Theory group DESY Zeuthen

Analysing first order factorizable systems of differential equations in one variable

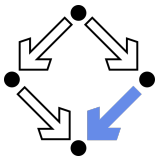
Nikolai Fadeev

Research Institute for Symbolic Computation (RISC)

June 3, 2022

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764850 (SAGEX).

SAGEX
Scattering Amplitudes:
from Geometry to Experiment



JKU
JOHANNES KEPLER
UNIVERSITY LINZ

Motivation

Amplitude computation in Quantum Field Theory (QFT)

- If we want to compute the amplitude up to some order/loops $\mathcal{O}(\alpha^n)$, we need to compute all the Feynman integrals up to $\mathcal{O}(\alpha^n)$

Motivation

Amplitude computation in Quantum Field Theory (QFT)

- If we want to compute the amplitude up to some order/loops $\mathcal{O}(\alpha^n)$, we need to compute all the Feynman integrals up to $\mathcal{O}(\alpha^n)$
- The first thing we need to do is to reduce as much as possible the number of integrals $\mathcal{I}$ that we need to compute.

Motivation

Amplitude computation in Quantum Field Theory (QFT)

- If we want to compute the amplitude up to some order/loops $\mathcal{O}(\alpha^n)$, we need to compute all the Feynman integrals up to $\mathcal{O}(\alpha^n)$
- The first thing we need to do is to reduce as much as possible the number of integrals $\mathcal{I}$ that we need to compute.
- For that, we use the **integration-by-parts** (IBP) method to reduce the thousands of initial Feynman integrals to a handful of **master integrals** (MIs) $\mathcal{I} = (\mathcal{I}_1(x,\epsilon), \ldots, \mathcal{I}_n(x,\epsilon))$ (with $s = -\frac{(1-x)^2}{x}$ and $\epsilon = D-4$ the dimensional parameter) and a system in the MIs $\mathcal{I}$:

$$\frac{d}{dx}\mathcal{I}(x,\epsilon) = \mathcal{M}(x,\epsilon)\mathcal{I}(x,\epsilon) + \mathcal{R}(x,\epsilon)$$

where the $\mathcal{R}(x,\epsilon)$ are new integrals called the **base integrals** (BIs).

Motivation

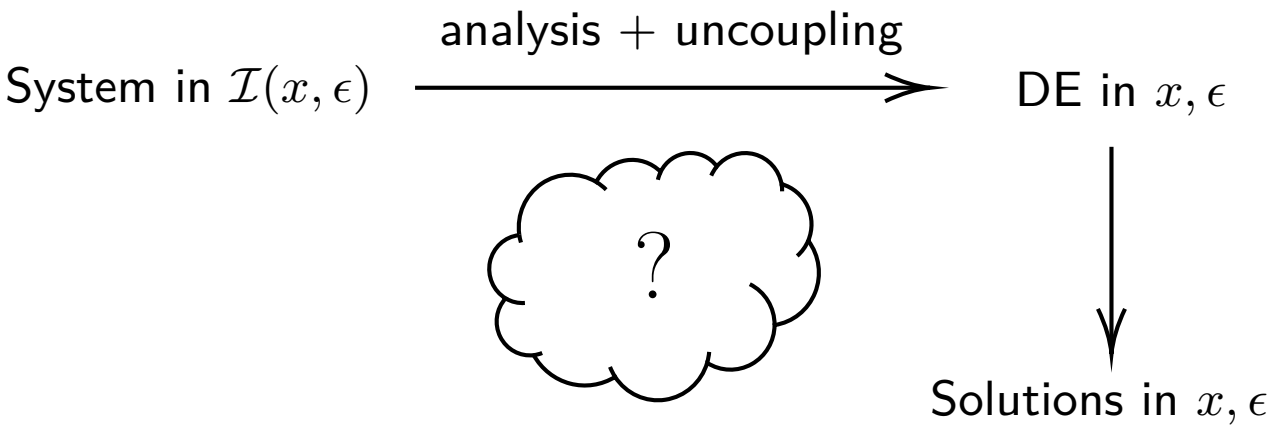
Amplitude computation in Quantum Field Theory (QFT)

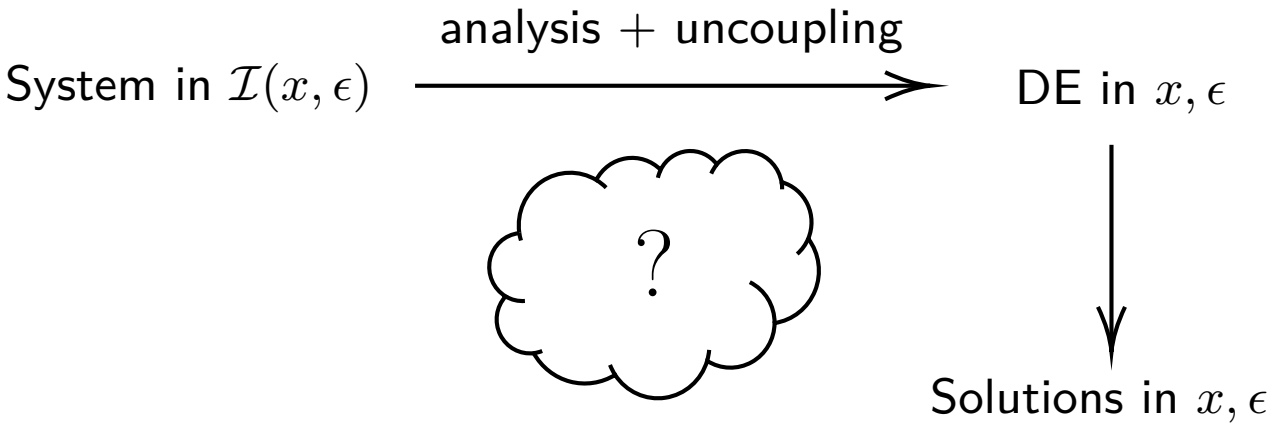
- If we want to compute the amplitude up to some order/loops $\mathcal{O}(\alpha^n)$, we need to compute all the Feynman integrals up to $\mathcal{O}(\alpha^n)$
- The first thing we need to do is to reduce as much as possible the number of integrals $\mathcal{I}$ that we need to compute.
- For that, we use the **integration-by-parts** (IBP) method to reduce the thousands of initial Feynman integrals to a handful of **master integrals** (MIs) $\mathcal{I} = (\mathcal{I}_1(x,\epsilon), \ldots, \mathcal{I}_n(x,\epsilon))$ (with $s = -\frac{(1-x)^2}{x}$ and $\epsilon = D-4$ the dimensional parameter) and a system in the MIs $\mathcal{I}$:

$$\frac{d}{dx}\mathcal{I}(x,\epsilon) = \mathcal{M}(x,\epsilon)\mathcal{I}(x,\epsilon) + \mathcal{R}(x,\epsilon)$$

where the $\mathcal{R}(x,\epsilon)$ are new integrals called the **base integrals** (BIs).

Aim: We need an algorithm to preprocess and solve this system of first order differential equations.


analysis + uncoupling
System in $\mathcal{I}(x, \epsilon)$
DE in x, ϵ
?
Solutions in x, ϵ



- In this presentation, we will concentrate on the preprocessing analysis algorithm, that has been the main focus of our recent work.

1. General setting
   - Assumptions
   - Main ideas for solving the system
   - Why preprocessing the system?
2. Finding the best uncouplings
   - Clustering the system
   - Zürcher scheme
   - Gauß scheme
3. Computing the $\epsilon-$orders for the whole system
   - Local extraction of the $\epsilon-$orders
   - Global correction and trees
   - Building trees in different schemes

- Differentiation/IBP of MIs $\rightarrow$ $\underbrace{\text{elements of } \mathcal{I}}_{\mathcal{MI}} + \underbrace{\text{new integrals (Base Integrals – BIs)}}_{\mathcal{R} \text{ (inhomogeneous part)}}$

$$\frac{d}{dx}\mathcal{I}(x,\epsilon) = \mathcal{M}(x,\epsilon)\mathcal{I}(x,\epsilon) + \mathcal{R}(x,\epsilon) \tag{1}$$

- The key ingredient is the **uncoupling** of the system, i.e. transforming the first-order system into one or several **higher order differential equations** (HODEs) in MIs and several **linear algebraic relations** (LFs) for the other MIs

- Differentiation/IBP of MIs $\rightarrow$ $\underbrace{\text{elements of } \mathcal{I}}_{\mathcal{MI}} + \underbrace{\text{new integrals (Base Integrals – BIs)}}_{\mathcal{R} \text{ (inhomogeneous part)}}$

$$\frac{d}{dx}\mathcal{I}(x,\epsilon) = \mathcal{M}(x,\epsilon)\mathcal{I}(x,\epsilon) + \mathcal{R}(x,\epsilon) \tag{1}$$

- The key ingredient is the **uncoupling** of the system, i.e. transforming the first-order system into one or several **higher order differential equations** (HODEs) in MIs and several **linear algebraic relations** (LFs) for the other MIs
- There exists **several schemes** that allow to solve the system order by order in ϵ, the main difference between them being the moment when we expand in ϵ, i.e. before uncoupling the system or after

- Differentiation/IBP of MIs $\rightarrow$ $\underbrace{\text{elements of } \mathcal{I}}_{\mathcal{MI}} + \underbrace{\text{new integrals (Base Integrals – BIs)}}_{\mathcal{R} \text{ (inhomogeneous part)}}$

$$\frac{d}{dx}\mathcal{I}(x,\epsilon) = \mathcal{M}(x,\epsilon)\mathcal{I}(x,\epsilon) + \mathcal{R}(x,\epsilon) \tag{1}$$

- The key ingredient is the **uncoupling** of the system, i.e. transforming the first-order system into one or several **higher order differential equations** (HODEs) in MIs and several **linear algebraic relations** (LFs) for the other MIs
- There exists **several schemes** that allow to solve the system order by order in ϵ, the main difference between them being the moment when we expand in ϵ, i.e. before uncoupling the system or after
- Some of them will rely on **differential equations** only, some of them make use of **recurrence/difference equations** (and therefore `Sigma`)

- Differentiation/IBP of MIs $\rightarrow$ $\underbrace{\text{elements of } \mathcal{I}}_{\mathcal{MI}} + \underbrace{\text{new integrals (Base Integrals – BIs)}}_{\mathcal{R} \text{ (inhomogeneous part)}}$

$$\frac{d}{dx}\mathcal{I}(x,\epsilon) = \mathcal{M}(x,\epsilon)\mathcal{I}(x,\epsilon) + \mathcal{R}(x,\epsilon) \tag{1}$$

- The key ingredient is the **uncoupling** of the system, i.e. transforming the first-order system into one or several **higher order differential equations** (HODEs) in MIs and several **linear algebraic relations** (LFs) for the other MIs
- There exists **several schemes** that allow to solve the system order by order in ϵ, the main difference between them being the moment when we expand in ϵ, i.e. before uncoupling the system or after
- Some of them will rely on **differential equations** only, some of them make use of **recurrence/difference equations** (and therefore `Sigma`)
- We concentrate here on the scheme where we expand in ϵ only in the latest stages of the computation and solve at this point recurrence equations

- In order to solve our system, it has to verify several general assumptions, which are the following:

Assumptions

1. $\mathcal{M}(x, \epsilon) \in \mathcal{M}_n(\mathbb{K}(x, \epsilon))$ ($\mathbb{K}$ is a computable subfield of $\mathbb{R}$) is invertible

- In order to solve our system, it has to verify several general assumptions, which are the following:

Assumptions

1. $\mathcal{M}(x,\epsilon) \in \mathcal{M}_n(\mathbb{K}(x,\epsilon))$ ($\mathbb{K}$ is a computable subfield of $\mathbb{R}$) is invertible
2. Each element of $\mathcal{R}(x,\epsilon) = (\mathcal{R}_1(x,\epsilon),\ldots,\mathcal{R}_n(x,\epsilon))$ and $\mathcal{I}(x,\epsilon) = (\mathcal{I}_1(x,\epsilon),\ldots,\mathcal{I}_n(x,\epsilon))$ can be expanded as a **power series** in x:

$$\forall i \in \{1,\ldots,n\} \quad \mathcal{R}_i(x,\epsilon) = \sum_{n=0}^{\infty} R_i(n,\epsilon)x^n, \quad \mathcal{I}_i(x,\epsilon) = \sum_{n=0}^{\infty} I_i(n,\epsilon)x^n$$

- In order to solve our system, it has to verify several general assumptions, which are the following:

Assumptions

1. $\mathcal{M}(x,\epsilon) \in \mathcal{M}_n(\mathbb{K}(x,\epsilon))$ ($\mathbb{K}$ is a computable subfield of $\mathbb{R}$) is invertible
2. Each element of $\mathcal{R}(x,\epsilon) = (\mathcal{R}_1(x,\epsilon), \ldots, \mathcal{R}_n(x,\epsilon))$ and $\mathcal{I}(x,\epsilon) = (\mathcal{I}_1(x,\epsilon), \ldots, \mathcal{I}_n(x,\epsilon))$ can be expanded as a **power series** in x:

$$\forall i \in \{1,\ldots,n\} \quad \mathcal{R}_i(x,\epsilon) = \sum_{n=0}^{\infty} R_i(n,\epsilon)x^n, \quad \mathcal{I}_i(x,\epsilon) = \sum_{n=0}^{\infty} I_i(n,\epsilon)x^n$$

3. The coefficients of the Taylor expansion above can themselves be **Laurent expanded**:

$$R_i(n,\epsilon) = \sum_{k=l}^{\infty} R_{i,k}(n)\epsilon^k, \quad I_i(n,\epsilon) = \sum_{k=l}^{\infty} I_{i,k}(n)\epsilon^k \text{ with } l \in \mathbb{Z}.$$

Where the coefficients $R_i(n,\epsilon)$ and $I_i(n,\epsilon)$ are **indefinite nested sums over hypergeometric products**

The main steps behind the solving algorithm are the following ones:

Solving iteratively the system (1)

1. Uncouple the system using the `OreSys` package. In all generality, we obtain one or several HODE of the form

$$\alpha_{0,i}(x,\epsilon)\mathcal{I}_i(x,\epsilon) + \cdots + \alpha_{m,i}(x,\epsilon)\mathrm{d}_x^{m_i}\mathcal{I}_i(x,\epsilon) = \beta_i(x,\epsilon) \tag{2}$$

where $\alpha_{i,j}(x,\epsilon) \in \mathbb{K}[x,\epsilon]$ and $\beta_i(x,\epsilon) \in \langle\{\mathcal{R}_i(x,\epsilon), \mathrm{d}_x^j\mathcal{R}_i(x,\epsilon)\}\rangle_{\mathbb{K}(x,\epsilon)}$, as well as several linear algebraic relations (LFs) for the other Mis of the form

$$\mathcal{I}_j(x,\epsilon) \in \langle\{\mathrm{d}_x^i\mathcal{I}_i(x,\epsilon), \mathcal{R}_i(x,\epsilon), \mathrm{d}_x^j\mathcal{R}_i(x,\epsilon)\}\rangle_{\mathbb{K}(x,\epsilon)}$$

The main steps behind the solving algorithm are the following ones:

Solving iteratively the system (1)

1. Uncouple the system using the `OreSys` package. In all generality, we obtain one or several HODE of the form

$$\alpha_{0,i}(x,\epsilon)\mathcal{I}_i(x,\epsilon) + \cdots + \alpha_{m,i}(x,\epsilon)\mathrm{d}_x^{m_i}\mathcal{I}_i(x,\epsilon) = \beta_i(x,\epsilon) \tag{2}$$

where $\alpha_{i,j}(x,\epsilon) \in \mathbb{K}[x,\epsilon]$ and $\beta_i(x,\epsilon) \in \langle\{\mathcal{R}_i(x,\epsilon), \mathrm{d}_x^j\mathcal{R}_i(x,\epsilon)\}\rangle_{\mathbb{K}(x,\epsilon)}$, as well as several linear algebraic relations (LFs) for the other Mis of the form

$$\mathcal{I}_j(x,\epsilon) \in \langle\{\mathrm{d}_x^i\mathcal{I}_i(x,\epsilon), \mathcal{R}_i(x,\epsilon), \mathrm{d}_x^j\mathcal{R}_i(x,\epsilon)\}\rangle_{\mathbb{K}(x,\epsilon)}$$

2. Taylor expanding $\mathcal{I}_i(x,\epsilon)$, plugging them into the HODEs and comparing coefficients we get recurrences of the form

$$a_0(n,\epsilon)I_i(n,\epsilon) + a_1(n,\epsilon)I_i(n+1,\epsilon) + \cdots + a_d(n,\epsilon)I_i(n+d,\epsilon) = b(n,\epsilon) \tag{3}$$

Where $d \in \mathbb{N}$, $a_i(n,\epsilon) \in \mathbb{K}[n,\epsilon]$ and $b(n,\epsilon)$ can be expanded in Laurent series.

Solve iteratively the system (2)

3. We solve the difference equation (3) order by order in ϵ up to the required order, i.e. we search for a Laurent expansion for $I_1(n,\epsilon)$ of the form

$$I_1(n,\epsilon) = \sum_{k=-l}^{\infty} I_{1,k}(n)\epsilon^k$$

where $l \in \mathbb{N}$ and the $I_{1,k}(n)$ can be given in terms of INSH, and after collecting terms with the same power of ϵ, we solve the $x,\epsilon-$independent difference equations using `Sigma`.

Solve iteratively the system (2)

3. We solve the difference equation (3) order by order in ϵ up to the required order, i.e. we search for a Laurent expansion for $I_1(n,\epsilon)$ of the form

$$I_1(n,\epsilon) = \sum_{k=-l}^{\infty} I_{1,k}(n)\epsilon^k$$

where $l \in \mathbb{N}$ and the $I_{1,k}(n)$ can be given in terms of INSH, and after collecting terms with the same power of ϵ, we solve the $x,\epsilon-$independent difference equations using `Sigma`.

4. Once we get a solution for the $\tilde{\mathcal{I}}_i(x,\epsilon)$ up to some order ℓ in ϵ, we plug it in the second step's relations in order to get the other MIs up to the required order in ϵ.

Caveats:

- The original system can have a huge order ($n \sim \mathcal{O}(10^3)$) and we need a way to break it down to smaller subsystems

Caveats:

- The original system can have a huge order ($n \sim \mathcal{O}(10^3)$) and we need a way to break it down to smaller subsystems
- The uncouplings highly depend on the variable in which we uncouple, i.e. the one that appear in the HODE

Caveats:

- The original system can have a huge order ($n \sim \mathcal{O}(10^3)$) and we need a way to break it down to smaller subsystems
- The uncouplings highly depend on the variable in which we uncouple, i.e. the one that appear in the HODE
- There exists different uncoupling schemes, in particular the Zürcher scheme and the Gauß, that might yield different expressions and HODEs/LFs

Caveats:

- The original system can have a huge order ($n \sim \mathcal{O}(10^3)$) and we need a way to break it down to smaller subsystems
- The uncouplings highly depend on the variable in which we uncouple, i.e. the one that appear in the HODE
- There exists different uncoupling schemes, in particular the Zürcher scheme and the Gauß, that might yield different expressions and HODEs/LFs
- The BIs that appear in the RHS of the system must be given to the solving algorithm in order to solve the system: up to which order in ϵ?

Caveats:

- The original system can have a huge order ($n \sim \mathcal{O}(10^3)$) and we need a way to break it down to smaller subsystems
- The uncouplings highly depend on the variable in which we uncouple, i.e. the one that appear in the HODE
- There exists different uncoupling schemes, in particular the Zürcher scheme and the Gauß, that might yield different expressions and HODEs/LFs
- The BIs that appear in the RHS of the system must be given to the solving algorithm in order to solve the system: up to which order in ϵ?

Idea: Write a preprocessing algorithm that analyses the system, finds the best possible uncouplings and computes the required $\epsilon-$orders for the BIs to speed up the actual solving

Remarks:

1. Usually, the system that we initially get is quite huge (of order $n \sim 1000$)
2. We can find several sub-systems where the MIs depend only on themselves and on MIs from lower sub-systems

Remarks:

1. Usually, the system that we initially get is quite huge (of order $n \sim 1000$)
2. We can find several sub-systems where the MIs depend only on themselves and on MIs from lower sub-systems

Preprocessing step 1: Clustering

We "triangularize/cluster" the system in n smaller sub-systems where the ith subsystem of coupled MIs depend effectively only on these MIs and on those from the systems 1 through $i-1$.

