Computing identifiable functions of parameters for ODE models

Alexey Ovchinnikov
Queens College and the CUNY Graduate Center

This is joint work with Anand Pillay, Gleb Pogudin, and Thomas Scanlon

Implementation is available here:
`https://github.com/pogudingleb/AllIdentifiableFunctions`

Plan

- Intro to identifiability

Plan

- Intro to identifiability
- Approach via input-output equations and subtleties

Plan

- Intro to identifiability
- Approach via input-output equations and subtleties
- Our solution

Intro to identifiability

What is identifiability: toy examples

Example

In the model described by $\dot{x} = kx$

- x can measured in an experiment
  and, therefore, its derivatives can be estimated,
- k is an unknown scalar parameter.

What is identifiability: toy examples

Example

In the model described by $\dot{x} = kx$

- x can measured in an experiment
  and, therefore, its derivatives can be estimated,
- k is an unknown scalar parameter.

$$k = \frac{\dot{x}}{x} \quad \Longrightarrow \quad k \text{ is identifiable.}$$

What is identifiability: toy examples

Example

In the model described by $\dot{x} = kx$

- x can measured in an experiment
  and, therefore, its derivatives can be estimated,
- k is an unknown scalar parameter.

$$k = \frac{\dot{x}}{x} \quad \Longrightarrow \quad k \text{ is identifiable.}$$

Example

In the model described by $\dot{x} = x + k_1 + k_2$

- x can measured in an experiment
  and, therefore, its derivatives can be estimated,
- k_1 and k_2 are unknown scalar parameters.

What is identifiability: toy examples

Example

In the model described by $\dot{x} = kx$

- x can measured in an experiment
  and, therefore, its derivatives can be estimated,
- k is an unknown scalar parameter.

$$k = \frac{\dot{x}}{x} \quad \Longrightarrow \quad k \text{ is identifiable.}$$

Example

In the model described by $\dot{x} = x + k_1 + k_2$

- x can measured in an experiment
  and, therefore, its derivatives can be estimated,
- k_1 and k_2 are unknown scalar parameters.

Impossible to find k_1 and k_2 $\quad \Longrightarrow \quad$ k_1 and k_2 are non-identifiable.

What is identifiability: toy examples

Example

In the model described by $\dot{x} = kx$

- x can measured in an experiment
  and, therefore, its derivatives can be estimated,
- k is an unknown scalar parameter.

$$k = \frac{\dot{x}}{x} \quad \Longrightarrow \quad k \text{ is identifiable.}$$

Example

In the model described by $\dot{x} = x + k_1 + k_2$

- x can measured in an experiment
  and, therefore, its derivatives can be estimated,
- k_1 and k_2 are unknown scalar parameters.

$$\text{Impossible to find } k_1 \text{ and } k_2 \quad \Longrightarrow \quad k_1 \text{ and } k_2 \text{ are non-identifiable.}$$

But $k_1 + k_2$ is identifiable.

What is identifiability: toy examples

Example

In the model described by $\dot{x} = kx$

- x can measured in an experiment
  and, therefore, its derivatives can be estimated,
- k is an unknown scalar parameter.

$$k = \frac{\dot{x}}{x} \implies k \text{ is identifiable.}$$

Example

In the model described by $\dot{x} = x + k_1 + k_2$

- x can measured in an experiment
  and, therefore, its derivatives can be estimated,
- k_1 and k_2 are unknown scalar parameters.

$$\text{Impossible to find } k_1 \text{ and } k_2 \implies k_1 \text{ and } k_2 \text{ are non-identifiable.}$$

But $k_1 + k_2$ is identifiable. How to detect this and use to reparametrize?

Identifiability: Motivation

Common problem: more than one parameter value fits the data.

Identifiability: Motivation

Common problem: more than one parameter value fits the data.

There are different options

Cause		**Remedy**
Noisy data	$\Longrightarrow$	More measurements or better equipment

Identifiability: Motivation

Common problem: more than one parameter value fits the data.

