\begin{matrix}\partial_x \\ \partial_y\end{matrix}} & 1 \\ \cdot & & 1 \\ \cdot & \cdot & \boxed{\begin{matrix}\partial_x \\ \partial_y \\ \partial_z\end{matrix}} \\ \cdot & \cdot & \\ \cdot & \cdot & \end{pmatrix} \xrightarrow{\cong} \operatorname{coker}\begin{pmatrix} \partial_x^2 - \partial_x\partial_y \\ \partial_x\partial_y^2 \\ \partial_x\partial_y\partial_z \end{pmatrix} =: C$$

has the general solution:

$$u(x,y,z) = C_1(y,z) + (x+y)C_2(z) + \bar{C}_2(z) + \frac{x^2+2xy+y^2}{2}C_3.$$

An example over the WEYL algebra

Example: A module M over $D = k[x,y,z][\partial_x,\partial_y,\partial_z]$

$$\operatorname{coker}\begin{pmatrix} \boxed{\partial_x} & 1 & \cdot \\ \cdot & \boxed{\begin{matrix}\partial_x \\ \partial_y\end{matrix}} & \begin{matrix}1 \\ 1\end{matrix} \\ \cdot & \cdot & \\ \cdot & \cdot & \boxed{\begin{matrix}\partial_x \\ \partial_y \\ \partial_z\end{matrix}} \\ \cdot & \cdot & \end{pmatrix} \xrightarrow{\cong} \operatorname{coker}\begin{pmatrix} \partial_x^2 - \partial_x\partial_y \\ \partial_x\partial_y^2 \\ \partial_x\partial_y\partial_z \end{pmatrix} =: C$$

has the general solution:

$$u(x,y,z) = C_1(y,z) + (x+y)C_2(z) + \bar{C}_2(z) + \frac{x^2+2xy+y^2}{2}C_3.$$

An example over the Weyl algebra

Example: A module M over $D = k[x,y,z][\partial_x,\partial_y,\partial_z]$

Leading to the isomorphism

$$M \xrightarrow{L} C = \operatorname{coker} \begin{pmatrix} \partial_x^2 - \partial_x\partial_y \\ \partial_x\partial_y^2 \\ \partial_x\partial_y\partial_z \end{pmatrix}$$

with

$$L = \begin{pmatrix} 2x\partial_x\partial_y - \partial_x - \partial_z \\ -2x\partial_x\partial_y + \partial_x + \partial_z + 1 \end{pmatrix}.$$

The desired general solution of the original system is then Lu.

An example over the WEYL algebra

Example: A module M over $D = k[x,y,z][\partial_x, \partial_y, \partial_z]$

Leading to the isomorphism

$$M \xrightarrow{L} C = \operatorname{coker} \begin{pmatrix} \partial_x^2 - \partial_x\partial_y \\ \partial_x\partial_y^2 \\ \partial_x\partial_y\partial_z \end{pmatrix}$$

with

$$L = \begin{pmatrix} 2x\partial_x\partial_y - \partial_x - \partial_z \\ -2x\partial_x\partial_y + \partial_x + \partial_z + 1 \end{pmatrix}.$$

The desired general solution of the original system is then Lu.

Why are we abstracting?

We can now write programs for Abelian categories:

Abelian algorithms

- Computers appreciate abstractness
  - Code simplicity
  - Maintenance
  - Extensibility
- Widens the range of applicatons

Why are we abstracting?

We can now write programs for ABELian categories:

ABELian algorithms

- Computers appreciate abstractness
  - Code simplicity
  - Maintenance
  - Extensibility
- Widens the range of applicatons

Why are we abstracting?

Using the same abstract implementation we want to compute the **Beilinson spectral sequence** of the bicomplex

$$B^e = \bigoplus_j H^j(\mathcal{F}(e-j)) \otimes \Omega^{j-e}_{\mathbb{P}^n}(j-e)$$

and the induced filtration on the sheaf $\mathcal{F} \in \mathfrak{Coh}(\mathbb{P}^n)$.

Thank you for your attention

Mohamed Barakat, *Spectral filtrations via generalized morphisms*, submitted (`arXiv:0904.0240`).

Mohamed Barakat and Markus Lange-Hegermann, *An axiomatic setup for algorithmic homological algebra and an alternative approach to localization*, J. Algebra Appl. **10** (2011), no. 2, 269–293, (`arXiv:1003.1943`).

The homalg project authors, *The* `homalg` *project*, 2003-2011, (`http://homalg.math.rwth-aachen.de/`).