Algorithmically, one can proceed in the following way to cluster one big system:

- Write the dependence relations of the LHS on the MIs appearing in the RHS
  Example:

$$\mathrm{d}_x \mathcal{I}_2(x,\epsilon) = 2\mathcal{I}_1(x,\epsilon) + \frac{\mathcal{I}_3(x,\epsilon)}{x} \;\; : \;\; \mathcal{I}_2(x,\epsilon) \to \{\mathcal{I}_1(x,\epsilon), \mathcal{I}_3(x,\epsilon)\}$$

Algorithmically, one can proceed in the following way to cluster one big system:

- Write the dependence relations of the LHS on the MIs appearing in the RHS
  Example:

$$\mathrm{d}_x \mathcal{I}_2(x,\epsilon) = 2\mathcal{I}_1(x,\epsilon) + \frac{\mathcal{I}_3(x,\epsilon)}{x} \;\; : \;\; \mathcal{I}_2(x,\epsilon) \to \{\mathcal{I}_1(x,\epsilon), \mathcal{I}_3(x,\epsilon)\}$$

- Find the full dependence relations by applying recursively the relations found above

Algorithmically, one can proceed in the following way to cluster one big system:

- Write the dependence relations of the LHS on the MIs appearing in the RHS
  Example:

$$\mathrm{d}_x \mathcal{I}_2(x,\epsilon) = 2\mathcal{I}_1(x,\epsilon) + \frac{\mathcal{I}_3(x,\epsilon)}{x} \;\; : \;\; \mathcal{I}_2(x,\epsilon) \to \{\mathcal{I}_1(x,\epsilon), \mathcal{I}_3(x,\epsilon)\}$$

- Find the full dependence relations by applying recursively the relations found above
- Cluster together the differential equations that have the same full dependence

Algorithmically, one can proceed in the following way to cluster one big system:

- Write the dependence relations of the LHS on the MIs appearing in the RHS
  Example:

$$\mathrm{d}_x \mathcal{I}_2(x,\epsilon) = 2\mathcal{I}_1(x,\epsilon) + \frac{\mathcal{I}_3(x,\epsilon)}{x} \ : \ \mathcal{I}_2(x,\epsilon) \to \{\mathcal{I}_1(x,\epsilon), \mathcal{I}_3(x,\epsilon)\}$$

- Find the full dependence relations by applying recursively the relations found above
- Cluster together the differential equations that have the same full dependence

Explicit example:

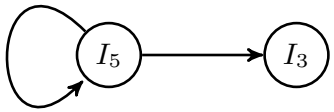
$$\begin{cases} \frac{1}{x^2-1} I_5'(x) &= \frac{x(x+2)}{x-2} I_5(x) + x I_3(x) + R_5(x) \\ (x+1) I_4'(x) &= -I_3(x) + \frac{I_1(x)}{x} + R_4(x) \\ \frac{x(x-2)}{3} I_3'(x) &= x I_4(x) + R_3(x) \\ I_2'(x) &= x^3 I_1(x) + R_2(x) \\ 2 I_1'(x) &= x I_1(x) - \frac{8}{x-1} I_2(x) + R_1(x) \end{cases}$$

- We write first all the fundamental rules:

$$I_5 \to \{I_3, I_5\},\ I_4 \to \{I_1, I_3\},\ I_3 \to \{I_4\},\ I_2 \to \{I_1\},\ I_1 \to \{I_1, I_2\}$$

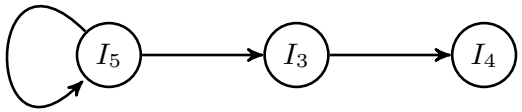
- We write first all the fundamental rules:

$$I_5 \to \{I_3, I_5\},\ I_4 \to \{I_1, I_3\},\ I_3 \to \{I_4\},\ I_2 \to \{I_1\},\ I_1 \to \{I_1, I_2\}$$



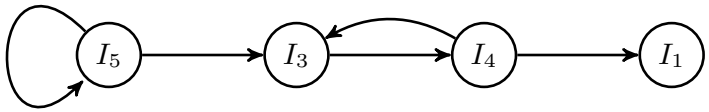
- We write first all the fundamental rules:

$$I_5 \to \{I_3, I_5\},\ I_4 \to \{I_1, I_3\},\ I_3 \to \{I_4\},\ I_2 \to \{I_1\},\ I_1 \to \{I_1, I_2\}$$



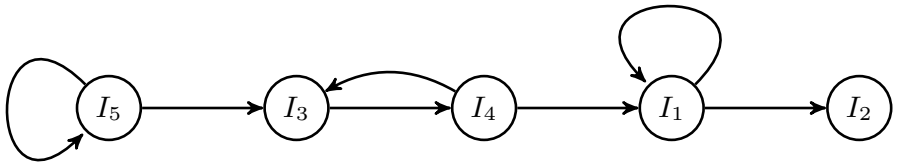
- We write first all the fundamental rules:

$$I_5 \to \{I_3, I_5\},\ I_4 \to \{I_1, I_3\},\ I_3 \to \{I_4\},\ I_2 \to \{I_1\},\ I_1 \to \{I_1, I_2\}$$



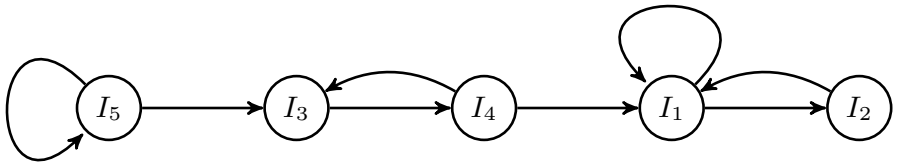
- We write first all the fundamental rules:

$$I_5 \to \{I_3, I_5\},\ I_4 \to \{I_1, I_3\},\ I_3 \to \{I_4\},\ I_2 \to \{I_1\},\ I_1 \to \{I_1, I_2\}$$



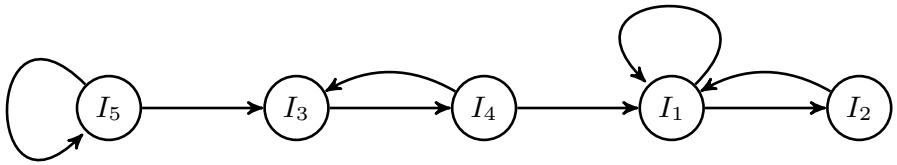
- We write first all the fundamental rules:

$$I_5 \to \{I_3, I_5\},\ I_4 \to \{I_1, I_3\},\ I_3 \to \{I_4\},\ I_2 \to \{I_1\},\ I_1 \to \{I_1, I_2\}$$



- We write first all the fundamental rules:

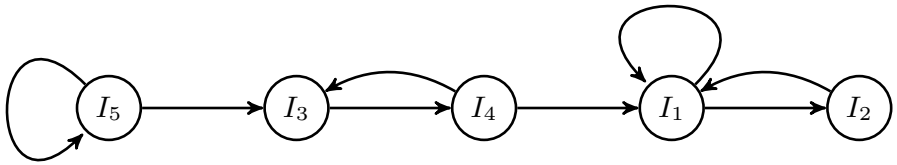
$$I_5 \to \{I_3, I_5\},\ I_4 \to \{I_1, I_3\},\ I_3 \to \{I_4\},\ I_2 \to \{I_1\},\ I_1 \to \{I_1, I_2\}$$



$$I_5 \to \{I_1, I_2, I_3, I_4, I_5\}$$

- We write first all the fundamental rules:

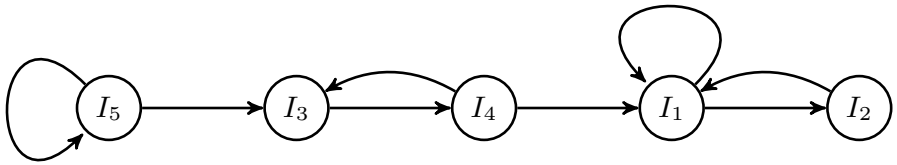
$$I_5 \to \{I_3, I_5\},\ I_4 \to \{I_1, I_3\},\ I_3 \to \{I_4\},\ I_2 \to \{I_1\},\ I_1 \to \{I_1, I_2\}$$



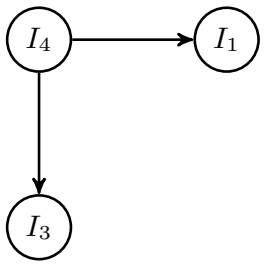
$$I_5 \to \{I_1, I_2, I_3, I_4, I_5\}$$

- We write first all the fundamental rules:

$$I_5 \to \{I_3, I_5\},\ I_4 \to \{I_1, I_3\},\ I_3 \to \{I_4\},\ I_2 \to \{I_1\},\ I_1 \to \{I_1, I_2\}$$

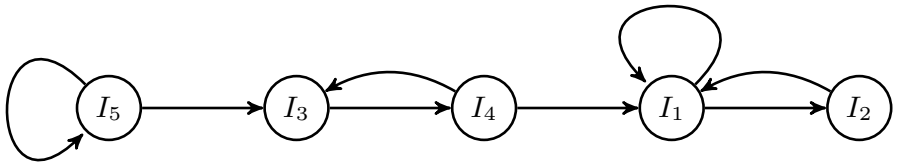


$$I_5 \to \{I_1, I_2, I_3, I_4, I_5\}$$

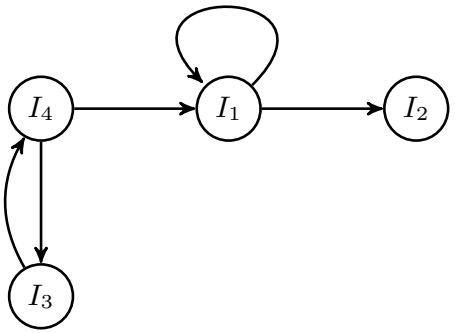


- We write first all the fundamental rules:

$$I_5 \to \{I_3, I_5\},\ I_4 \to \{I_1, I_3\},\ I_3 \to \{I_4\},\ I_2 \to \{I_1\},\ I_1 \to \{I_1, I_2\}$$

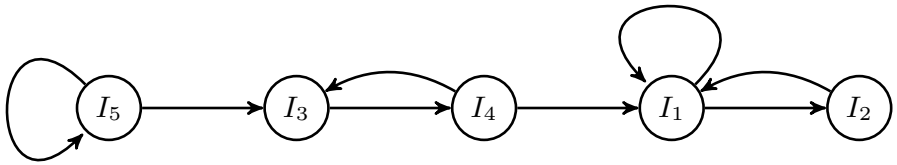


$$I_5 \to \{I_1, I_2, I_3, I_4, I_5\}$$

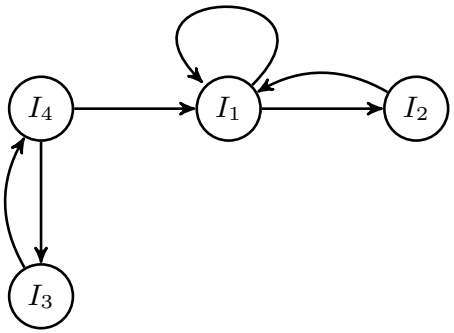


- We write first all the fundamental rules:

$$I_5 \to \{I_3, I_5\},\ I_4 \to \{I_1, I_3\},\ I_3 \to \{I_4\},\ I_2 \to \{I_1\},\ I_1 \to \{I_1, I_2\}$$

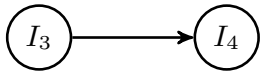


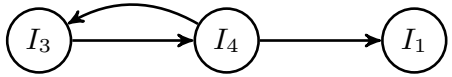
$$I_5 \to \{I_1, I_2, I_3, I_4, I_5\}$$

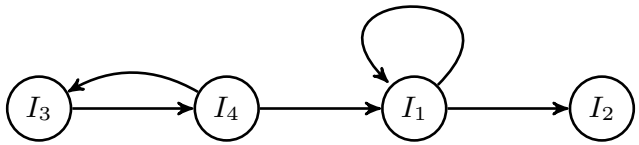


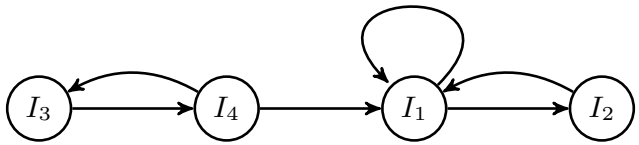
$$I_4 \to \{I_1, I_2, I_3, I_4\}$$


I_3

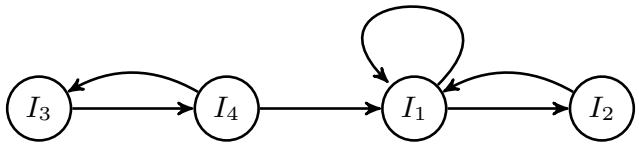

I_3
I_4


I_3
I_4
I_1

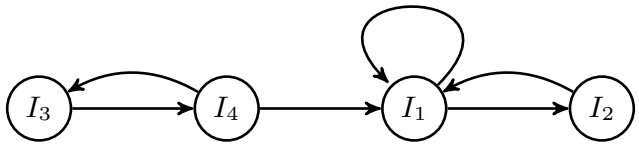

I_3
I_4
I_1
I_2



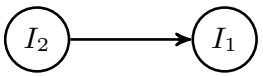
$$I_3 \to \{I_1, I_2, I_3, I_4\}$$

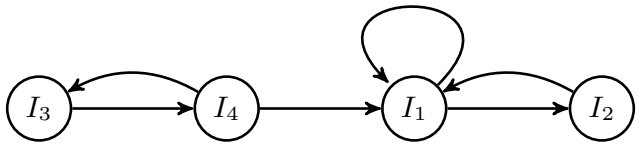


$$I_3 \to \{I_1, I_2, I_3, I_4\}$$

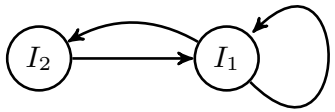


$$I_3 \to \{I_1, I_2, I_3, I_4\}$$

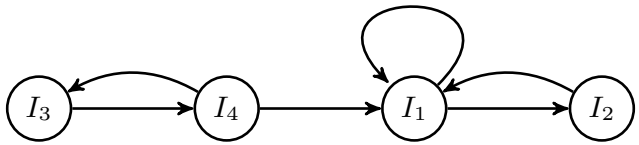




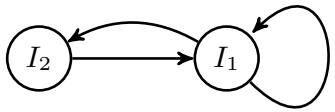
$$I_3 \to \{I_1, I_2, I_3, I_4\}$$



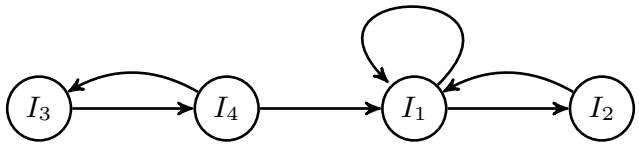
$$I_2 \to \{I_1, I_2\}$$



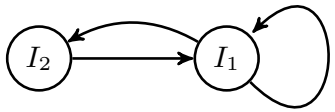
$$I_3 \to \{I_1, I_2, I_3, I_4\}$$



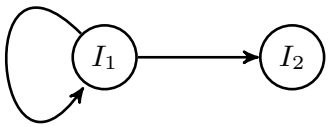
$$I_2 \to \{I_1, I_2\}$$

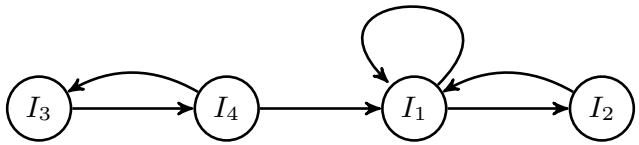


$$I_3 \to \{I_1, I_2, I_3, I_4\}$$

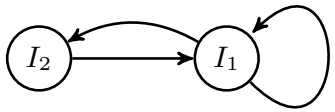


$$I_2 \to \{I_1, I_2\}$$

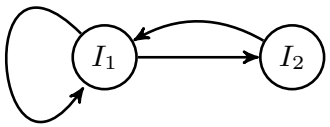




$$I_3 \to \{I_1, I_2, I_3, I_4\}$$



$$I_2 \to \{I_1, I_2\}$$



$$I_1 \to \{I_1, I_2\}$$

The full rules are therefore the following:

$$\begin{cases} I_5 & \rightarrow & \{I_1, I_2, I_3, I_4, I_5\} \\ I_4 & \rightarrow & \{I_1, I_2, I_3, I_4\} \\ I_3 & \rightarrow & \{I_1, I_2, I_3, I_4\} \\ I_2 & \rightarrow & \{I_1, I_2\} \\ I_1 & \rightarrow & \{I_1, I_2\} \end{cases}$$

That is to say, the system can be clustered in 3 subsystems:

$$\{I_5\}, \quad \{I_3, I_4\}, \quad \{I_1, I_2\}$$

The full rules are therefore the following:

$$\begin{cases} I_5 & \to & \{I_1, I_2, I_3, I_4, I_5\} \\ I_4 & \to & \{I_1, I_2, I_3, I_4\} \\ I_3 & \to & \{I_1, I_2, I_3, I_4\} \\ I_2 & \to & \{I_1, I_2\} \\ I_1 & \to & \{I_1, I_2\} \end{cases}$$

That is to say, the system can be clustered in 3 subsystems:

$$\{I_5\}, \quad \{I_3, I_4\}, \quad \{I_1, I_2\}$$

Notes:

- This algorithm always terminates in finite time: at each step of the recursion we add at least one new MI (and there are n of them) or we stop

The full rules are therefore the following:

$$\begin{cases} I_5 & \to & \{I_1, I_2, I_3, I_4, I_5\} \\ I_4 & \to & \{I_1, I_2, I_3, I_4\} \\ I_3 & \to & \{I_1, I_2, I_3, I_4\} \\ I_2 & \to & \{I_1, I_2\} \\ I_1 & \to & \{I_1, I_2\} \end{cases}$$

That is to say, the system can be clustered in 3 subsystems:

$$\{I_5\}, \quad \{I_3, I_4\}, \quad \{I_1, I_2\}$$

Notes:

- This algorithm always terminates in finite time: at each step of the recursion we add at least one new MI (and there are n of them) or we stop
- If some differential equation (DE) is of the trivial form:

$$\alpha_i(x, \epsilon) \mathrm{d}_x \mathcal{I}_i(x, \epsilon) = 0$$

then $\mathcal{I}_i(x, \epsilon)$ is actually a trivial MI that can be moved to the base case integrals (that by assumption can be computed easily)

The full rules are therefore the following:

$$\begin{cases} I_5 & \to & \{I_1, I_2, I_3, I_4, I_5\} \\ I_4 & \to & \{I_1, I_2, I_3, I_4\} \\ I_3 & \to & \{I_1, I_2, I_3, I_4\} \\ I_2 & \to & \{I_1, I_2\} \\ I_1 & \to & \{I_1, I_2\} \end{cases}$$

That is to say, the system can be clustered in 3 subsystems:

$$\{I_5\}, \quad \{I_3, I_4\}, \quad \{I_1, I_2\}$$

Notes:

- This algorithm always terminates in finite time: at each step of the recursion we add at least one new MI (and there are n of them) or we stop
- If some differential equation (DE) is of the trivial form:

$$\alpha_i(x, \epsilon) \mathrm{d}_x \mathcal{I}_i(x, \epsilon) = 0$$

  then $\mathcal{I}_i(x, \epsilon)$ is actually a trivial MI that can be moved to the base case integrals (that by assumption can be computed easily)
- A MI that appears as a BI in a subsystem is called an **induced BI** (IBI), as opposed to the real BI, that we call **fundamental BI** (FBI)

From now on, we consider a subsystem of the original system, and to distinguish them we will denote all the quantities with a tilde:

$$\frac{\mathrm{d}}{\mathrm{d}x}\tilde{\mathcal{I}}(x,\epsilon) = \tilde{\mathcal{M}}(x,\epsilon)\tilde{\mathcal{I}}(x,\epsilon) + \tilde{\mathcal{R}}(x,\epsilon) \quad \text{(of order } n\text{)}$$

From now on, we consider a subsystem of the original system, and to distinguish them we will denote all the quantities with a tilde:

$$\frac{\mathrm{d}}{\mathrm{d}x}\tilde{\mathcal{I}}(x,\epsilon) = \tilde{\mathcal{M}}(x,\epsilon)\tilde{\mathcal{I}}(x,\epsilon) + \tilde{\mathcal{R}}(x,\epsilon) \quad \text{(of order } n\text{)}$$

Task: Find the best possible uncoupling.

- We use for the uncoupling the `OreSys` package by Stefan Gerhold

From now on, we consider a subsystem of the original system, and to distinguish them we will denote all the quantities with a tilde:

$$\frac{\mathrm{d}}{\mathrm{d}x}\tilde{\mathcal{I}}(x,\epsilon) = \tilde{\mathcal{M}}(x,\epsilon)\tilde{\mathcal{I}}(x,\epsilon) + \tilde{\mathcal{R}}(x,\epsilon) \quad \text{(of order } n\text{)}$$

Task: Find the best possible uncoupling.

- We use for the uncoupling the `OreSys` package by Stefan Gerhold
- We use as the starting point for our algorithm the Zürcher uncoupling scheme, which provides (almost) always the following simple structure in the uncoupling:

From now on, we consider a subsystem of the original system, and to distinguish them we will denote all the quantities with a tilde:

$$\frac{\mathrm{d}}{\mathrm{d}x}\tilde{\mathcal{I}}(x,\epsilon) = \tilde{\mathcal{M}}(x,\epsilon)\tilde{\mathcal{I}}(x,\epsilon) + \tilde{\mathcal{R}}(x,\epsilon) \quad \text{(of order } n\text{)}$$

Task: Find the best possible uncoupling.

- We use for the uncoupling the `OreSys` package by Stefan Gerhold
- We use as the starting point for our algorithm the Zürcher uncoupling scheme, which provides (almost) always the following simple structure in the uncoupling:
  - One HODE of order n in the first variable of the system

From now on, we consider a subsystem of the original system, and to distinguish them we will denote all the quantities with a tilde:

$$\frac{\mathrm{d}}{\mathrm{d}x}\tilde{\mathcal{I}}(x,\epsilon) = \tilde{\mathcal{M}}(x,\epsilon)\tilde{\mathcal{I}}(x,\epsilon) + \tilde{\mathcal{R}}(x,\epsilon) \quad \text{(of order } n\text{)}$$

Task: Find the best possible uncoupling.

- We use for the uncoupling the `OreSys` package by Stefan Gerhold
- We use as the starting point for our algorithm the Zürcher uncoupling scheme, which provides (almost) always the following simple structure in the uncoupling:
  - One HODE of order n in the first variable of the system
  - $n-1$ LFs expressing the other MIs as a function of the HODE variable as well as its derivatives

From now on, we consider a subsystem of the original system, and to distinguish them we will denote all the quantities with a tilde:

$$\frac{\mathrm{d}}{\mathrm{d}x}\tilde{\mathcal{I}}(x,\epsilon) = \tilde{\mathcal{M}}(x,\epsilon)\tilde{\mathcal{I}}(x,\epsilon) + \tilde{\mathcal{R}}(x,\epsilon) \quad \text{(of order } n\text{)}$$

Task: Find the best possible uncoupling.

- We use for the uncoupling the `OreSys` package by Stefan Gerhold
- We use as the starting point for our algorithm the Zürcher uncoupling scheme, which provides (almost) always the following simple structure in the uncoupling:
  - One HODE of order n in the first variable of the system
  - $n-1$ LFs expressing the other MIs as a function of the HODE variable as well as its derivatives
- All of these equations will depend **explicitely** on the $\tilde{\mathcal{R}}_i(x,\epsilon)$ variables and their derivatives (that we keep generic to speed up the uncoupling)

- More precisely, the `OreSys` Zürcher inital uncoupling actually makes use of dummy variables (here denoted with W_i) and we have to get rid of them:

Zürcher uncoupling (OreSys output)

- $\mathrm{HODE}\left(\underline{W_1, \{\mathrm{d}_x^j W_1\}}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}\right)$
- $$\begin{cases} W_2 &= F_2(W_1, \{\mathrm{d}_x^j W_1\}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}) \\ W_3 &= F_3(W_1, \{\mathrm{d}_x^j W_1\}; W_2, \{\mathrm{d}_x^j W_2\}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}) \\ \vdots & \qquad\qquad \vdots \\ W_n &= F_n(W_1, \{\mathrm{d}_x^j W_1\}; \cdots; W_{n-1}, \{\mathrm{d}_x^j W_{n-1}\}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}) \end{cases}$$
- $$\begin{cases} \tilde{\mathcal{I}}_1 &= G_1(\{W_i, \mathrm{d}_x^j W_i\}) \ (= W_1) \\ \vdots & \qquad\qquad \vdots \\ \tilde{\mathcal{I}}_n &= G_n(\{W_i, \mathrm{d}_x^j W_i\}) \end{cases}$$

- More precisely, the `OreSys` Zürcher inital uncoupling actually makes use of dummy variables (here denoted with W_i) and we have to get rid of them:

Zürcher uncoupling (OreSys output)

- $\mathrm{HODE}\left(\underline{W_1, \{\mathrm{d}_x^j W_1\}}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}\right)$
- $$\begin{cases} W_2 &= F_2(W_1, \{\mathrm{d}_x^j W_1\}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}) \\ W_3 &= F_3(W_1, \{\mathrm{d}_x^j W_1\}; W_2, \{\mathrm{d}_x^j W_2\}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}) \\ \vdots & \qquad\qquad \vdots \\ W_n &= F_n(W_1, \{\mathrm{d}_x^j W_1\}; \cdots; W_{n-1}, \{\mathrm{d}_x^j W_{n-1}\}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}) \end{cases}$$
- $$\begin{cases} \tilde{\mathcal{I}}_1 &= G_1(\{W_i, \mathrm{d}_x^j W_i\}) \ (= W_1) \\ \vdots & \qquad\qquad \vdots \\ \tilde{\mathcal{I}}_n &= G_n(\{W_i, \mathrm{d}_x^j W_i\}) \end{cases}$$

Notes:

- Eliminating the W_i dummy variables and replacing the generic $\tilde{\mathcal{R}}_i$ by their actual values is sometimes as costly as the uncoupling itself
- Non-trivial case when $G_1(\{W_i, \mathrm{d}_x^j W_i\}) \neq W_1$

The very first step is to convert the `OreSys` output into the actual uncoupling with the full dependence on the BIs, that we note from now on $\tilde{\mathcal{J}}_1(x,\epsilon),\ldots,\tilde{\mathcal{J}}_m(x,\epsilon)$:

Zürcher uncoupling (with full dependence)

- $$\sum_{k=0}^{m} p_k(x,\epsilon)\mathrm{d}_x^k\tilde{\mathcal{I}}_1(x,\epsilon) = r(x,\epsilon)$$ HODE in $\tilde{\mathcal{I}}_1(x)$, $p_i(x,\epsilon)\in\mathbb{K}(x,\epsilon)$
- $$r(x,\epsilon) = \sum_{i=0}^{p}\sum_{j=1}^{m} r_{i,j}(x,\epsilon)\mathrm{d}_x^i\tilde{\mathcal{J}}_j(x,\epsilon)$$ $p\in\mathbb{N}$, $r_{i,j}(x,\epsilon)\in\mathbb{K}(x,\epsilon)$
- $$\tilde{\mathcal{I}}_j(x,\epsilon) = \sum_{i=0}^{m-1} a_{j,i}(x,\epsilon)\mathrm{d}_x^i\tilde{\mathcal{I}}_1(x,\epsilon) + \rho_k(x,\epsilon)$$ $j\in\{2,\ldots,m\}$, $a_{k,i}(x,\epsilon)\in\mathbb{K}(x,\epsilon)$, $\rho_j(x)$ like $r(x,\epsilon)$

Question: What does it mean that one uncoupling is "better" than another? In other words, what's the comparison point between different uncouplings?

Question: What does it mean that one uncoupling is "better" than another? In other words, what's the comparison point between different uncouplings?

- Suppose we have a system in three MIs $\tilde{\mathcal{I}}_{i=1,2,3}(x,\epsilon)$ with one BI $\tilde{\mathcal{J}}(x,\epsilon)$, that we uncoupled in $\tilde{\mathcal{I}}_1$ so that we get the following expression for $\tilde{\mathcal{I}}_2$:

$$\tilde{\mathcal{I}}_2(x,\epsilon) = F(\{\tilde{\mathcal{I}}_1, \mathrm{d}_x^i \tilde{\mathcal{I}}_1\}) + \frac{1}{\epsilon^2}\tilde{\rho}_{2,1}(x,\epsilon)\tilde{\mathcal{J}}(x,\epsilon) + \epsilon\tilde{\rho}_{2,2}(x,\epsilon)\mathrm{d}_x\tilde{\mathcal{J}}(x,\epsilon)$$

Where $\mathbb{K}(x,\epsilon) \ni \tilde{\rho}_{2,i\in\{1,2\}}(x,\epsilon) \underset{\epsilon\to 0}{\longrightarrow} \tilde{\rho}_{2,i\in\{1,2\}}(x) \in \mathbb{K}(x)^\star$

Question: What does it mean that one uncoupling is "better" than another? In other words, what's the comparison point between different uncouplings?

- Suppose we have a system in three MIs $\tilde{\mathcal{I}}_{i=1,2,3}(x,\epsilon)$ with one BI $\tilde{\mathcal{J}}(x,\epsilon)$, that we uncoupled in $\tilde{\mathcal{I}}_1$ so that we get the following expression for $\tilde{\mathcal{I}}_2$:

$$\tilde{\mathcal{I}}_2(x,\epsilon) = F(\{\tilde{\mathcal{I}}_1, \mathrm{d}_x^i \tilde{\mathcal{I}}_1\}) + \frac{1}{\epsilon^2}\tilde{\rho}_{2,1}(x,\epsilon)\tilde{\mathcal{J}}(x,\epsilon) + \epsilon\tilde{\rho}_{2,2}(x,\epsilon)\mathrm{d}_x\tilde{\mathcal{J}}(x,\epsilon)$$

  Where $\mathbb{K}(x,\epsilon) \ni \tilde{\rho}_{2,i\in\{1,2\}}(x,\epsilon) \underset{\epsilon\to 0}{\longrightarrow} \tilde{\rho}_{2,i\in\{1,2\}}(x) \in \mathbb{K}(x)^\star$

- This means that if we want for instance to solve the system up to the 1^{st} order in ϵ, we need to give an expansion of $\tilde{\mathcal{J}}(x,\epsilon) = \sum_{n=l}^{\infty} \tilde{\mathcal{J}}_n(x)\epsilon^n$ up to the 3^{rd} order in ϵ:

$$\begin{aligned}\tilde{\mathcal{I}}_2(x,\epsilon) \sim \tfrac{1}{\epsilon^2}\tilde{\mathcal{J}}(x,\epsilon) &= \tfrac{1}{\epsilon^2}\left(\cdots + \tilde{\mathcal{J}}_0(x) + \epsilon\tilde{\mathcal{J}}_1(x) + \epsilon^2\tilde{\mathcal{J}}_2(x) + \epsilon^3\tilde{\mathcal{J}}_3(x) + \cdots\right)\\ &= \cdots + \tfrac{1}{\epsilon^2}\tilde{\mathcal{J}}_0(x) + \tfrac{1}{\epsilon}\tilde{\mathcal{J}}_1(x) + \tilde{\mathcal{J}}_2(x) + \boxed{\epsilon\tilde{\mathcal{J}}_3(x)} + \cdots\end{aligned}$$

Question: What does it mean that one uncoupling is "better" than another? In other words, what's the comparison point between different uncouplings?

- Suppose we have a system in three MIs $\tilde{\mathcal{I}}_{i=1,2,3}(x,\epsilon)$ with one BI $\tilde{\mathcal{J}}(x,\epsilon)$, that we uncoupled in $\tilde{\mathcal{I}}_1$ so that we get the following expression for $\tilde{\mathcal{I}}_2$:

$$\tilde{\mathcal{I}}_2(x,\epsilon) = F(\{\tilde{\mathcal{I}}_1, \mathrm{d}_x^i \tilde{\mathcal{I}}_1\}) + \frac{1}{\epsilon^2}\tilde{\rho}_{2,1}(x,\epsilon)\tilde{\mathcal{J}}(x,\epsilon) + \epsilon\tilde{\rho}_{2,2}(x,\epsilon)\mathrm{d}_x\tilde{\mathcal{J}}(x,\epsilon)$$

  Where $\mathbb{K}(x,\epsilon) \ni \tilde{\rho}_{2,i\in\{1,2\}}(x,\epsilon) \underset{\epsilon\to 0}{\longrightarrow} \tilde{\rho}_{2,i\in\{1,2\}}(x) \in \mathbb{K}(x)^\star$

- This means that if we want for instance to solve the system up to the 1^{st} order in ϵ, we need to give an expansion of $\tilde{\mathcal{J}}(x,\epsilon) = \sum_{n=l}^{\infty} \tilde{\mathcal{J}}_n(x)\epsilon^n$ up to the 3^{rd} order in ϵ:

$$\begin{aligned}\tilde{\mathcal{I}}_2(x,\epsilon) \sim \tfrac{1}{\epsilon^2}\tilde{\mathcal{J}}(x,\epsilon) &= \tfrac{1}{\epsilon^2}\left(\cdots + \tilde{\mathcal{J}}_0(x) + \epsilon\tilde{\mathcal{J}}_1(x) + \epsilon^2\tilde{\mathcal{J}}_2(x) + \epsilon^3\tilde{\mathcal{J}}_3(x) + \cdots\right)\\ &= \cdots + \tfrac{1}{\epsilon^2}\tilde{\mathcal{J}}_0(x) + \tfrac{1}{\epsilon}\tilde{\mathcal{J}}_1(x) + \tilde{\mathcal{J}}_2(x) + \boxed{\epsilon\tilde{\mathcal{J}}_3(x)} + \cdots\end{aligned}$$

Answer: To compare two uncouplings, we compare the $\epsilon-$orders for the BIs appearing in the inhomogeneous part of the HODE and in the LFs: **the less the expansion orders are, the better is the uncoupling**

Question: What uncouplings are we comparing?

Question: What uncouplings are we comparing?

- Simple version: permute cyclically the MIs vector $\tilde{\mathcal{I}}(x,\epsilon) = \left(\tilde{\mathcal{I}}_1(x,\epsilon),\ldots,\tilde{\mathcal{I}}_n(x,\epsilon)\right)$ with the permutation $\sigma = (1\cdots n)$, so that we uncouple first in $\tilde{\mathcal{I}}_1$, then in $\tilde{\mathcal{I}}_2$, etc. until $\tilde{\mathcal{I}}_n$.
  - We compare the n resulting uncouplings through $\epsilon-$orders comparison.
  - Minimizes the number of HODEs (only one per uncoupling), but not optimal

Question: What uncouplings are we comparing?

- Simple version: permute cyclically the MIs vector $\tilde{\mathcal{I}}(x,\epsilon) = \left(\tilde{\mathcal{I}}_1(x,\epsilon),\ldots,\tilde{\mathcal{I}}_n(x,\epsilon)\right)$ with the permutation $\sigma = (1\cdots n)$, so that we uncouple first in $\tilde{\mathcal{I}}_1$, then in $\tilde{\mathcal{I}}_2$, etc. until $\tilde{\mathcal{I}}_n$.
  - We compare the n resulting uncouplings through $\epsilon-$orders comparison.
  - Minimizes the number of HODEs (only one per uncoupling), but not optimal
- Advanced version: uncouple cyclically as above, but choose in a recursive way the best equations (HODEs or LFs) at each step for each MI among the n^2 available equations
  - Increases potentially the number of **initial conditions** (ICs) required, but improves the $\epsilon-$orders

Example:

- We have a system of size $n = 3$, with MIs $\tilde{\mathcal{I}}_{i=1,2,3}(x, \epsilon)$ and 4 BIs $\tilde{\mathcal{J}}_{j=1,\ldots,4}(x, \epsilon)$
- From now on, a couple $(\tilde{\mathcal{J}}_i, m_i)$ means that in the HODE/one of the LFs we encounter a factor of the form

$$\epsilon^{m_i} \frac{\alpha_i(x, \epsilon)}{\beta_i(x, \epsilon)} \tilde{\mathcal{J}}_i(x, \epsilon) \text{ with } \alpha_i(x, \epsilon), \beta_i(x, \epsilon) \in \mathbb{K}[x, \epsilon] \text{ and } \epsilon \nmid \alpha_i(x, \epsilon), \beta_i(x, \epsilon)$$

Example:

- We have a system of size $n = 3$, with MIs $\tilde{\mathcal{I}}_{i=1,2,3}(x, \epsilon)$ and 4 BIs $\tilde{\mathcal{J}}_{j=1,\ldots,4}(x, \epsilon)$
- From now on, a couple $(\tilde{\mathcal{J}}_i, m_i)$ means that in the HODE/one of the LFs we encounter a factor of the form

$$\epsilon^{m_i} \frac{\alpha_i(x,\epsilon)}{\beta_i(x,\epsilon)} \tilde{\mathcal{J}}_i(x,\epsilon) \text{ with } \alpha_i(x,\epsilon), \beta_i(x,\epsilon) \in \mathbb{K}[x,\epsilon] \text{ and } \epsilon \nmid \alpha_i(x,\epsilon), \beta_i(x,\epsilon)$$

MIs	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\text{HODE}(\tilde{\mathcal{I}}_1) = \tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$\tilde{\mathcal{I}}_2 = f_2(\tilde{\mathcal{I}}_1) = \tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_3 = f_3(\tilde{\mathcal{I}}_1) = \tilde{\mathcal{I}}_3^{(1)}$	-1	0	-1	-1
$\text{HODE}(\tilde{\mathcal{I}}_2) = \tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$\tilde{\mathcal{I}}_1 = f_1(\tilde{\mathcal{I}}_2) = \tilde{\mathcal{I}}_1^{(2)}$	-2	-1	0	-2
$\tilde{\mathcal{I}}_3 = f_3(\tilde{\mathcal{I}}_2) = \tilde{\mathcal{I}}_3^{(2)}$	0	0	0	0
$\text{HODE}(\tilde{\mathcal{I}}_3) = \tilde{\mathcal{I}}_3^{(H)}$	0	1	0	0
$\tilde{\mathcal{I}}_1 = f_1(\tilde{\mathcal{I}}_3) = \tilde{\mathcal{I}}_1^{(3)}$	0	0	0	0
$\tilde{\mathcal{I}}_2 = f_2(\tilde{\mathcal{I}}_3) = \tilde{\mathcal{I}}_2^{(3)}$	0	1	0	0

- First we select the best HODE between the three equations $\tilde{\mathcal{I}}_1^{(H)}$, $\tilde{\mathcal{I}}_2^{(H)}$ and $\tilde{\mathcal{I}}_3^{(H)}$:

	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$\tilde{\mathcal{I}}_3^{(H)}$	0	1	0	0
$n_1^{(H)}$				
$n_2^{(H)}$				
$n_3^{(H)}$				

- First we select the best HODE between the three equations $\tilde{\mathcal{I}}_1^{(H)}$, $\tilde{\mathcal{I}}_2^{(H)}$ and $\tilde{\mathcal{I}}_3^{(H)}$:

	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$\tilde{\mathcal{I}}_3^{(H)}$	0	1	0	0
$n_1^{(H)}$	1			
$n_2^{(H)}$	0			
$n_3^{(H)}$	1			

- First we select the best HODE between the three equations $\tilde{\mathcal{I}}_1^{(H)}$, $\tilde{\mathcal{I}}_2^{(H)}$ and $\tilde{\mathcal{I}}_3^{(H)}$:

	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$\tilde{\mathcal{I}}_3^{(H)}$	0	1	0	0
$n_1^{(H)}$	1	2		
$n_2^{(H)}$	0	0		
$n_3^{(H)}$	1	2		

- First we select the best HODE between the three equations $\tilde{\mathcal{I}}_1^{(H)}$, $\tilde{\mathcal{I}}_2^{(H)}$ and $\tilde{\mathcal{I}}_3^{(H)}$:

	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$\tilde{\mathcal{I}}_3^{(H)}$	0	1	0	0
$n_1^{(H)}$	1	2	2	
$n_2^{(H)}$	0	0	1	
$n_3^{(H)}$	1	2	2	

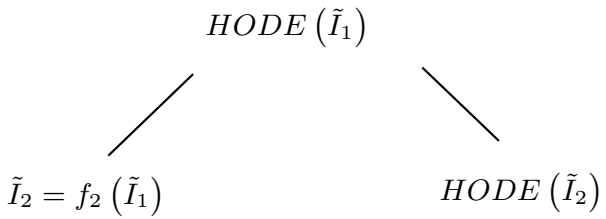
- First we select the best HODE between the three equations $\tilde{\mathcal{I}}_1^{(H)}$, $\tilde{\mathcal{I}}_2^{(H)}$ and $\tilde{\mathcal{I}}_3^{(H)}$:

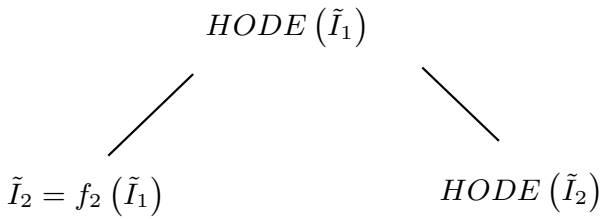
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$\tilde{\mathcal{I}}_3^{(H)}$	0	1	0	0
$n_1^{(H)}$	1	2	2	(3)
$n_2^{(H)}$	0	0	1	1
$n_3^{(H)}$	1	2	2	(3)

- First we select the best HODE between the three equations $\tilde{\mathcal{I}}_1^{(H)}$, $\tilde{\mathcal{I}}_2^{(H)}$ and $\tilde{\mathcal{I}}_3^{(H)}$:

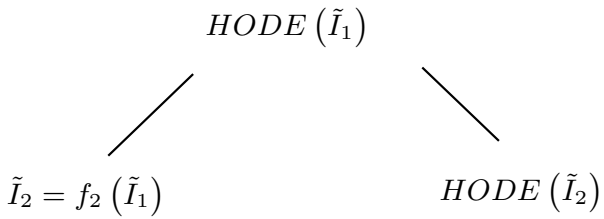
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$\tilde{\mathcal{I}}_3^{(H)}$	0	1	0	0
$n_1^{(H)}$	1	2	2	(3)
$n_2^{(H)}$	0	0	1	1
$n_3^{(H)}$	1	2	2	(3)

This means that there are 2 possibilities: $\tilde{\mathcal{I}}_1^{(H)}$ and $\tilde{\mathcal{I}}_3^{(H)}$. We'll have to examine both of them.

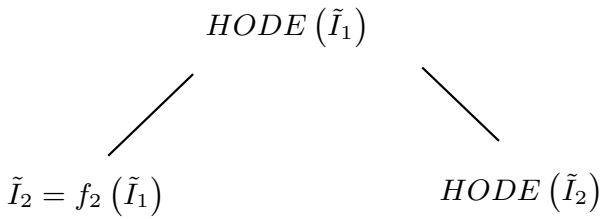

$HODE\left(\tilde{I}_1\right)$
$\tilde{I}_2 = f_2\left(\tilde{I}_1\right)$
$HODE\left(\tilde{I}_2\right)$



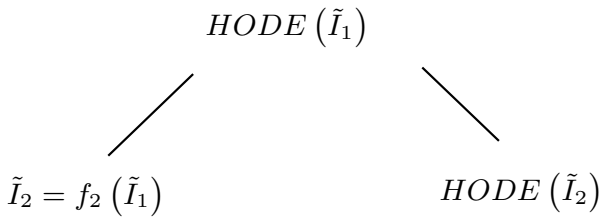
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$				
$n_2^{(H)}$				



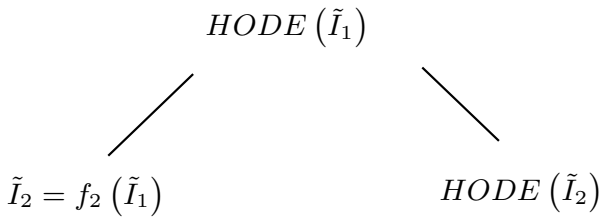
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1			
$n_2^{(H)}$	0			



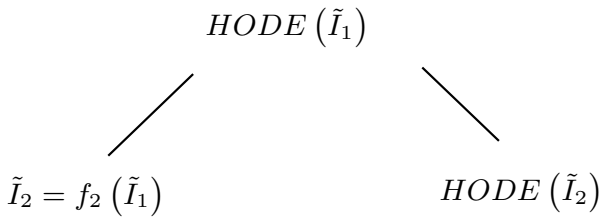
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1	2		
$n_2^{(H)}$	0	1		



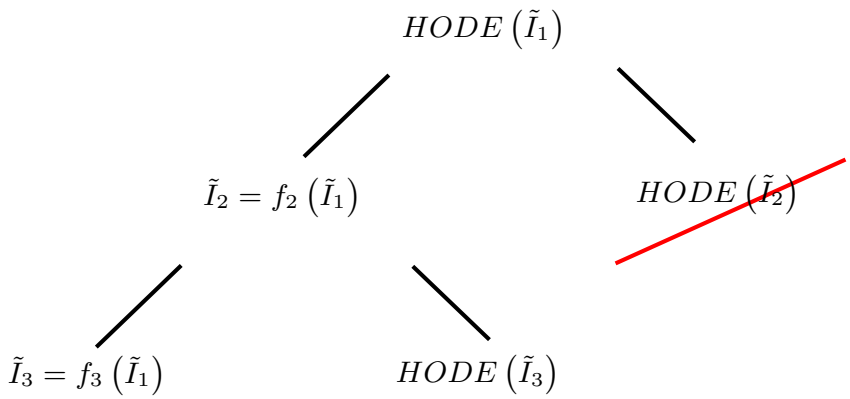
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1	2	2	
$n_2^{(H)}$	0	1	2	

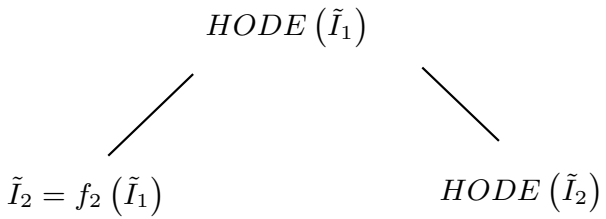


	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1	2	2	(3)
$n_2^{(H)}$	0	1	2	2

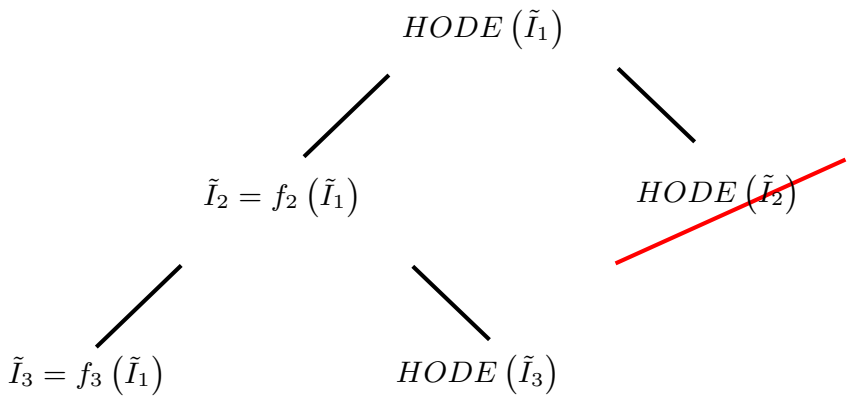


	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1	2	2	(3)
$n_2^{(H)}$	0	1	2	2

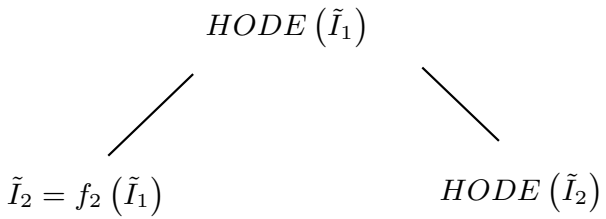




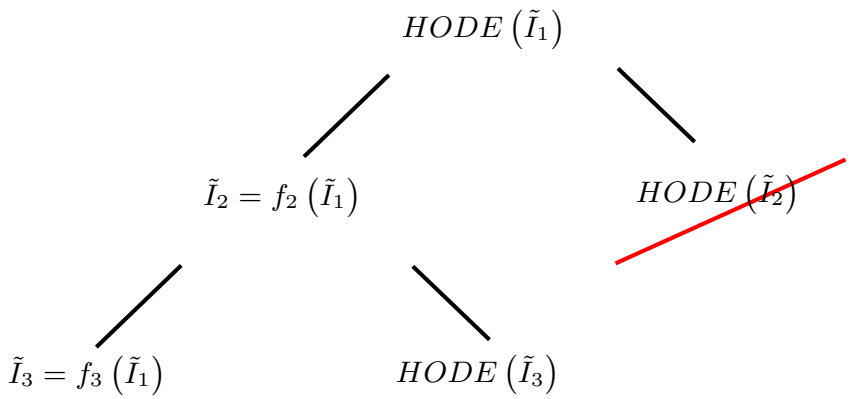
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1	2	2	(3)
$n_2^{(H)}$	0	1	2	2



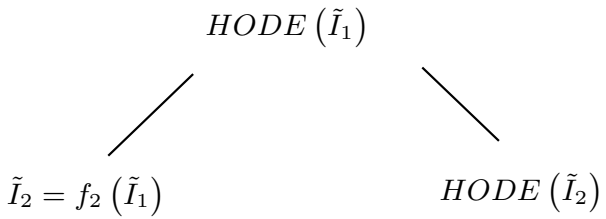
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_3^{(1)}$	-1	0	-1	-1
$\tilde{\mathcal{I}}_3^{(H)}$	0	1	0	0
$n_3^{(1)}$				
$n_3^{(H)}$				



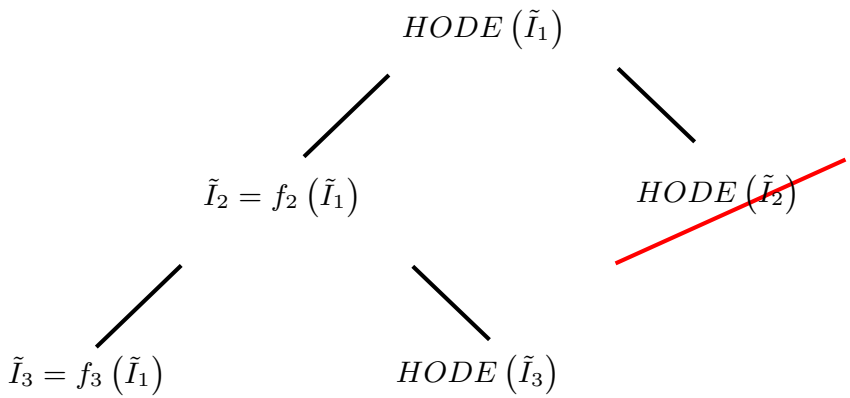
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1	2	2	(3)
$n_2^{(H)}$	0	1	2	2



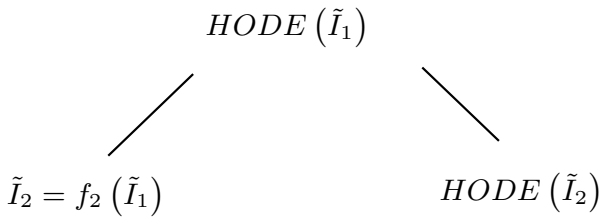
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_3^{(1)}$	-1	0	-1	-1
$\tilde{\mathcal{I}}_3^{(H)}$	0	1	0	0
$n_3^{(1)}$	0			
$n_3^{(H)}$	1			



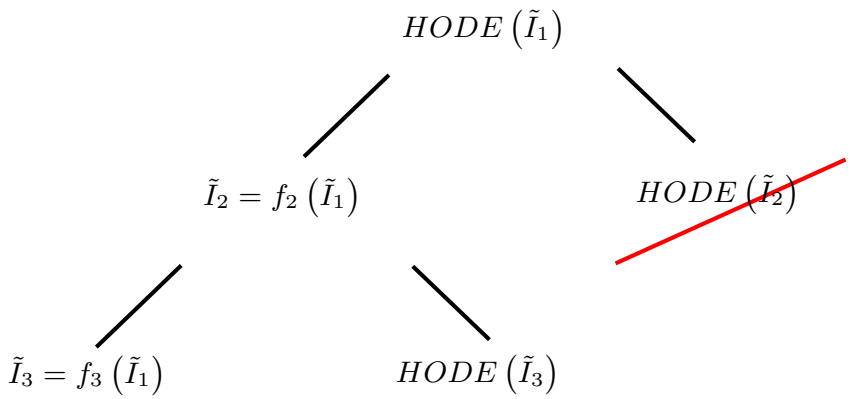
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1	2	2	(3)
$n_2^{(H)}$	0	1	2	2



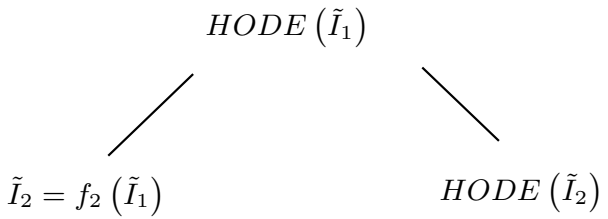
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_3^{(1)}$	-1	0	-1	-1
$\tilde{\mathcal{I}}_3^{(H)}$	0	1	0	0
$n_3^{(1)}$	0	0		
$n_3^{(H)}$	1	2		



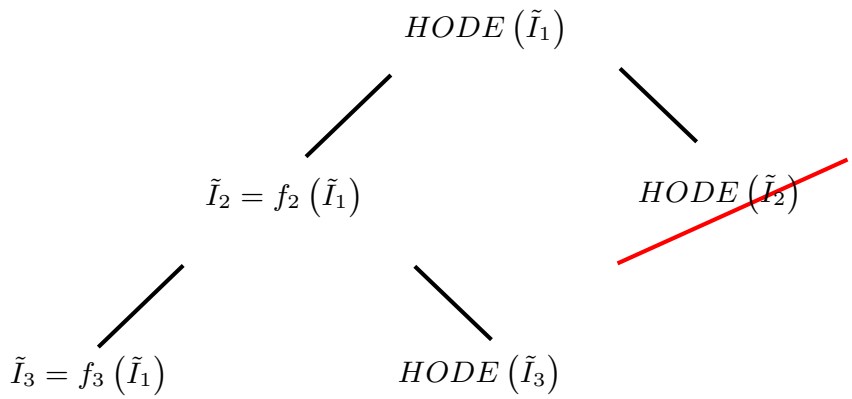
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1	2	2	(3)
$n_2^{(H)}$	0	1	2	2



	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_3^{(1)}$	-1	0	-1	-1
$\tilde{\mathcal{I}}_3^{(H)}$	0	1	0	0
$n_3^{(1)}$	0	0	0	
$n_3^{(H)}$	1	2	3	

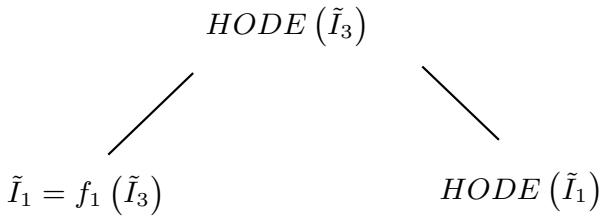


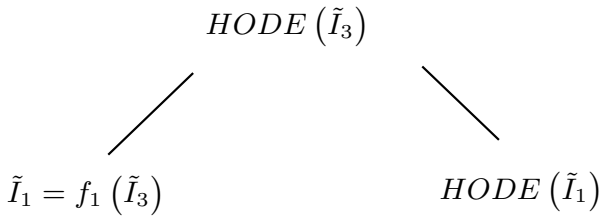
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1	2	2	(3)
$n_2^{(H)}$	0	1	2	2



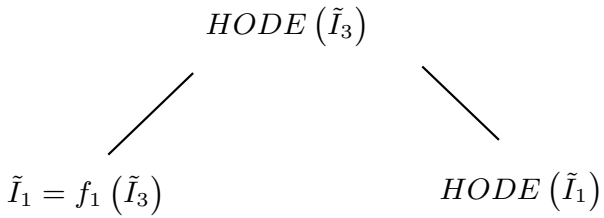
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_3^{(1)}$	-1	0	-1	-1
$\tilde{\mathcal{I}}_3^{(H)}$	0	1	0	0
$n_3^{(1)}$	0	0	0	0
$n_3^{(H)}$	1	2	3	(4)

First solution: $\mathrm{HODE}(\tilde{\mathcal{I}}_1),\ \tilde{\mathcal{I}}_2 = f_2(\tilde{\mathcal{I}}_1),\ \mathrm{HODE}(\tilde{\mathcal{I}}_3)$

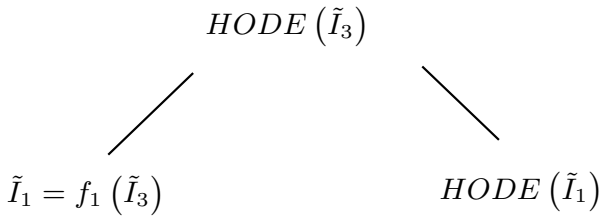

$HODE\left(\tilde{I}_3\right)$
$\tilde{I}_1 = f_1\left(\tilde{I}_3\right)$
$HODE\left(\tilde{I}_1\right)$



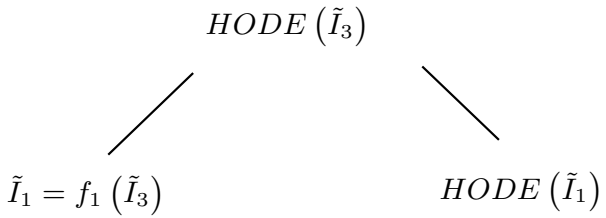
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(3)}$	0	0	0	0
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$n_1^{(3)}$				
$n_1^{(H)}$				



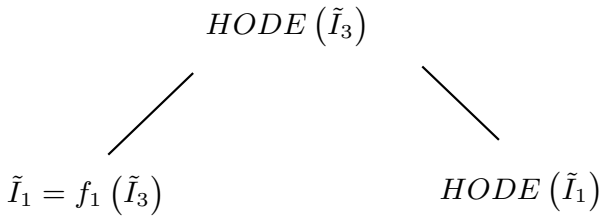
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(3)}$	0	0	0	0
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$n_1^{(3)}$	1			
$n_1^{(H)}$	1			



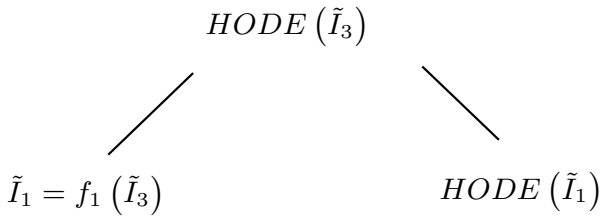
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(3)}$	0	0	0	0
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$n_1^{(3)}$	1	1		
$n_1^{(H)}$	1	2		



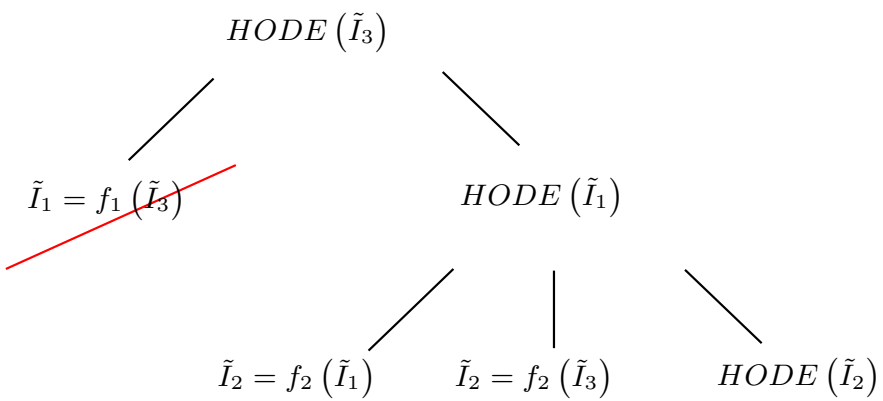
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(3)}$	0	0	0	0
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$n_1^{(3)}$	1	1	2	
$n_1^{(H)}$	1	2	3	

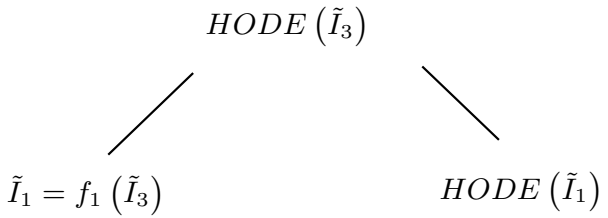


	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(3)}$	0	0	0	0
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$n_1^{(3)}$	1	1	2	3
$n_1^{(H)}$	1	2	3	(4)

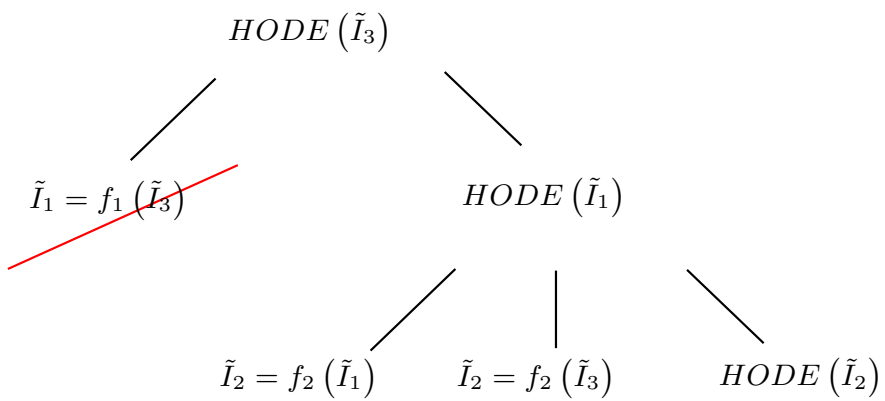


	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(3)}$	0	0	0	0
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$n_1^{(3)}$	1	1	2	3
$n_1^{(H)}$	1	2	3	(4)

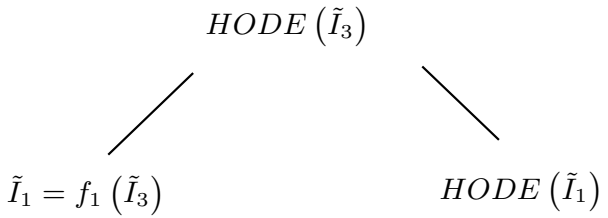




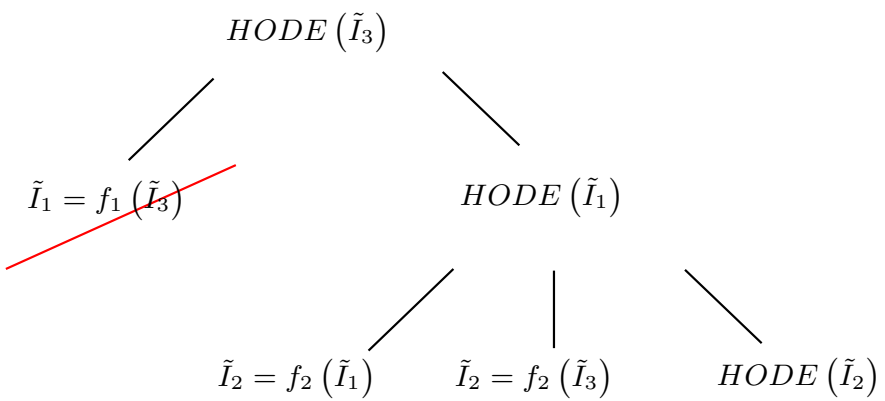
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(3)}$	0	0	0	0
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$n_1^{(3)}$	1	1	2	3
$n_1^{(H)}$	1	2	3	(4)



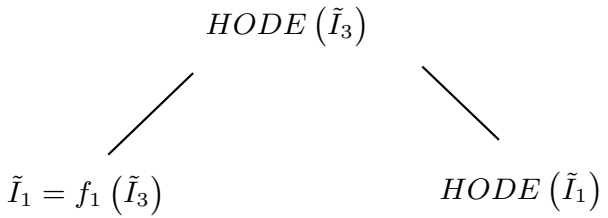
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(3)}$	0	1	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$				
$n_2^{(3)}$				
$n_2^{(H)}$				



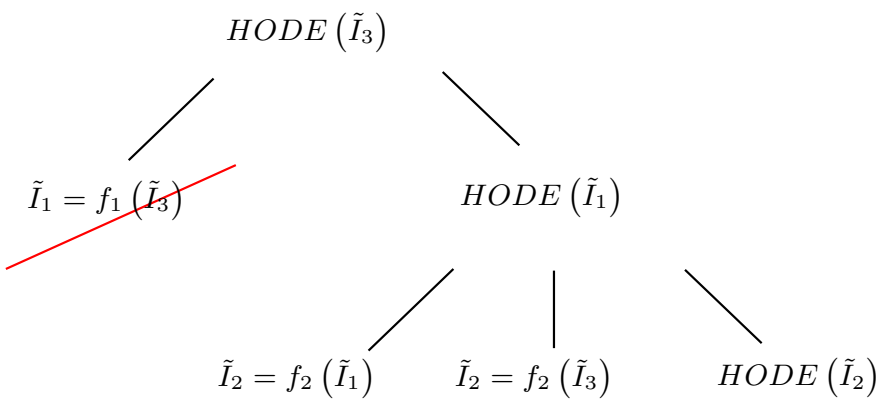
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(3)}$	0	0	0	0
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$n_1^{(3)}$	1	1	2	3
$n_1^{(H)}$	1	2	3	(4)



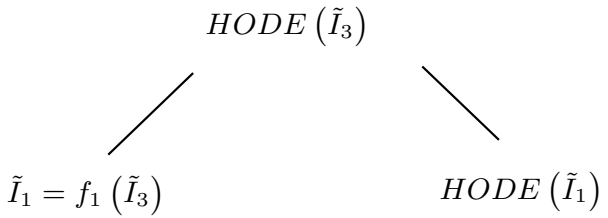
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(3)}$	0	1	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1			
$n_2^{(3)}$	1			
$n_2^{(H)}$	0			



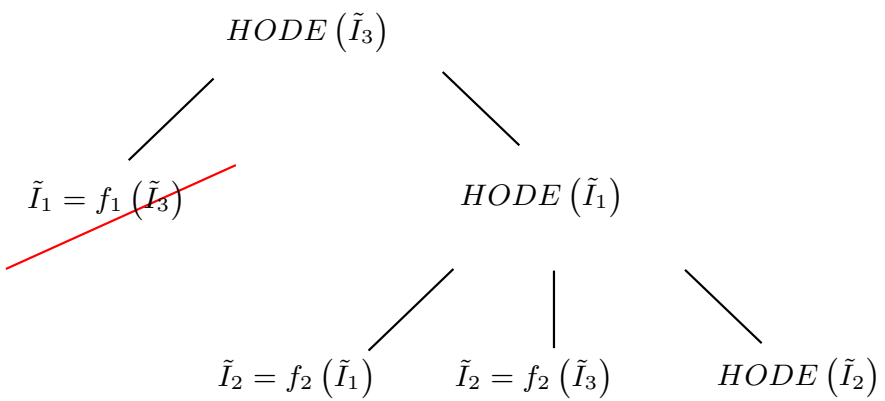
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(3)}$	0	0	0	0
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$n_1^{(3)}$	1	1	2	3
$n_1^{(H)}$	1	2	3	(4)



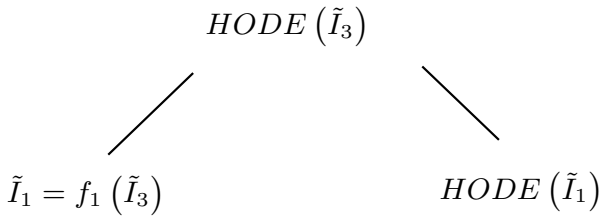
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(3)}$	0	1	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1	1		
$n_2^{(3)}$	1	2		
$n_2^{(H)}$	0	0		



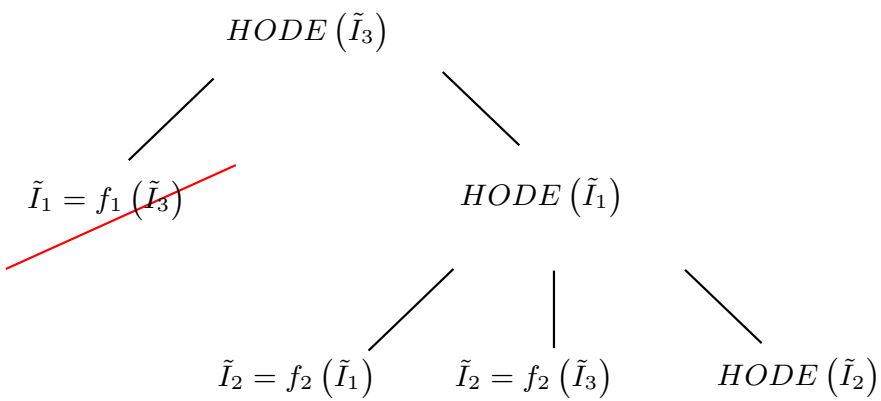
	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(3)}$	0	0	0	0
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$n_1^{(3)}$	1	1	2	3
$n_1^{(H)}$	1	2	3	(4)



	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(3)}$	0	1	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1	1	1	
$n_2^{(3)}$	1	2	2	
$n_2^{(H)}$	0	0	1	



	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_1^{(3)}$	0	0	0	0
$\tilde{\mathcal{I}}_1^{(H)}$	0	1	0	0
$n_1^{(3)}$	1	1	2	3
$n_1^{(H)}$	1	2	3	(4)



	$\tilde{\mathcal{J}}_1$	$\tilde{\mathcal{J}}_2$	$\tilde{\mathcal{J}}_3$	$\tilde{\mathcal{J}}_4$
$\tilde{\mathcal{I}}_2^{(1)}$	0	0	0	0
$\tilde{\mathcal{I}}_2^{(3)}$	0	1	0	0
$\tilde{\mathcal{I}}_2^{(H)}$	-1	0	1	-1
$n_2^{(1)}$	1	1	1	2
$n_2^{(3)}$	1	2	2	(3)
$n_2^{(H)}$	0	0	1	1

Second solution: $\mathrm{HODE}(\tilde{\mathcal{I}}_1),\ \tilde{\mathcal{I}}_2 = f_2(\tilde{\mathcal{I}}_3),\ \mathrm{HODE}(\tilde{\mathcal{I}}_3)$

Note: Doing this procedure means selecting the equations based only on the ϵ-orders. This might not be the actual optimal solution $\rightarrow$ **HODE order**, **recurrence order** play also an important role

Note: Doing this procedure means selecting the equations based only on the ϵ-orders. This might not be the actual optimal solution $\rightarrow$ **HODE order**, **recurrence order** play also an important role

- Denoting with n_ϵ the best $\epsilon-$order cumulative index, n_d the order of the HODE ($n_d = 0$ if it's a LF) and n_r the order of the recurence ($n_r = 0$ if it's a LF), we have to compare and "minimize" lists of the form:

$$\{-n_\epsilon, n_d, n_r\}$$

Note: Doing this procedure means selecting the equations based only on the ϵ-orders. This might not be the actual optimal solution $\rightarrow$ **HODE order**, **recurrence order** play also an important role

- Denoting with n_ϵ the best $\epsilon-$order cumulative index, n_d the order of the HODE ($n_d = 0$ if it's a LF) and n_r the order of the recurence ($n_r = 0$ if it's a LF), we have to compare and "minimize" lists of the form:

$$\{-n_\epsilon, n_d, n_r\}$$

We have included four weight functions:

1. Equal weight (default): $f_1(\{-n_\epsilon, n_d, n_r\}) := -n_\epsilon + n_d + n_r$
2. Only ϵ-order: $f_2(\{-n_\epsilon, n_d, n_r\}) := -n_\epsilon$
3. Only HODE order: $f_3(\{-n_\epsilon, n_d, n_r\}) := n_d$
4. Only recurence order: $f_4(\{-n_\epsilon, n_d, n_r\}) := n_r$

Note: Doing this procedure means selecting the equations based only on the ϵ-orders. This might not be the actual optimal solution $\rightarrow$ **HODE order**, **recurrence order** play also an important role

- Denoting with n_ϵ the best $\epsilon-$order cumulative index, n_d the order of the HODE ($n_d = 0$ if it's a LF) and n_r the order of the recurence ($n_r = 0$ if it's a LF), we have to compare and "minimize" lists of the form:

$$\{-n_\epsilon, n_d, n_r\}$$

  We have included four weight functions:

  1. Equal weight (default): $f_1(\{-n_\epsilon, n_d, n_r\}) := -n_\epsilon + n_d + n_r$
  2. Only ϵ-order: $f_2(\{-n_\epsilon, n_d, n_r\}) := -n_\epsilon$
  3. Only HODE order: $f_3(\{-n_\epsilon, n_d, n_r\}) := n_d$
  4. Only recurence order: $f_4(\{-n_\epsilon, n_d, n_r\}) := n_r$

- The user can also define himself the weight function f and give it as an optional argument to the main function

The procedure described above can actually be improved even further: in this recursive choice of equations, we prioritize the $\epsilon-$orders at the expense of the number of IC (several HODEs each of order n possible) $\rightarrow$ can be improved

The procedure described above can actually be improved even further: in this recursive choice of equations, we prioritize the $\epsilon-$orders at the expense of the number of IC (several HODEs each of order n possible) $\rightarrow$ can be improved

- Uncouple cyclically the system $S(\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_n)$ to get n^2 equations (n HODEs and $n(n-1)$ LFs)

The procedure described above can actually be improved even further: in this recursive choice of equations, we prioritize the $\epsilon-$orders at the expense of the number of IC (several HODEs each of order n possible) $\rightarrow$ can be improved

- Uncouple cyclically the system $S(\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_n)$ to get n^2 equations (n HODEs and $n(n-1)$ LFs)
- Choose as before the best HODE by comparing the $\epsilon-$orders, as before (wlog suppose the best HODE is $\tilde{\mathcal{I}}_1$)

The procedure described above can actually be improved even further: in this recursive choice of equations, we prioritize the $\epsilon-$orders at the expense of the number of IC (several HODEs each of order n possible) $\rightarrow$ can be improved

- Uncouple cyclically the system $S(\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_n)$ to get n^2 equations (n HODEs and $n(n-1)$ LFs)
- Choose as before the best HODE by comparing the $\epsilon-$orders, as before (wlog suppose the best HODE is $\tilde{\mathcal{I}}_1$)
- The next equation we need to find is for $\tilde{\mathcal{I}}_2$. This time, instead of the previous choice, we uncouple the subsystem $\tilde{S}_1(\tilde{\mathcal{I}}_2, \ldots, \tilde{\mathcal{I}}_n) \subsetneq S(\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_n)$ **where $\tilde{\mathcal{I}}_1$ is seen as a BI**

$$\begin{array}{cccc} & S & & \tilde{S}_1 \\ \text{MIs:} & \tilde{\mathcal{I}}_1, \tilde{\mathcal{I}}_2, \ldots, \tilde{\mathcal{I}}_n & \longrightarrow & \tilde{\mathcal{I}}_2, \ldots, \tilde{\mathcal{I}}_n \\ \text{BIs:} & \tilde{\mathcal{J}}_1, \ldots, \tilde{\mathcal{J}}_m & \longrightarrow & \tilde{\mathcal{I}}_1, \tilde{\mathcal{J}}_1 \ldots, \tilde{\mathcal{J}}_m \end{array}$$

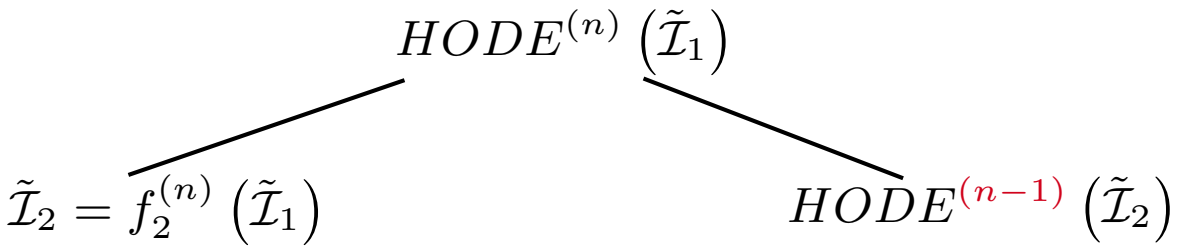
The procedure described above can actually be improved even further: in this recursive choice of equations, we prioritize the $\epsilon-$orders at the expense of the number of IC (several HODEs each of order n possible) $\rightarrow$ can be improved

- Uncouple cyclically the system $S(\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_n)$ to get n^2 equations (n HODEs and $n(n-1)$ LFs)
- Choose as before the best HODE by comparing the $\epsilon-$orders, as before (wlog suppose the best HODE is $\tilde{\mathcal{I}}_1$)
- The next equation we need to find is for $\tilde{\mathcal{I}}_2$. This time, instead of the previous choice, we uncouple the subsystem $\tilde{S}_1(\tilde{\mathcal{I}}_2, \ldots, \tilde{\mathcal{I}}_n) \subsetneq S(\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_n)$ **where $\tilde{\mathcal{I}}_1$ is seen as a BI**

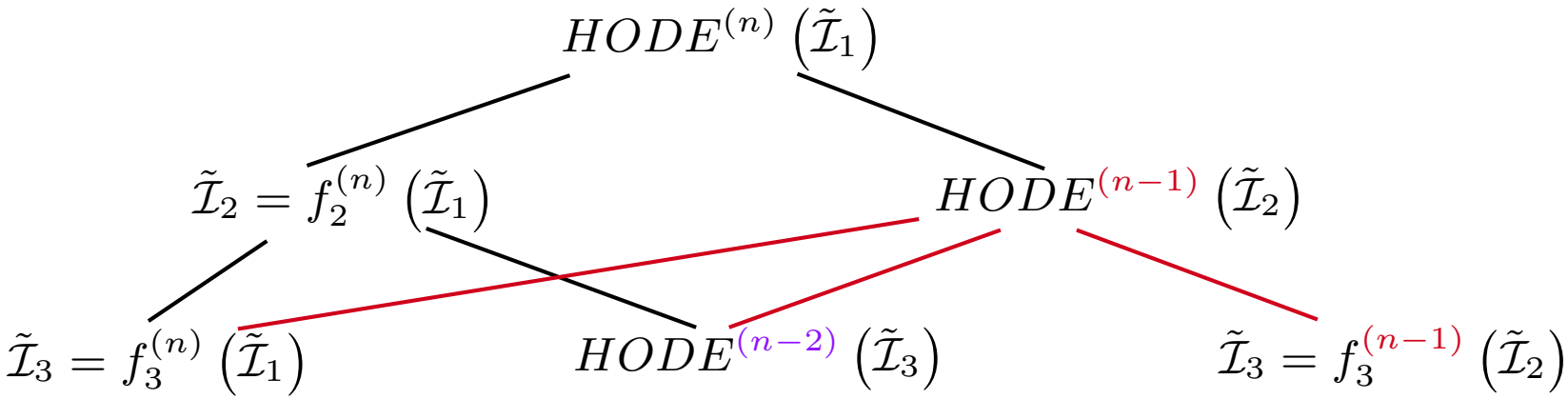
$$\begin{array}{cccc} & S & & \tilde{S}_1 \\ \text{MIs:} & \tilde{\mathcal{I}}_1, \tilde{\mathcal{I}}_2, \ldots, \tilde{\mathcal{I}}_n & \longrightarrow & \tilde{\mathcal{I}}_2, \ldots, \tilde{\mathcal{I}}_n \\ \text{BIs:} & \tilde{\mathcal{J}}_1, \ldots, \tilde{\mathcal{J}}_m & \longrightarrow & \tilde{\mathcal{I}}_1, \tilde{\mathcal{J}}_1 \ldots, \tilde{\mathcal{J}}_m \end{array}$$

- Use the new HODE for $\tilde{\mathcal{I}}_2$, of order $n-1$, instead of that obtained during the original uncoupling
- Reiterate for $\tilde{\mathcal{I}}_3$, etc.

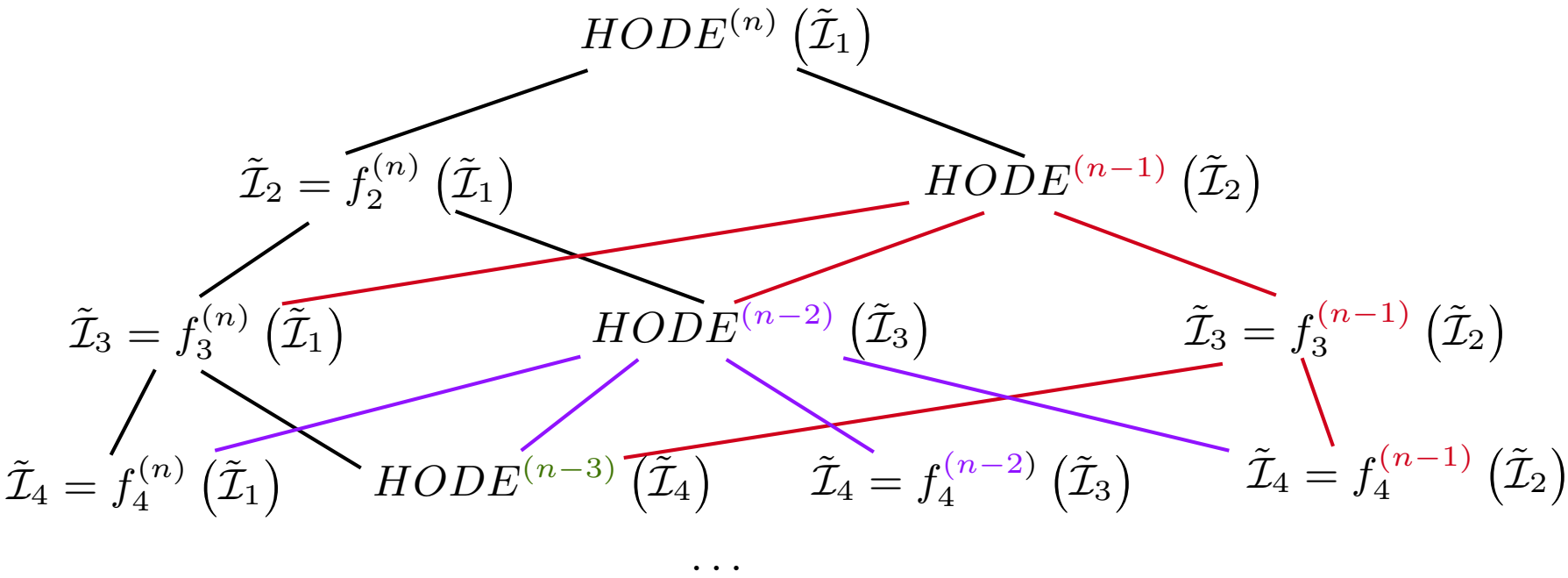
- This time, we have the following choice tree:



- This time, we have the following choice tree:



- This time, we have the following choice tree:



- In short: uncouple at step i in $\tilde{\mathcal{I}}_i$ by considering that all $\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_{i-1}$ are BIs, so that we reduce the orders of the HODEs

When we get several "good" solutions according to this algorithm, we need to find a way to select only one of them. How to do it?

When we get several "good" solutions according to this algorithm, we need to find a way to select only one of them. How to do it?

- For each "good" uncoupling, let's us denote with $m_{i,j}$ the $\epsilon-$order for the BI $\tilde{\mathcal{J}}_j(x,\epsilon)$, $1 \leq j \leq m$ appearing in the expression for the MI $\tilde{\mathcal{I}}_i(x,\epsilon)$, $1 \leq i \leq n$, and write the orders matrix:

$$M = (m_{i,j})_{1 \leq i \leq n, 1 \leq j \leq m}$$

Example:

$$M_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{matrix} \tilde{\mathcal{J}}_1^{(H)} \\ \tilde{\mathcal{J}}_2^{(1)} \\ \tilde{\mathcal{J}}_3^{(H)} \end{matrix}, \quad M_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{matrix} \tilde{\mathcal{J}}_1^{(H)} \\ \tilde{\mathcal{J}}_2^{(3)} \\ \tilde{\mathcal{J}}_3^{(H)} \end{matrix}$$

- Define the functions

$$\sigma : M \mapsto \sum_{i,j} m_{i,j}$$

$$\|\bullet\| : M \mapsto \sqrt{\mathrm{Tr}(M^\top M)}$$

And finally combine them in a function

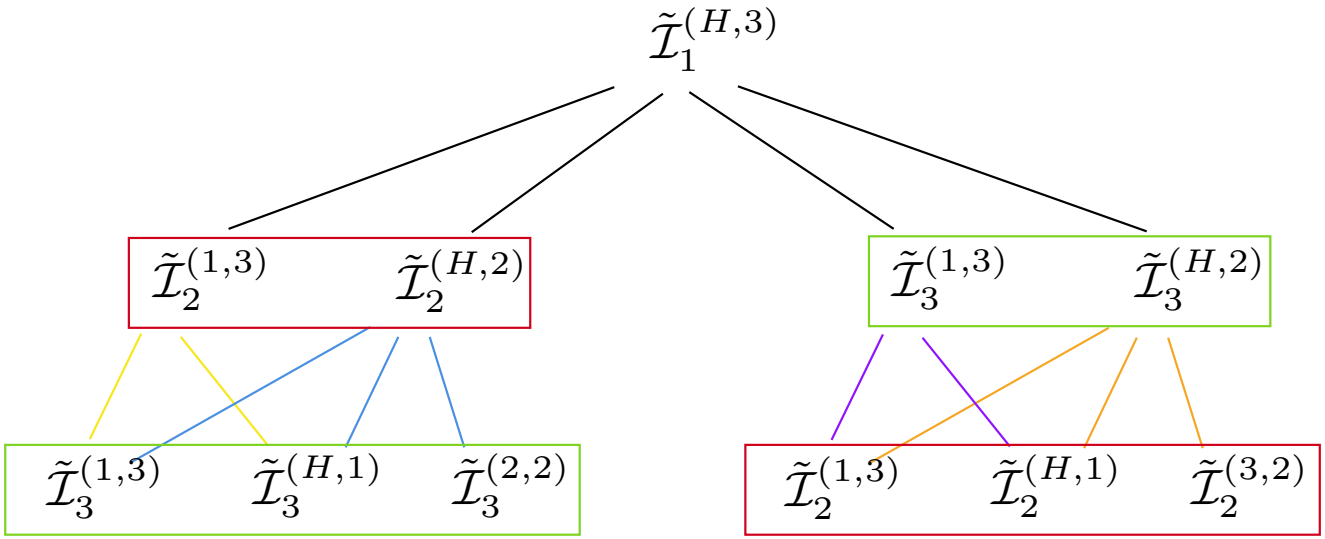
$$\boxed{\mathcal{N} : \begin{cases} \mathcal{M}_{n,m}(\mathbb{Z}) & \longrightarrow & \mathbb{Z} \times \mathbb{R}_- \\ M & \longmapsto & (\sigma(M), -\|M\|) \end{cases}}$$

- We obtain the "best" solution by finding the "minimal" $\mathcal{N}(M)$, i.e. the element that maximizes the sum of orders and minimizes their variance (in this order)

- One last improvement we can make to the Zürcher uncoupling is choosing among *all* the possible remaining MIs at each step and not only one of them

- One last improvement we can make to the Zürcher uncoupling is choosing among *all* the possible remaining MIs at each step and not only one of them

Example: Defining a shorter notation $\tilde{\mathcal{I}}_k^{(i,j)} := f_k^{(j)}(\tilde{\mathcal{I}}_i)$, $\tilde{\mathcal{I}}_k^{(H,i)} := \mathrm{HODE}^{(i)}(\tilde{\mathcal{I}}_k)$, for a system of order 3, here's the choice tree we would get:



- The second possible uncoupling scheme is the Gauß scheme. The result returned by `OreSys` in whole generality takes the following form:

Gauß uncoupling (OreSys output)

$$\begin{cases} \text{HODE}(\underline{\tilde{\mathcal{I}}_{i_1}, \{\mathrm{d}_x^j \tilde{\mathcal{I}}_{i_1}\}}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}) \\ \text{HODE}(\underline{\tilde{\mathcal{I}}_{i_2}, \{\mathrm{d}_x^j \tilde{\mathcal{I}}_{i_2}\}}; \tilde{\mathcal{I}}_{i_1}, \{\mathrm{d}_x^j \tilde{\mathcal{I}}_{i_1}\}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}) \\ \qquad\qquad \vdots \\ \text{HODE}(\underline{\tilde{\mathcal{I}}_{i_n}, \{\mathrm{d}_x^j \tilde{\mathcal{I}}_{i_n}\}}; \cdots; \tilde{\mathcal{I}}_{i_1}, \{\mathrm{d}_x^j \tilde{\mathcal{I}}_{i_1}\}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}) \end{cases}$$

- Compared to the Zürcher uncoupling, Gauß is more sensible to the order of the equations for the uncoupling, so we simple uncouple in all the possible ways, i.e. we get $n!$ uncouplings

- The second possible uncoupling scheme is the Gauß scheme. The result returned by `OreSys` in whole generality takes the following form:

Gauß uncoupling (OreSys output)

$$\begin{cases} \text{HODE}(\underline{\tilde{\mathcal{I}}_{i_1}, \{\mathrm{d}_x^j \tilde{\mathcal{I}}_{i_1}\}}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}) \\ \text{HODE}(\underline{\tilde{\mathcal{I}}_{i_2}, \{\mathrm{d}_x^j \tilde{\mathcal{I}}_{i_2}\}}; \tilde{\mathcal{I}}_{i_1}, \{\mathrm{d}_x^j \tilde{\mathcal{I}}_{i_1}\}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}) \\ \qquad\qquad \vdots \\ \text{HODE}(\underline{\tilde{\mathcal{I}}_{i_n}, \{\mathrm{d}_x^j \tilde{\mathcal{I}}_{i_n}\}}; \cdots; \tilde{\mathcal{I}}_{i_1}, \{\mathrm{d}_x^j \tilde{\mathcal{I}}_{i_1}\}; \{\tilde{\mathcal{R}}_i, \mathrm{d}_x^j \tilde{\mathcal{R}}_i\}) \end{cases}$$

- Compared to the Zürcher uncoupling, Gauß is more sensible to the order of the equations for the uncoupling, so we simple uncouple in all the possible ways, i.e. we get $n!$ uncouplings
- Finally, we can combine **both** Zürcher and Gauß uncouplings, i.e. at each step choose among **all** available equations (HODE and LF from Zürcher and Gauß) for remaining MIs: this is actually the scheme that is ultimately used, because it allows for the most freedom

Because we get to do a lot of different uncouplings ($2n-1$ for Zürcher and $n!$ for Gauß), we have implemented different "speed" schemes:

- **Slowest (greedy)**: The algorithm computes everything. Gives the best result, but finishes in reasonable time only for rather small systems (up to order ~ 5, and not too much subsystems, around ~ 10)

Because we get to do a lot of different uncouplings ($2n-1$ for Zürcher and $n!$ for Gauß), we have implemented different "speed" schemes:

- **Slowest (greedy)**: The algorithm computes everything. Gives the best result, but finishes in reasonable time only for rather small systems (up to order ~ 5, and not too much subsystems, around ~ 10)
- **Intermediate (optimal)**: The algorithm computes the Zürcher uncouplings with an upper time limit defined by the first uncoupling. For the Gauß part, it computes the average Zürcher time and limits the time for Gauß uncouplings. Faster, but a little bit less optimal

Because we get to do a lot of different uncouplings ($2n-1$ for Zürcher and $n!$ for Gauß), we have implemented different "speed" schemes:

- **Slowest (greedy)**: The algorithm computes everything. Gives the best result, but finishes in reasonable time only for rather small systems (up to order ~ 5, and not too much subsystems, around ~ 10)
- **Intermediate (optimal)**: The algorithm computes the Zürcher uncouplings with an upper time limit defined by the first uncoupling. For the Gauß part, it computes the average Zürcher time and limits the time for Gauß uncouplings. Faster, but a little bit less optimal
- **Fastest (lenient)**: The algorithm computes in parallel the Zürcher and the Gauß schemes and only gets the one that finishes first. Mostly for test purposes, or for exceptionally big systems for which the other schemes take too much time

Mathematica: All of this is implemented in our (beta) package `SystemAnalysis` into an algorithm called `bestUncpl`, that returns the best uncouplings (i.e. the collection of best HODEs/LFs) for Zürcher and Gauß schemes in a list

Recap:

- We clustered the system into smaller irreducible subsystems
- We uncoupled them in Zürcher and Gauß scheme and, using the method described above, obtained a set of "best" equations $\epsilon-$order-wise

Recap:

- We clustered the system into smaller irreducible subsystems
- We uncoupled them in Zürcher and Gauß scheme and, using the method described above, obtained a set of "best" equations $\epsilon-$order-wise

What remains to do?

- Compute for each subsystem in a symbolic way the order up to which we need to give the BIs in order to solve the system up to the user-required order in the MIs
- Using these results, compute the real $\epsilon-$order correction for each BI in a recursive way for the whole system

Recap:

- We clustered the system into smaller irreducible subsystems
- We uncoupled them in Zürcher and Gauß scheme and, using the method described above, obtained a set of "best" equations $\epsilon-$order-wise

What remains to do?

- Compute for each subsystem in a symbolic way the order up to which we need to give the BIs in order to solve the system up to the user-required order in the MIs
- Using these results, compute the real $\epsilon-$order correction for each BI in a recursive way for the whole system

Why?

- A non-irreducible system clusters non-trivially into several subsystems
- This gives rise to **induced BIs** (IBIs) in the subsystems, i.e. MIs that become locally BIs, and therefore the dependence of **fundamental BIs** (FBIs) might depend not only on the MIs where they appear, but also on other subsystems' MIs which have in the RHS the BIs' MIs as IBIs

- First, we extract the different $\epsilon-$order corrections in a completely general and symbolic way and make them into Mathematica rules.

Toy example 1: System in two variables $\tilde{\mathcal{I}}_1, \tilde{\mathcal{I}}_2$, with one BI $\mathcal{J}$, with the following uncoupling (for simplicity's sake, the dependence in x is suppressed):

$$\text{HODE}(\tilde{\mathcal{I}}_1) = -2\epsilon^8(1-2\epsilon^2)\mathcal{J}(x) + \epsilon^{27}\mathcal{J}'(x)$$

$$\tilde{\mathcal{I}}_2 = -\frac{\epsilon^{14}}{2}\tilde{\mathcal{I}}_1(x) + \frac{\epsilon^{19}}{2}\tilde{\mathcal{I}}_1' - 3\epsilon^{19}\mathcal{J}(x)$$

Initialisation:

$$(\tilde{\mathcal{I}}_1, a) \quad (\tilde{\mathcal{I}}_2, b)$$

- First, we extract the different $\epsilon-$order corrections in a completely general and symbolic way and make them into Mathematica rules.

Toy example 1: System in two variables $\tilde{\mathcal{I}}_1, \tilde{\mathcal{I}}_2$, with one BI $\mathcal{J}$, with the following uncoupling (for simplicity's sake, the dependence in x is suppressed):

$$\text{HODE}(\tilde{\mathcal{I}}_1) = -2\epsilon^8(1-2\epsilon^2)\mathcal{J}(x) + \epsilon^{27}\mathcal{J}'(x)$$

$$\tilde{\mathcal{I}}_2 = -\frac{\epsilon^{14}}{2}\tilde{\mathcal{I}}_1(x) + \frac{\epsilon^{19}}{2}\tilde{\mathcal{I}}_1' - 3\epsilon^{19}\mathcal{J}(x)$$

Initialisation:

$$(\tilde{\mathcal{I}}_1, a) \quad (\tilde{\mathcal{I}}_2, b)$$

HODE corrections:

$$(\tilde{\mathcal{I}}_1, \max(a, b-14))$$

- First, we extract the different $\epsilon-$order corrections in a completely general and symbolic way and make them into Mathematica rules.

Toy example 1: System in two variables $\tilde{\mathcal{I}}_1, \tilde{\mathcal{I}}_2$, with one BI $\mathcal{J}$, with the following uncoupling (for simplicity's sake, the dependence in x is suppressed):

$$\text{HODE}(\tilde{\mathcal{I}}_1) = -2\epsilon^8(1-2\epsilon^2)\mathcal{J}(x) + \epsilon^{27}\mathcal{J}'(x)$$

$$\tilde{\mathcal{I}}_2 = -\frac{\epsilon^{14}}{2}\tilde{\mathcal{I}}_1(x) + \frac{\epsilon^{19}}{2}\tilde{\mathcal{I}}_1' - 3\epsilon^{19}\mathcal{J}(x)$$

Initialisation:

$$(\tilde{\mathcal{I}}_1, a) \quad (\tilde{\mathcal{I}}_2, b)$$

HODE corrections:

$$(\tilde{\mathcal{I}}_1, \max(a, b-14))$$

$\mathcal{J}$ **correction:**

$$\max\left[\max(a, b-14) - 8,\right.$$

- First, we extract the different $\epsilon-$order corrections in a completely general and symbolic way and make them into Mathematica rules.

Toy example 1: System in two variables $\tilde{\mathcal{I}}_1, \tilde{\mathcal{I}}_2$, with one BI $\mathcal{J}$, with the following uncoupling (for simplicity's sake, the dependence in x is suppressed):

$$\text{HODE}(\tilde{\mathcal{I}}_1) = -2\epsilon^8(1-2\epsilon^2)\mathcal{J}(x) + \epsilon^{27}\mathcal{J}'(x)$$

$$\tilde{\mathcal{I}}_2 = -\frac{\epsilon^{14}}{2}\tilde{\mathcal{I}}_1(x) + \frac{\epsilon^{19}}{2}\tilde{\mathcal{I}}_1' - 3\epsilon^{19}\mathcal{J}(x)$$

Initialisation:

$$(\tilde{\mathcal{I}}_1, a) \quad (\tilde{\mathcal{I}}_2, b)$$

HODE corrections:

$$(\tilde{\mathcal{I}}_1, \max(a, b-14))$$

$\mathcal{J}$ **correction:**

$$\max\left[\max(a, b-14) - 8, b-19\right]$$

Problem: This only works if we choose equations from the simple n Zürcher, but if we choose from Gauß and sub-Zürcher uncouplings, this adds a recursive dependence of the BIs ϵ-orders on the chosen equations

Problem: This only works if we choose equations from the simple n Zürcher, but if we choose from Gauß and sub-Zürcher uncouplings, this adds a recursive dependence of the BIs ϵ-orders on the chosen equations

Toy example 2: Let's take as system of order $n = 4$ with 4 MIs $\tilde{\mathcal{I}}_{i=1,\dots,4}$ and 2 BIs $\tilde{\mathcal{J}}_{i=1,2}$, with the following chosen equations:

$$\begin{cases} \mathrm{HODE}(\tilde{\mathcal{I}}_1) &= \epsilon^{\alpha_1}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_1}\tilde{\mathcal{J}}_2 \\ \mathrm{HODE}(\tilde{\mathcal{I}}_4) &= \epsilon^{\alpha_2}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_2}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_2}\tilde{\mathcal{I}}_1 \\ \tilde{\mathcal{I}}_3 &= \epsilon^{\alpha_3}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_3}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_3}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_3}\tilde{\mathcal{I}}_4 \\ \tilde{\mathcal{I}}_2 &= \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3 \end{cases}$$

Problem: This only works if we choose equations from the simple n Zürcher, but if we choose from Gauß and sub-Zürcher uncouplings, this adds a recursive dependence of the BIs ϵ-orders on the chosen equations

Toy example 2: Let's take as system of order $n = 4$ with 4 MIs $\tilde{\mathcal{I}}_{i=1,\dots,4}$ and 2 BIs $\tilde{\mathcal{J}}_{i=1,2}$, with the following chosen equations:

$$\begin{cases} \mathrm{HODE}(\tilde{\mathcal{I}}_1) &= \epsilon^{\alpha_1}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_1}\tilde{\mathcal{J}}_2 \\ \mathrm{HODE}(\tilde{\mathcal{I}}_4) &= \epsilon^{\alpha_2}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_2}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_2}\tilde{\mathcal{I}}_1 \\ \tilde{\mathcal{I}}_3 &= \epsilon^{\alpha_3}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_3}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_3}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_3}\tilde{\mathcal{I}}_4 \\ \tilde{\mathcal{I}}_2 &= \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3 \end{cases}$$

Initialization: $(\tilde{\mathcal{I}}_1, a)$, $(\tilde{\mathcal{I}}_2, b)$, $(\tilde{\mathcal{I}}_3, c)$, $(\tilde{\mathcal{I}}_4, d)$

- We have to update recursively the ϵ-orders, from the last to the first equation, let us show how it works explicitely for $\tilde{\mathcal{J}}_1$

1. Updating with 4^{th} equation $\left(\tilde{\mathcal{I}}_2 = \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3\right)$

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b - \gamma_4))$$

1. Updating with 4^{th} equation $\left(\tilde{\mathcal{I}}_2 = \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3\right)$

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b - \gamma_4))$$
$$(\tilde{\mathcal{I}}_4, d_1 = \max(d, b - \delta_4))$$

1. Updating with 4^{th} equation $\left(\tilde{\mathcal{I}}_2 = \epsilon^{\alpha 4}\,\tilde{\mathcal{J}}_1 + \epsilon^{\beta 4}\,\tilde{\mathcal{J}}_2 + \epsilon^{\gamma 4}\,\tilde{\mathcal{I}}_1 + \epsilon^{\delta 4}\,\tilde{\mathcal{I}}_4 + \epsilon^{\eta 4}\,\tilde{\mathcal{I}}_3\right)$

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b - \gamma_4))$$
$$(\tilde{\mathcal{I}}_4, d_1 = \max(d, b - \delta_4))$$
$$(\tilde{\mathcal{I}}_3, c_1 = \max(c, b - \eta_4))$$

1. Updating with 4^{th} equation $\left(\tilde{\mathcal{I}}_2 = \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3\right)$

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b - \gamma_4)) \quad (\tilde{\mathcal{I}}_2, b_1 = b - \alpha_4)$$
$$(\tilde{\mathcal{I}}_4, d_1 = \max(d, b - \delta_4))$$
$$(\tilde{\mathcal{I}}_3, c_1 = \max(c, b - \eta_4))$$

$$(\tilde{\mathcal{I}}_1, a_1),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_1),\ (\tilde{\mathcal{I}}_4, d_1)$$

1. Updating with 4^{th} equation $\left(\tilde{\mathcal{I}}_2 = \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3\right)$

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b - \gamma_4)) \quad (\tilde{\mathcal{I}}_2, b_1 = b - \alpha_4)$$
$$(\tilde{\mathcal{I}}_4, d_1 = \max(d, b - \delta_4))$$
$$(\tilde{\mathcal{I}}_3, c_1 = \max(c, b - \eta_4))$$
$$(\tilde{\mathcal{I}}_1, a_1),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_1),\ (\tilde{\mathcal{I}}_4, d_1)$$

2. Updating with 3^{th} equation $\left(\tilde{\mathcal{I}}_3 = \epsilon^{\alpha_3}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_3}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_3}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_3}\tilde{\mathcal{I}}_4\right)$

$$(\tilde{\mathcal{I}}_1, a_2 = \max(a_1, c_1 - \gamma_3))$$

1. Updating with 4^{th} equation $\left(\tilde{\mathcal{I}}_2 = \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3\right)$

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b - \gamma_4)) \quad (\tilde{\mathcal{I}}_2, b_1 = b - \alpha_4)$$
$$(\tilde{\mathcal{I}}_4, d_1 = \max(d, b - \delta_4))$$
$$(\tilde{\mathcal{I}}_3, c_1 = \max(c, b - \eta_4))$$
$$(\tilde{\mathcal{I}}_1, a_1),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_1),\ (\tilde{\mathcal{I}}_4, d_1)$$

2. Updating with 3^{th} equation $\left(\tilde{\mathcal{I}}_3 = \epsilon^{\alpha_3}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_3}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_3}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_3}\tilde{\mathcal{I}}_4\right)$

$$(\tilde{\mathcal{I}}_1, a_2 = \max(a_1, c_1 - \gamma_3))$$
$$(\tilde{\mathcal{I}}_4, d_2 = \max(d_1, c_1 - \delta_3))$$

1. Updating with 4^{th} equation $\left(\tilde{\mathcal{I}}_2 = \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3\right)$

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b - \gamma_4)) \quad (\tilde{\mathcal{I}}_2, b_1 = b - \alpha_4)$$
$$(\tilde{\mathcal{I}}_4, d_1 = \max(d, b - \delta_4))$$
$$(\tilde{\mathcal{I}}_3, c_1 = \max(c, b - \eta_4))$$
$$(\tilde{\mathcal{I}}_1, a_1),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_1),\ (\tilde{\mathcal{I}}_4, d_1)$$

2. Updating with 3^{th} equation $\left(\tilde{\mathcal{I}}_3 = \epsilon^{\alpha_3}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_3}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_3}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_3}\tilde{\mathcal{I}}_4\right)$

$$(\tilde{\mathcal{I}}_1, a_2 = \max(a_1, c_1 - \gamma_3)) \quad (\tilde{\mathcal{I}}_3, c_2 = c_1 - \alpha_3)$$
$$(\tilde{\mathcal{I}}_4, d_2 = \max(d_1, c_1 - \delta_3))$$
$$(\tilde{\mathcal{I}}_1, a_2),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_2),\ (\tilde{\mathcal{I}}_4, d_2)$$

1. Updating with 4^{th} equation $\left(\tilde{\mathcal{I}}_2 = \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3\right)$

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b - \gamma_4)) \quad (\tilde{\mathcal{I}}_2, b_1 = b - \alpha_4)$$
$$(\tilde{\mathcal{I}}_4, d_1 = \max(d, b - \delta_4))$$
$$(\tilde{\mathcal{I}}_3, c_1 = \max(c, b - \eta_4))$$
$$(\tilde{\mathcal{I}}_1, a_1),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_1),\ (\tilde{\mathcal{I}}_4, d_1)$$

2. Updating with 3^{th} equation $\left(\tilde{\mathcal{I}}_3 = \epsilon^{\alpha_3}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_3}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_3}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_3}\tilde{\mathcal{I}}_4\right)$

$$(\tilde{\mathcal{I}}_1, a_2 = \max(a_1, c_1 - \gamma_3)) \quad (\tilde{\mathcal{I}}_3, c_2 = c_1 - \alpha_3)$$
$$(\tilde{\mathcal{I}}_4, d_2 = \max(d_1, c_1 - \delta_3))$$
$$(\tilde{\mathcal{I}}_1, a_2),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_2),\ (\tilde{\mathcal{I}}_4, d_2)$$

3. Updating with 2^{nd} equation $\left(\text{HODE}(\tilde{\mathcal{I}}_4) = \epsilon^{\alpha_2}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_2}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_2}\tilde{\mathcal{I}}_1\right)$

$$(\tilde{\mathcal{I}}_1, a_3 = \max(a_2, d_2 - \gamma_2))$$

1. Updating with 4^{th} equation $\left(\tilde{\mathcal{I}}_2 = \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3\right)$

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b - \gamma_4)) \quad (\tilde{\mathcal{I}}_2, b_1 = b - \alpha_4)$$
$$(\tilde{\mathcal{I}}_4, d_1 = \max(d, b - \delta_4))$$
$$(\tilde{\mathcal{I}}_3, c_1 = \max(c, b - \eta_4))$$
$$(\tilde{\mathcal{I}}_1, a_1),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_1),\ (\tilde{\mathcal{I}}_4, d_1)$$

2. Updating with 3^{th} equation $\left(\tilde{\mathcal{I}}_3 = \epsilon^{\alpha_3}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_3}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_3}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_3}\tilde{\mathcal{I}}_4\right)$

$$(\tilde{\mathcal{I}}_1, a_2 = \max(a_1, c_1 - \gamma_3)) \quad (\tilde{\mathcal{I}}_3, c_2 = c_1 - \alpha_3)$$
$$(\tilde{\mathcal{I}}_4, d_2 = \max(d_1, c_1 - \delta_3))$$
$$(\tilde{\mathcal{I}}_1, a_2),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_2),\ (\tilde{\mathcal{I}}_4, d_2)$$

3. Updating with 2^{nd} equation $\left(\text{HODE}(\tilde{\mathcal{I}}_4) = \epsilon^{\alpha_2}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_2}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_2}\tilde{\mathcal{I}}_1\right)$

$$(\tilde{\mathcal{I}}_1, a_3 = \max(a_2, d_2 - \gamma_2)) \quad (\tilde{\mathcal{I}}_4, d_3 = d_2 - \alpha_2)$$
$$(\tilde{\mathcal{I}}_1, a_3),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_2),\ (\tilde{\mathcal{I}}_4, d_3)$$

1. Updating with 4^{th} equation $\left(\tilde{\mathcal{I}}_2 = \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3\right)$

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b - \gamma_4)) \quad (\tilde{\mathcal{I}}_2, b_1 = b - \alpha_4)$$
$$(\tilde{\mathcal{I}}_4, d_1 = \max(d, b - \delta_4))$$
$$(\tilde{\mathcal{I}}_3, c_1 = \max(c, b - \eta_4))$$
$$(\tilde{\mathcal{I}}_1, a_1),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_1),\ (\tilde{\mathcal{I}}_4, d_1)$$

2. Updating with 3^{th} equation $\left(\tilde{\mathcal{I}}_3 = \epsilon^{\alpha_3}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_3}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_3}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_3}\tilde{\mathcal{I}}_4\right)$

$$(\tilde{\mathcal{I}}_1, a_2 = \max(a_1, c_1 - \gamma_3)) \quad (\tilde{\mathcal{I}}_3, c_2 = c_1 - \alpha_3)$$
$$(\tilde{\mathcal{I}}_4, d_2 = \max(d_1, c_1 - \delta_3))$$
$$(\tilde{\mathcal{I}}_1, a_2),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_2),\ (\tilde{\mathcal{I}}_4, d_2)$$

3. Updating with 2^{nd} equation $\left(\text{HODE}(\tilde{\mathcal{I}}_4) = \epsilon^{\alpha_2}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_2}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_2}\tilde{\mathcal{I}}_1\right)$

$$(\tilde{\mathcal{I}}_1, a_3 = \max(a_2, d_2 - \gamma_2)) \quad (\tilde{\mathcal{I}}_4, d_3 = d_2 - \alpha_2)$$
$$(\tilde{\mathcal{I}}_1, a_3),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_2),\ (\tilde{\mathcal{I}}_4, d_3)$$

4. Updating with 1^{st} equation $\left(\text{HODE}(\tilde{\mathcal{I}}_1) = \epsilon^{\alpha_1}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_1}\tilde{\mathcal{J}}_2\right)$

$$\emptyset$$

1. Updating with 4^{th} equation $\left(\tilde{\mathcal{I}}_2 = \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3\right)$

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b - \gamma_4)) \quad (\tilde{\mathcal{I}}_2, b_1 = b - \alpha_4)$$
$$(\tilde{\mathcal{I}}_4, d_1 = \max(d, b - \delta_4))$$
$$(\tilde{\mathcal{I}}_3, c_1 = \max(c, b - \eta_4))$$
$$(\tilde{\mathcal{I}}_1, a_1),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_1),\ (\tilde{\mathcal{I}}_4, d_1)$$

2. Updating with 3^{th} equation $\left(\tilde{\mathcal{I}}_3 = \epsilon^{\alpha_3}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_3}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_3}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_3}\tilde{\mathcal{I}}_4\right)$

$$(\tilde{\mathcal{I}}_1, a_2 = \max(a_1, c_1 - \gamma_3)) \quad (\tilde{\mathcal{I}}_3, c_2 = c_1 - \alpha_3)$$
$$(\tilde{\mathcal{I}}_4, d_2 = \max(d_1, c_1 - \delta_3))$$
$$(\tilde{\mathcal{I}}_1, a_2),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_2),\ (\tilde{\mathcal{I}}_4, d_2)$$

3. Updating with 2^{nd} equation $\left(\text{HODE}(\tilde{\mathcal{I}}_4) = \epsilon^{\alpha_2}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_2}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_2}\tilde{\mathcal{I}}_1\right)$

$$(\tilde{\mathcal{I}}_1, a_3 = \max(a_2, d_2 - \gamma_2)) \quad (\tilde{\mathcal{I}}_4, d_3 = d_2 - \alpha_2)$$
$$(\tilde{\mathcal{I}}_1, a_3),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_2),\ (\tilde{\mathcal{I}}_4, d_3)$$

4. Updating with 1^{st} equation $\left(\text{HODE}(\tilde{\mathcal{I}}_1) = \epsilon^{\alpha_1}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_1}\tilde{\mathcal{J}}_2\right)$

$$\emptyset \qquad (\tilde{\mathcal{I}}_1, a_4 = a_3 - \alpha_1)$$
$$(\tilde{\mathcal{I}}_1, a_4),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_2),\ (\tilde{\mathcal{I}}_4, d_3)$$

1. Updating with 4^{th} equation $\left(\tilde{\mathcal{I}}_2 = \epsilon^{\alpha_4}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_4}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_4}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_4}\tilde{\mathcal{I}}_4 + \epsilon^{\eta_4}\tilde{\mathcal{I}}_3\right)$

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b-\gamma_4)) \quad (\tilde{\mathcal{I}}_2, b_1 = b-\alpha_4)$$
$$(\tilde{\mathcal{I}}_4, d_1 = \max(d, b-\delta_4))$$
$$(\tilde{\mathcal{I}}_3, c_1 = \max(c, b-\eta_4))$$
$$(\tilde{\mathcal{I}}_1, a_1),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_1),\ (\tilde{\mathcal{I}}_4, d_1)$$

2. Updating with 3^{th} equation $\left(\tilde{\mathcal{I}}_3 = \epsilon^{\alpha_3}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_3}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_3}\tilde{\mathcal{I}}_1 + \epsilon^{\delta_3}\tilde{\mathcal{I}}_4\right)$

$$(\tilde{\mathcal{I}}_1, a_2 = \max(a_1, c_1-\gamma_3)) \quad (\tilde{\mathcal{I}}_3, c_2 = c_1-\alpha_3)$$
$$(\tilde{\mathcal{I}}_4, d_2 = \max(d_1, c_1-\delta_3))$$
$$(\tilde{\mathcal{I}}_1, a_2),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_2),\ (\tilde{\mathcal{I}}_4, d_2)$$

3. Updating with 2^{nd} equation $\left(\text{HODE}(\tilde{\mathcal{I}}_4) = \epsilon^{\alpha_2}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_2}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_2}\tilde{\mathcal{I}}_1\right)$

$$(\tilde{\mathcal{I}}_1, a_3 = \max(a_2, d_2-\gamma_2)) \quad (\tilde{\mathcal{I}}_4, d_3 = d_2-\alpha_2)$$
$$(\tilde{\mathcal{I}}_1, a_3),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_2),\ (\tilde{\mathcal{I}}_4, d_3)$$

4. Updating with 1^{st} equation $\left(\text{HODE}(\tilde{\mathcal{I}}_1) = \epsilon^{\alpha_1}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_1}\tilde{\mathcal{J}}_2\right)$

$$\emptyset \qquad (\tilde{\mathcal{I}}_1, a_4 = a_3-\alpha_1)$$
$$(\tilde{\mathcal{I}}_1, a_4),\ (\tilde{\mathcal{I}}_2, b_1),\ (\tilde{\mathcal{I}}_3, c_2),\ (\tilde{\mathcal{I}}_4, d_3)$$

The correction for $\tilde{\mathcal{J}}_1$ is:

$$\max(a_4, b_1, c_2, d_3)$$

- This generalizes very easily if there are derivatives of the MIs/BIs, e.g. if the second equation of the example was

$$\mathrm{HODE}(\tilde{\mathcal{I}}_4) = \epsilon^{\alpha_2}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_2}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_2}\tilde{\mathcal{I}}_1 + \epsilon^{\gamma_2^{(1)}}\mathrm{d}_x\tilde{\mathcal{I}}_1$$

- This generalizes very easily if there are derivatives of the MIs/BIs, e.g. if the second equation of the example was

$$\mathrm{HODE}(\tilde{\mathcal{I}}_4) = \epsilon^{\alpha_2}\tilde{\mathcal{J}}_1 + \epsilon^{\beta_2}\tilde{\mathcal{J}}_2 + \epsilon^{\gamma_2}\tilde{\mathcal{I}}_1 + \epsilon^{\gamma_2^{(1)}}\mathrm{d}_x\tilde{\mathcal{I}}_1$$

Then at step 3 the correction would be

$$(\tilde{\mathcal{I}}_1, a_3 = \max(a_2, d_2 - \gamma_2, d_2 - \gamma_2^{(1)})$$

- Let us use this recursive algorithm with the toy example 1 as a sanity check:

$$\text{HODE}(\tilde{\mathcal{I}}_1) = -2\epsilon^8(1-2\epsilon^2)\mathcal{J} + \epsilon^{27}\mathcal{J}'$$

$$\tilde{\mathcal{I}}_2 = -\frac{\epsilon^{14}}{2}\tilde{\mathcal{I}}_1 + \frac{\epsilon^{19}}{2}\tilde{\mathcal{I}}_1' - 3\epsilon^{19}\mathcal{J}$$

Initialization: (I_1, a), $(\tilde{\mathcal{I}}_2, b)$

- Let us use this recursive algorithm with the toy example 1 as a sanity check:

$$\text{HODE}(\tilde{\mathcal{I}}_1) = -2\epsilon^8(1-2\epsilon^2)\mathcal{J} + \epsilon^{27}\mathcal{J}'$$

$$\tilde{\mathcal{I}}_2 = -\frac{\epsilon^{14}}{2}\tilde{\mathcal{I}}_1 + \frac{\epsilon^{19}}{2}\tilde{\mathcal{I}}_1' - 3\epsilon^{19}\mathcal{J}$$

Initialization: (I_1, a), $(\tilde{\mathcal{I}}_2, b)$

1. Updating with 2^{nd} equation:

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b-14))$$

- Let us use this recursive algorithm with the toy example 1 as a sanity check:

$$\text{HODE}(\tilde{\mathcal{I}}_1) = -2\epsilon^8(1-2\epsilon^2)\mathcal{J} + \epsilon^{27}\mathcal{J}'$$

$$\tilde{\mathcal{I}}_2 = -\frac{\epsilon^{14}}{2}\tilde{\mathcal{I}}_1 + \frac{\epsilon^{19}}{2}\tilde{\mathcal{I}}_1' - 3\epsilon^{19}\mathcal{J}$$

Initialization: (I_1, a), $(\tilde{\mathcal{I}}_2, b)$

1. Updating with 2^{nd} equation:

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b-14)) \qquad (\tilde{\mathcal{I}}_2, b_1 = b-19)$$

- Let us use this recursive algorithm with the toy example 1 as a sanity check:

$$\text{HODE}(\tilde{\mathcal{I}}_1) = -2\epsilon^8(1-2\epsilon^2)\mathcal{J} + \epsilon^{27}\mathcal{J}'$$

$$\tilde{\mathcal{I}}_2 = -\frac{\epsilon^{14}}{2}\tilde{\mathcal{I}}_1 + \frac{\epsilon^{19}}{2}\tilde{\mathcal{I}}_1' - 3\epsilon^{19}\mathcal{J}$$

Initialization: (I_1, a), $(\tilde{\mathcal{I}}_2, b)$

1. Updating with 2^{nd} equation:

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b-14)) \qquad (\tilde{\mathcal{I}}_2, b_1 = b-19)$$

2. Updating with 1^{st} equation:

$$\emptyset$$

- Let us use this recursive algorithm with the toy example 1 as a sanity check:

$$\text{HODE}(\tilde{\mathcal{I}}_1) = -2\epsilon^8(1-2\epsilon^2)\mathcal{J} + \epsilon^{27}\mathcal{J}'$$

$$\tilde{\mathcal{I}}_2 = -\frac{\epsilon^{14}}{2}\tilde{\mathcal{I}}_1 + \frac{\epsilon^{19}}{2}\tilde{\mathcal{I}}_1' - 3\epsilon^{19}\mathcal{J}$$

Initialization: (I_1, a), $(\tilde{\mathcal{I}}_2, b)$

1. Updating with 2^{nd} equation:

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b-14)) \qquad (\tilde{\mathcal{I}}_2, b_1 = b-19)$$

2. Updating with 1^{st} equation:

$$\emptyset \qquad\qquad (\tilde{\mathcal{I}}_1, a_2 = a_1 - 8)$$

- Let us use this recursive algorithm with the toy example 1 as a sanity check:

$$\text{HODE}(\tilde{\mathcal{I}}_1) = -2\epsilon^8(1-2\epsilon^2)\mathcal{J} + \epsilon^{27}\mathcal{J}'$$

$$\tilde{\mathcal{I}}_2 = -\frac{\epsilon^{14}}{2}\tilde{\mathcal{I}}_1 + \frac{\epsilon^{19}}{2}\tilde{\mathcal{I}}_1' - 3\epsilon^{19}\mathcal{J}$$

Initialization: (I_1, a), $(\tilde{\mathcal{I}}_2, b)$

1. Updating with 2^{nd} equation:

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b-14)) \qquad (\tilde{\mathcal{I}}_2, b_1 = b-19)$$

2. Updating with 1^{st} equation:

$$\emptyset \qquad\qquad (\tilde{\mathcal{I}}_1, a_2 = a_1 - 8)$$

$$\tilde{\mathcal{J}} \;:\; \max(a_2, b_1) = \max(\max(a, b-14) - 8, b-19) \quad \checkmark$$

- Let us use this recursive algorithm with the toy example 1 as a sanity check:

$$\text{HODE}(\tilde{\mathcal{I}}_1) = -2\epsilon^8(1-2\epsilon^2)\mathcal{J} + \epsilon^{27}\mathcal{J}'$$

$$\tilde{\mathcal{I}}_2 = -\frac{\epsilon^{14}}{2}\tilde{\mathcal{I}}_1 + \frac{\epsilon^{19}}{2}\tilde{\mathcal{I}}_1' - 3\epsilon^{19}\mathcal{J}$$

Initialization: $(I_1, a),\ (\tilde{\mathcal{I}}_2, b)$

1. Updating with 2^{nd} equation:

$$(\tilde{\mathcal{I}}_1, a_1 = \max(a, b-14)) \qquad (\tilde{\mathcal{I}}_2, b_1 = b-19)$$

2. Updating with 1^{st} equation:

$$\emptyset \qquad\qquad (\tilde{\mathcal{I}}_1, a_2 = a_1 - 8)$$

$$\tilde{\mathcal{J}} \ : \ \max(a_2, b_1) = \max(\max(a, b-14) - 8, b-19) \quad \checkmark$$

Mathematica: A subfunction in our package does this computation and returns a symbolic "function-rule" that we need for the next step. In our example, the rule would look like:

$$\{\{\tilde{\mathcal{I}}_1, a\_\}, \{\tilde{\mathcal{I}}_2, b\_\}\} \longrightarrow \{\{\{\tilde{\mathcal{I}}_1, \max(a, b-14)\}, \{\tilde{\mathcal{I}}_2, b\}\}, \{\{\{\mathcal{J}, \max[\max(a, b-14) - 8, b-19]\}\}\}$$

Now that we have the local correction rules, we are almost done:

1. We identify what we call the **root MIs** (RMIs), i.e. the MIs that do not appear as IBIs in any equation, since these won't get any retro-correction and define in a sense the "root" of the correction trees

Now that we have the local correction rules, we are almost done:

1. We identify what we call the **root MIs** (RMIs), i.e. the MIs that do not appear as IBIs in any equation, since these won't get any retro-correction and define in a sense the "root" of the correction trees
2. We apply the aforementioned correction rules recursively on the RMIs, which gives rise to a tree where the roots are the RMIs, the nodes are the IBIs and the treetops are the FBIs

Now that we have the local correction rules, we are almost done:

1. We identify what we call the **root MIs** (RMIs), i.e. the MIs that do not appear as IBIs in any equation, since these won't get any retro-correction and define in a sense the "root" of the correction trees
2. We apply the aforementioned correction rules recursively on the RMIs, which gives rise to a tree where the roots are the RMIs, the nodes are the IBIs and the treetops are the FBIs

Note: It is clear that this procedure terminates as each step will either:

- Add a new variable (MI or BI) $\rightarrow$ there is only a finite number of them
- Nothing happens, but that means the algorithm is finished

Example:

- We have an initial system of order $n = 5$, i.e. the MIs are $\tilde{\mathcal{I}}_i(x, \epsilon)$, $i \in \{1, \ldots, 5\}$, and there is one FBI $\mathcal{J}$
- Suppose the system splits into three irreducible subsystems:

MIs	(Full) Dependence
$\tilde{\mathcal{I}}_1$	$\tilde{\mathcal{I}}_1, \tilde{\mathcal{I}}_2, \tilde{\mathcal{I}}_3, \tilde{\mathcal{I}}_4, \tilde{\mathcal{I}}_5; \mathcal{J}$
$\tilde{\mathcal{I}}_2, \tilde{\mathcal{I}}_3$	$\tilde{\mathcal{I}}_2, \tilde{\mathcal{I}}_3, \tilde{\mathcal{I}}_4, \tilde{\mathcal{I}}_5; \mathcal{J}$
$\tilde{\mathcal{I}}_4, \tilde{\mathcal{I}}_5$	$\tilde{\mathcal{I}}_4, \tilde{\mathcal{I}}_5; \mathcal{J}$

Example:

- We have an initial system of order $n = 5$, i.e. the MIs are $\tilde{\mathcal{I}}_i(x, \epsilon)$, $i \in \{1, \ldots, 5\}$, and there is one FBI $\mathcal{J}$
- Suppose the system splits into three irreducible subsystems:

MIs	(Full) Dependence
$\tilde{\mathcal{I}}_1$	$\tilde{\mathcal{I}}_1, \tilde{\mathcal{I}}_2, \tilde{\mathcal{I}}_3, \tilde{\mathcal{I}}_4, \tilde{\mathcal{I}}_5; \mathcal{J}$
$\tilde{\mathcal{I}}_2, \tilde{\mathcal{I}}_3$	$\tilde{\mathcal{I}}_2, \tilde{\mathcal{I}}_3, \tilde{\mathcal{I}}_4, \tilde{\mathcal{I}}_5; \mathcal{J}$
$\tilde{\mathcal{I}}_4, \tilde{\mathcal{I}}_5$	$\tilde{\mathcal{I}}_4, \tilde{\mathcal{I}}_5; \mathcal{J}$

- We have only one RMI in this case: $\tilde{\mathcal{I}}_1$

- We suppose that after the computation of the local $\epsilon-$order corrections, we obtain the following rules:

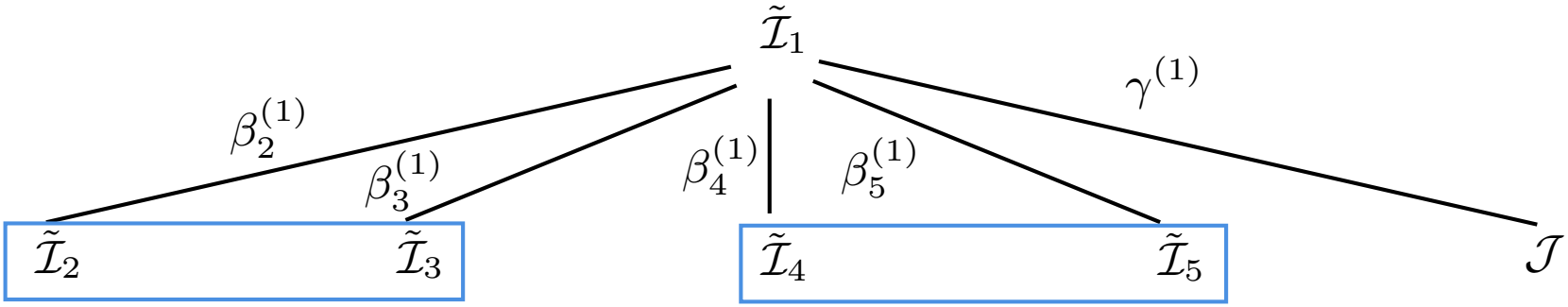
$$\{\{\tilde{\mathcal{I}}_1, a_1\}\} \to \{\{\{\tilde{\mathcal{I}}_1, \alpha_1^{(1)}\}\}, \{\{\{\tilde{\mathcal{I}}_2, \beta_2^{(1)}\}, \{\tilde{\mathcal{I}}_3, \beta_3^{(1)}\}, \{\tilde{\mathcal{I}}_4, \beta_4^{(1)}\}, \{\tilde{\mathcal{I}}_5, \beta_5^{(1)}\}, \{\mathcal{J}, \gamma^{(1)}\}\}\}\}$$

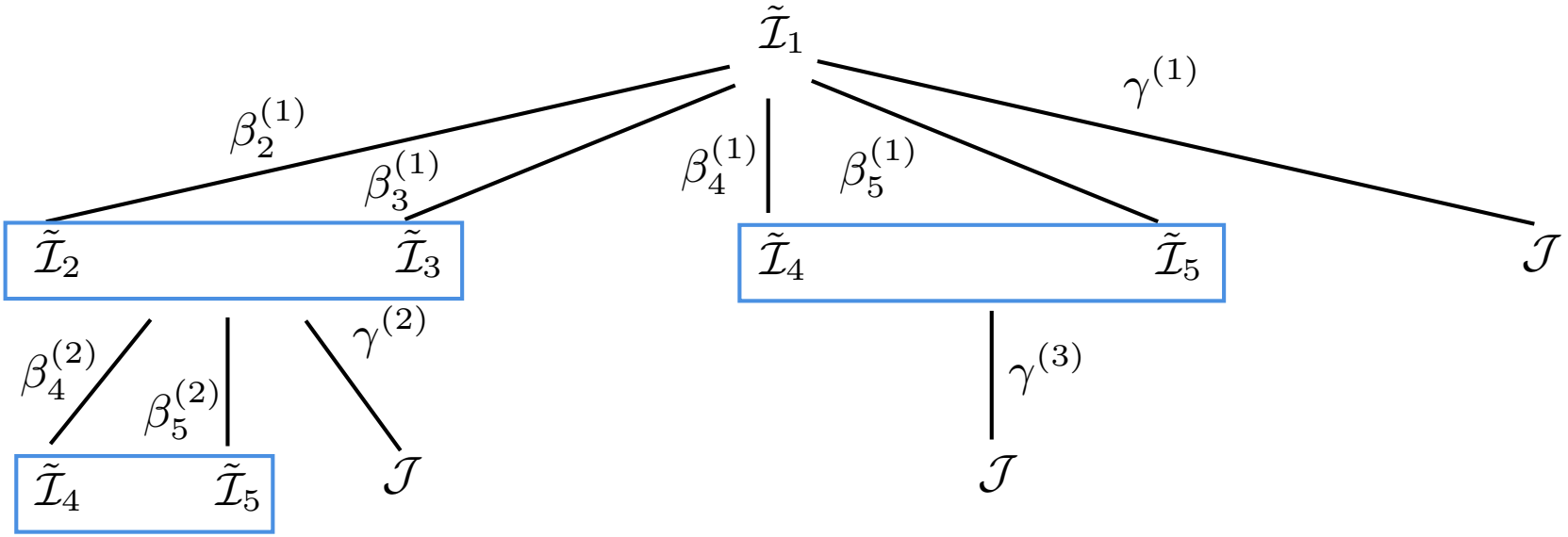
$$\{\{\tilde{\mathcal{I}}_2, a_2\}, \{\tilde{\mathcal{I}}_3, a_3\}\} \to \{\{\{\tilde{\mathcal{I}}_2, \alpha_2^{(2)}\}, \{\tilde{\mathcal{I}}_3, \alpha_3^{(2)}\}\}, \{\{\{\tilde{\mathcal{I}}_4, \beta_4^{(2)}\}, \{\tilde{\mathcal{I}}_5, \beta_5^{(2)}\}, \{\mathcal{J}, \gamma^{(2)}\}\}\}$$

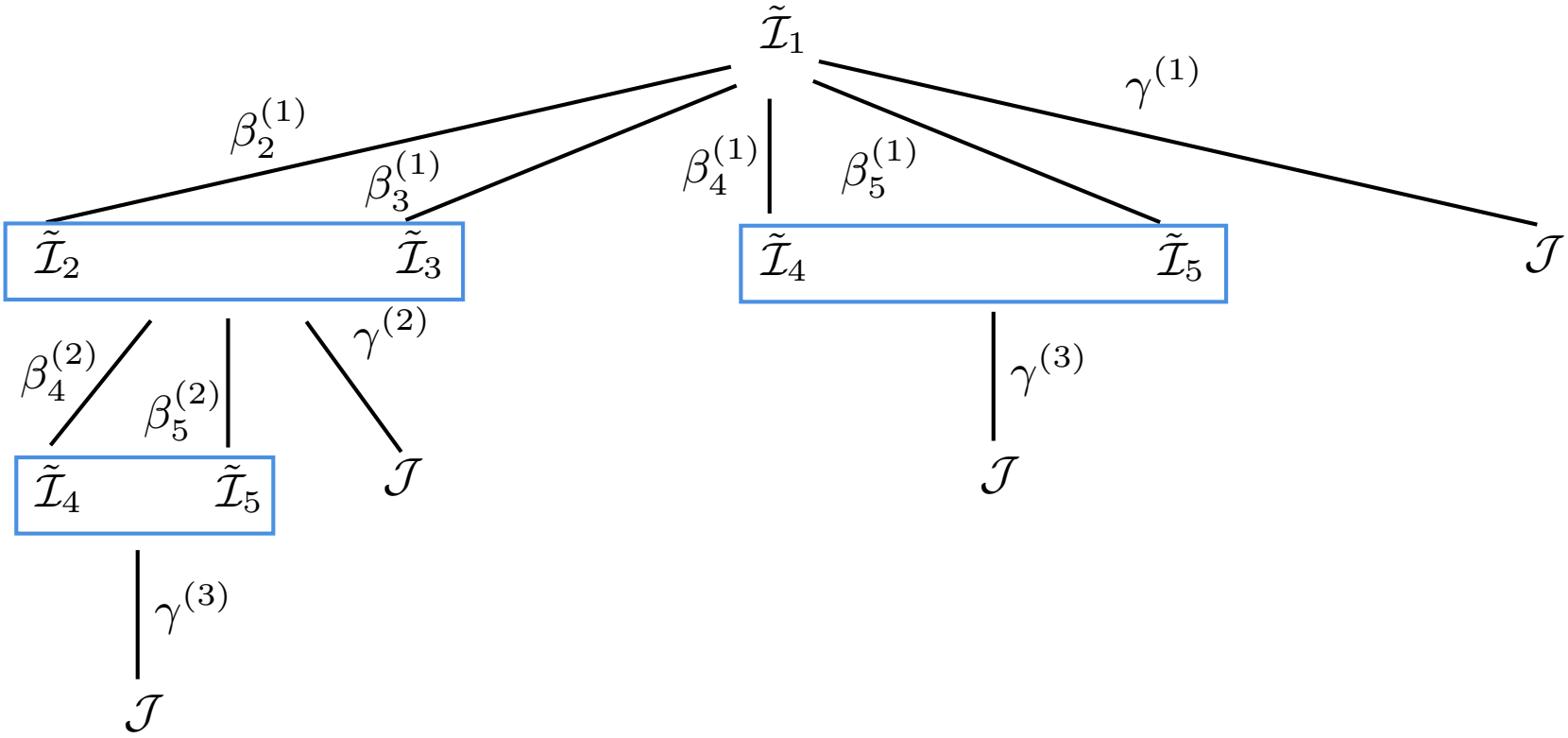
$$\{\{\tilde{\mathcal{I}}_4, a_4\}, \{\tilde{\mathcal{I}}_5, a_5\}\} \to \{\{\{\tilde{\mathcal{I}}_4, \alpha_4^{(3)}\}, \{\tilde{\mathcal{I}}_5, \alpha_5^{(3)}\}\}, \{\{\{\mathcal{J}, \gamma^{(3)}\}\}\}$$

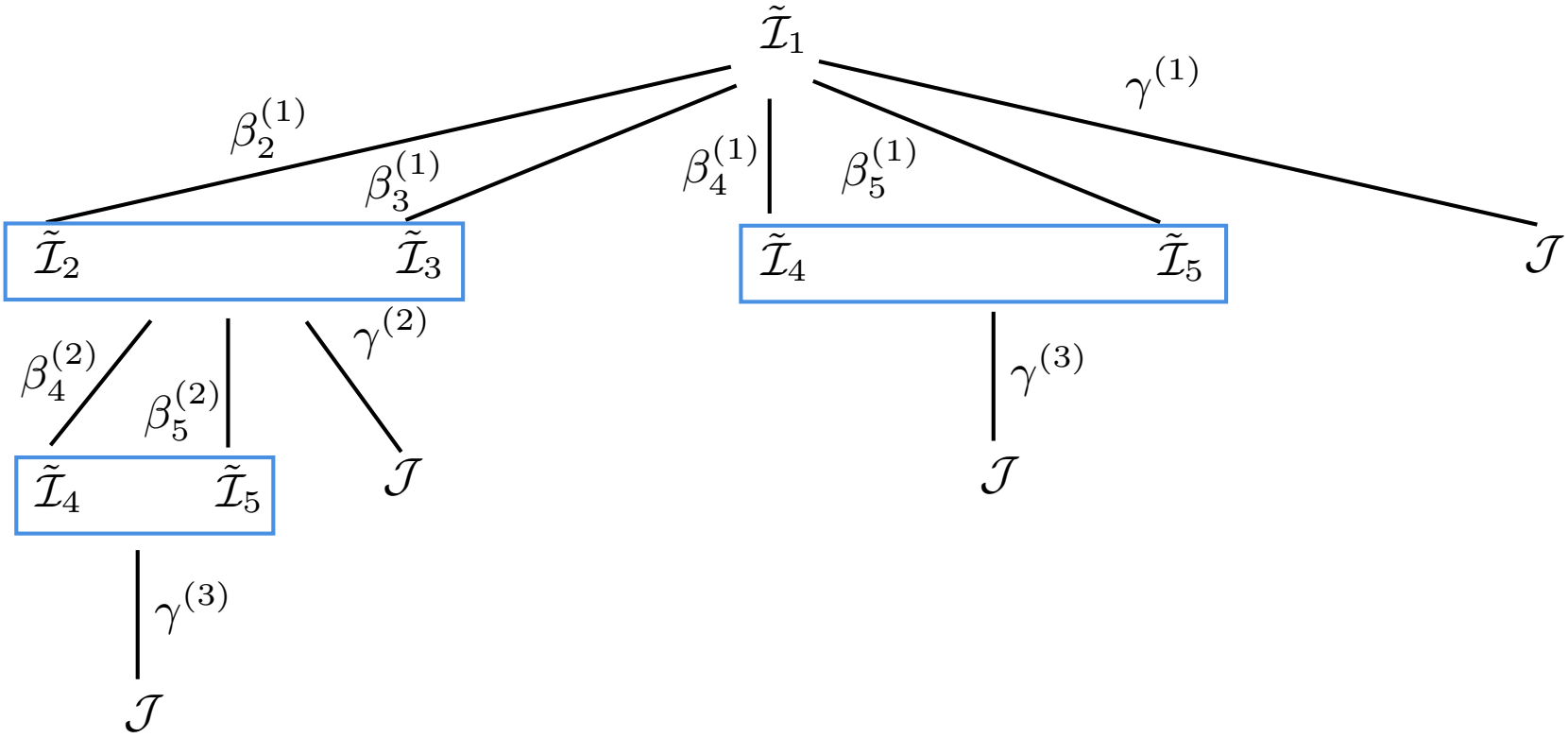
Where the $\alpha_i^{(j)}, \beta_i^{(j)}, \gamma^{(j)}$ are functions of the a_i

$$\tilde{\mathcal{I}}_1$$

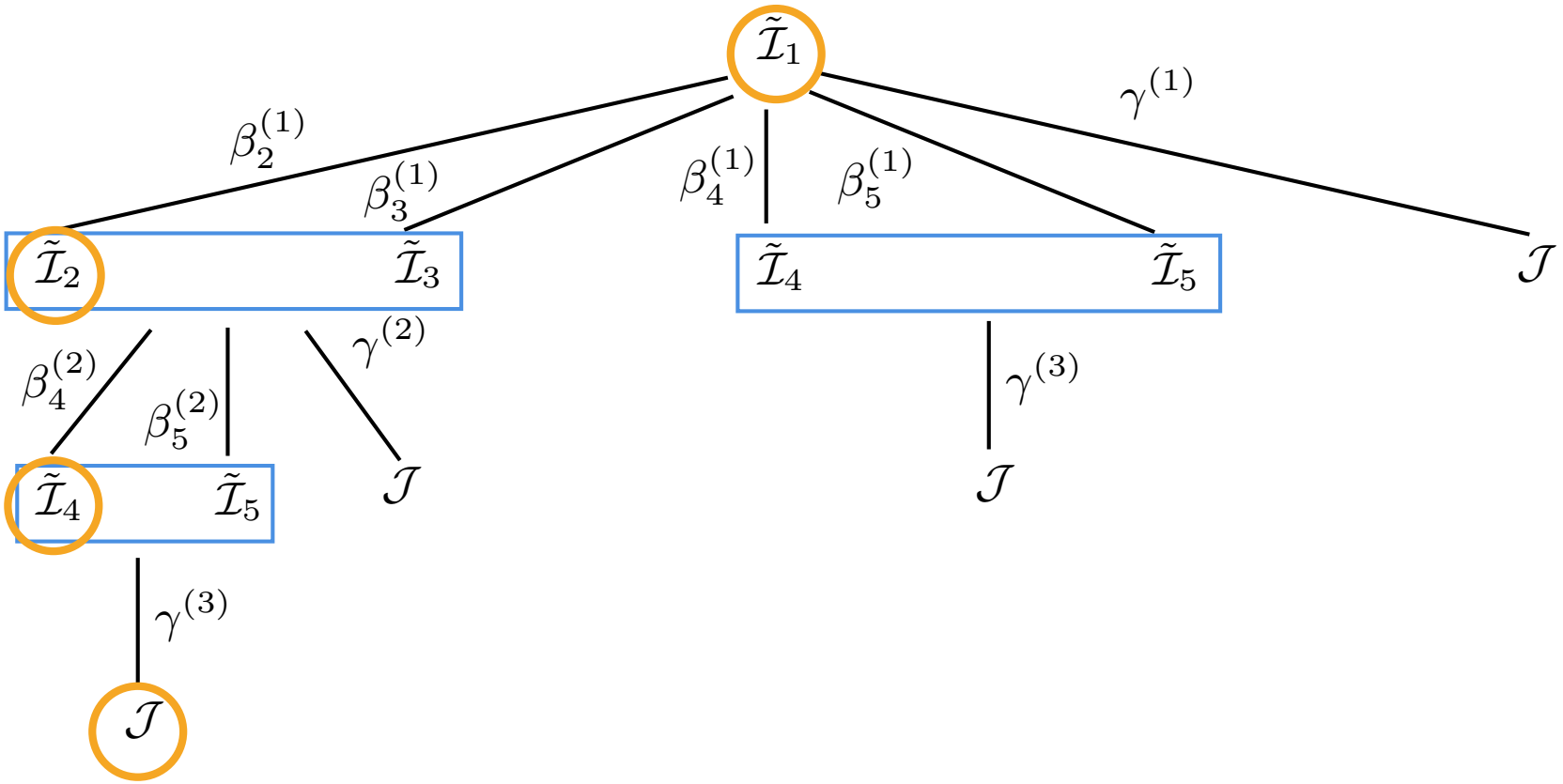

$\tilde{\mathcal{I}}_1$
$\beta_2^{(1)}$
$\beta_3^{(1)}$
$\beta_4^{(1)}$
$\beta_5^{(1)}$
$\gamma^{(1)}$
$\tilde{\mathcal{I}}_2$
$\tilde{\mathcal{I}}_3$
$\tilde{\mathcal{I}}_4$
$\tilde{\mathcal{I}}_5$
$\mathcal{J}$


$\tilde{\mathcal{I}}_1$
$\beta_2^{(1)}$
$\beta_3^{(1)}$
$\beta_4^{(1)}$
$\beta_5^{(1)}$
$\gamma^{(1)}$
$\tilde{\mathcal{I}}_2$
$\tilde{\mathcal{I}}_3$
$\tilde{\mathcal{I}}_4$
$\tilde{\mathcal{I}}_5$
$\mathcal{J}$
$\beta_4^{(2)}$
$\beta_5^{(2)}$
$\gamma^{(2)}$
$\gamma^{(3)}$
$\tilde{\mathcal{I}}_4$
$\tilde{\mathcal{I}}_5$
$\mathcal{J}$
$\mathcal{J}$


$\tilde{\mathcal{I}}_1$
$\beta_2^{(1)}$
$\beta_3^{(1)}$
$\beta_4^{(1)}$
$\beta_5^{(1)}$
$\gamma^{(1)}$
$\tilde{\mathcal{I}}_2$
$\tilde{\mathcal{I}}_3$
$\tilde{\mathcal{I}}_4$
$\tilde{\mathcal{I}}_5$
$\mathcal{J}$
$\beta_4^{(2)}$
$\beta_5^{(2)}$
$\gamma^{(2)}$
$\gamma^{(3)}$
$\tilde{\mathcal{I}}_4$
$\tilde{\mathcal{I}}_5$
$\mathcal{J}$
$\gamma^{(3)}$
$\mathcal{J}$



Note: The corrections are applied in a recursive way, i.e. following a branch we need to compose the $\alpha_i^{(j)}/\beta_i^{(j)}/\gamma^{(j)}$ functions from the outermost node all the way down to the root.



Note: The corrections are applied in a recursive way, i.e. following a branch we need to compose the $\alpha_i^{(j)}/\beta_i^{(j)}/\gamma^{(j)}$ functions from the outermost node all the way down to the root. Example:

$$\boxed{\gamma^{(3)} \circ \beta_4^{(2)} \circ \beta_2^{(1)}(a_1)}$$

- We collect in this way all the corrections at the end of the branches, and we get the following list:

$$L_{\text{corr}} : \left\{ \begin{array}{ccccc} \gamma^{(3)} \circ \beta_2^{(4)} \circ \beta_2^{(1)}, & \gamma^{(3)} \circ \beta_2^{(1)} \circ \beta_3^{(1)}, & \gamma^{(3)} \circ \beta_4^{(1)}, & \gamma^{(3)} \circ \beta_5^{(1)}, & \gamma^{(1)} \\ \gamma^{(3)} \circ \beta_5^{(2)} \circ \beta_2^{(1)}, & \gamma^{(3)} \circ \beta_5^{(2)} \circ \beta_3^{(1)}, & & & \\ \gamma^{(2)} \circ \beta_1^{(2)}, & \gamma^{(2)} \circ \beta_1^{(3)}, & & & \end{array} \right\}$$

- Since we have to satisfy all of these corrections, i.e. $\mathcal{J}$ will effectively get at some step of the computations ϵ coefficients to the power of each element of this list, we simply return $\max(L_{\text{corr}})$ as the $\epsilon-$order correction for $\mathcal{J}$

- We collect in this way all the corrections at the end of the branches, and we get the following list:

$$L_{\text{corr}} : \left\{ \begin{array}{ccccc} \gamma^{(3)} \circ \beta_2^{(4)} \circ \beta_2^{(1)}, & \gamma^{(3)} \circ \beta_2^{(1)} \circ \beta_3^{(1)}, & \gamma^{(3)} \circ \beta_4^{(1)}, & \gamma^{(3)} \circ \beta_5^{(1)}, & \gamma^{(1)} \\ \gamma^{(3)} \circ \beta_5^{(2)} \circ \beta_2^{(1)}, & \gamma^{(3)} \circ \beta_5^{(2)} \circ \beta_3^{(1)}, & & & \\ \gamma^{(2)} \circ \beta_1^{(2)}, & \gamma^{(2)} \circ \beta_1^{(3)}, & & & \end{array} \right\}$$

- Since we have to satisfy all of these corrections, i.e. $\mathcal{J}$ will effectively get at some step of the computations ϵ coefficients to the power of each element of this list, we simply return $\max(L_{\text{corr}})$ as the $\epsilon-$order correction for $\mathcal{J}$
- It suffices to replace now all the symbolic orders by the actual orders given as input by the user in order to get the corrections for the BIs

Notes:

- In the case of several sets of RMIs, i.e. several trees, we would have to return the maximum maximorum of all the lists $L_{i,\text{corr}}$, i.e. $\max(\max(L_{i,\text{corr}}))$

Notes:

- In the case of several sets of RMIs, i.e. several trees, we would have to return the maximum maximorum of all the lists $L_{i,\text{corr}}$, i.e. $\max(\max(L_{i,\text{corr}}))$
- Actually, it might happen that some subsystem contains an incomplete set of IBIs, i.e. only some of the MIs that make up the main variables of another subsystem. In this case, we simply compute "reduced" rules by replacing the missing variables symbolic a_i' by their real values, given as input by the user.

Example: Supposing that $\tilde{\mathcal{I}}_5$ is missing for the IBI, we compute the following reduced rule:

$$\{\{\tilde{\mathcal{I}}_4, a_4\}, \{\tilde{\mathcal{I}}_5, a_5\}\} \to \{\{\{\tilde{\mathcal{I}}_4, \alpha_4^{(3)}\}, \{\tilde{\mathcal{I}}_5, \alpha_5^{(3)}\}\}, \{\{\{\mathcal{J}, \gamma^{(3)}\}\}\}\}$$

$$\Downarrow$$

$$\{\{\tilde{\mathcal{I}}_4, a_4\}\} \to \{\{\{\tilde{\mathcal{I}}_4, \alpha_4^{(3)}\}, \{\tilde{\mathcal{I}}_5, \alpha_5^{(3)}\}\}, \{\{\{\mathcal{J}, \gamma^{(3)}\}\}\}\Big|_{a_5 \to a_5^{\text{input}}}$$

Remark: Since we computed the local symbolic rule in the Gauß and Zürcher schemes, there are actually different trees that we can build:

- Purely Gauß: We apply recursively the local Gauß rules to the RMIs to build the tree
- Purely Zürcher: We apply recursively the local Zürcher rules to the RMIs to build the tree
- Combined Zürcher and Gauß: We apply recursively the local combined rules to the RMIs to build the tree

Remark: Since we computed the local symbolic rule in the Gauß and Zürcher schemes, there are actually different trees that we can build:

- Purely Gauß: We apply recursively the local Gauß rules to the RMIs to build the tree
- Purely Zürcher: We apply recursively the local Zürcher rules to the RMIs to build the tree
- Combined Zürcher and Gauß: We apply recursively the local combined rules to the RMIs to build the tree
- Mixed scheme: Given the list of Zürcher rules $L_Z = (r_1^Z, r_2^Z, \cdots, r_n^Z)$ and that of Gauß $L_G - (r_1^G, r_2^G, \cdots, r_n^G)$, we intertwine them in all the possible ways, i.e.:

$$(r_1^Z, r_2^Z, \cdots, r_n^Z), (r_1^G, r_2^Z, \cdots, r_n^Z), (r_1^Z, r_2^G, \cdots, r_n^Z), \cdots, (r_1^G, r_2^G, \cdots, r_n^G)$$

Then we construct a tree for each of these intertwined lists, replace the symbolic orders by the real ones and select the best tree (the lowest average correction)

Examples of system computations results:

(n; subsystems,BI)	BI orders (Carsten)	BI orders (NF)
$(3; 1; 4)$	$(1, 0, 0, 1)$	$(1, 0, 0, 1)$
$(48; 19; 7)$	$(5, 2, 4, 1, 1, 4, 0)$	$(3, 2, 3, 1, 1, 3, 0)$

Examples of system computations results:

(n; subsystems,BI)	BI orders (Carsten)	BI orders (NF)
$(3; 1; 4)$	$(1, 0, 0, 1)$	$(1, 0, 0, 1)$
$(48; 19; 7)$	$(5, 2, 4, 1, 1, 4, 0)$	$(3, 2, 3, 1, 1, 3, 0)$

- Main application: faster solving and speed up for the large momentum method

To do:

- Intensive testing of the algorithm to debug the problems and make sure that it's really giving the optimal solution!
- Speed up some (resilient) parts of the algorithm: uncoupling, converting `OreSys` output into a usable form,...
- In relation with the previous point, use parallelization to make computations faster

Examples of system computations results:

(n; subsystems,BI)	BI orders (Carsten)	BI orders (NF)
$(3; 1; 4)$	$(1, 0, 0, 1)$	$(1, 0, 0, 1)$
$(48; 19; 7)$	$(5, 2, 4, 1, 1, 4, 0)$	$(3, 2, 3, 1, 1, 3, 0)$

- Main application: faster solving and speed up for the large momentum method

To do:

- Intensive testing of the algorithm to debug the problems and make sure that it's really giving the optimal solution!
- Speed up some (resilient) parts of the algorithm: uncoupling, converting `OreSys` output into a usable form,...
- In relation with the previous point, use parallelization to make computations faster

Thank you!

Inspired by and built on I


J. Ablinger et al. *A toolbox to solve coupled systems of differential and difference equations*. 2016. arXiv: 1601.01856 [cs.SC].


J. Ablinger et al. *Algorithms to solve coupled systems of differential equations in terms of power series*. 2016. arXiv: 1608.05376 [cs.SC].


J. Ablinger et al. "Automated solution of first order factorizable systems of differential equations in one variable". In: *Nuclear Physics B* 939 (Dec. 2018). DOI: 10.1016/j.nuclphysb.2018.12.010.


J. Ablinger et al. "Heavy quark form factors at two loops". In: *Physical Review D* 97.9 (May 2018). ISSN: 2470-0029. DOI: 10.1103/physrevd.97.094022.


S.A. Abramov and M. Petkovšek. "D'Alembertian Solutions of Linear Differential and Difference Equations". In: (1994), pp. 169–174.


J. Blümlein, A. De Freitas, and C. Schneider. *Recent Symbolic Summation Methods to Solve Coupled Systems of Differential and Difference Equations*. 2014. arXiv: 1407.2537 [cs.SC].

Inspired by and built on II

J. Blümlein, A. De Freitas, and C. Schneider. "Recent Symbolic Summation Methods to Solve Coupled Systems of Differential and Difference Equations". In: *PoS* LL2014 (2014), p. 017. DOI: 10.22323/1.211.0017.

J. Blümlein and C. Schneider. "The method of arbitrarily large moments to calculate single scale processes in quantum field theory". In: *Physics Letters B* 771 (Aug. 2017), pp. 31–36. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2017.05.001.

J. Blümlein et al. "A symbolic summation approach to Feynman integral calculus". In: *Journal of Symbolic Computation* 47.10 (Oct. 2012), pp. 1267–1289. ISSN: 0747-7171. DOI: 10.1016/j.jsc.2011.12.044.

S. Gerhold. "Uncoupling systems of linear operator equations". PhD thesis. RISC, J. Kepler University, Linz, Feb. 2002.

E. Remiddi. "Differential equations for Feynman graph amplitudes". In: *Nuovo Cim. A* 110 (1997), pp. 1435–1452. arXiv: hep-th/9711188.

Inspired by and built on III

C. Schneider, J. Blümlein, and P. Marquard. "A refined machinery to calculate large moments from coupled systems of linear differential equations". In: *PoS* RADCOR2019 (2019), p. 078. DOI: 10.22323/1.375.0078.

B. Zürcher. "Rationale Normalformen von pseudo-linearen Abbildungen". MA thesis. ETH Zürich, 1994.