There are different options

Cause		**Remedy**
Noisy data	$\Longrightarrow$	More measurements or better equipment
Non-identfiability	$\Longrightarrow$	Another model or new equipment

Identifiability: Motivation

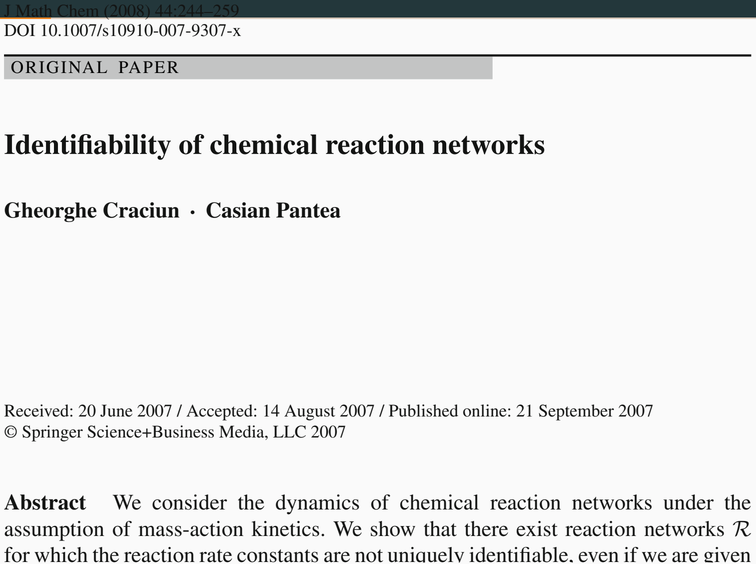
Common problem: more than one parameter value fits the data.

There are different options

Cause		**Remedy**
Noisy data	$\Longrightarrow$	More measurements or better equipment
Non-identfiability	$\Longrightarrow$	Another model or new equipment

Verifying identifiabilty allows a modeller to find the cause and choose the correct remedy.

Is this really an issue?


J Math Chem (2008) 44:244–259
DOI 10.1007/s10910-007-9307-x

ORIGINAL PAPER

Identifiability of chemical reaction networks

Gheorghe Craciun · Casian Pantea

Received: 20 June 2007 / Accepted: 14 August 2007 / Published online: 21 September 2007
© Springer Science+Business Media, LLC 2007

Abstract We consider the dynamics of chemical reaction networks under the assumption of mass-action kinetics. We show that there exist reaction networks $\mathcal{R}$ for which the reaction rate constants are not uniquely identifiable, even if we are given

Is this really an issue?

SIAM REVIEW
Vol. 53, No. 1, pp. 3–39

On Identifiability of Nonlinear ODE Models and Applications in Viral Dynamics*

Hongyu Miao[†]
Xiaohua Xia[‡]
Alan S. Perelson[§]
Hulin Wu[†]

Abstract. Ordinary differential equations (ODEs) are a powerful tool for modeling dynamic processes with wide applications in a variety of scientific fields. Over the last two decades, ODEs have also emerged as a prevailing tool in various biomedical research fields, especially in infectious disease modeling. In practice, it is important and necessary to determine unknown parameters in ODE models based on experimental data. Identifiability analysis is the first step in determining unknown parameters in ODE models and such analysis techniques for nonlinear ODE models are still under development. In this article, we review identifiability analysis methodologies for nonlinear ODE models developed in the past couple of decades, including structural identifiability analysis, practical identifiability

Is this really an issue?

Animal (2018), **12:4**, pp 701–712 © The Animal Consortium 2017
doi:10.1017/S1751731117002774

animal

Review: To be or not to be an identifiable model. Is this a relevant question in animal science modelling?

R. Muñoz-Tamayo[1†], L. Puillet[1], J. B. Daniel[1,2], D. Sauvant[1], O. Martin[1], M. Taghipoor[3] and P. Blavy[1]

[1]UMR Modélisation Systémique Appliquée aux Ruminants, INRA, AgroParisTech, Université Paris-Saclay, 75005 Paris, France; [2]Trouw Nutrition R&D, P.O. Box 220, 5830 AE Boxmeer, The Netherlands; [3]PEGASE, AgroCampus Ouest, INRA, 35590 Saint-Gilles, France

(Received 4 May 2017; Accepted 24 September 2017; First published online 3 November 2017)

What is a good (useful) mathematical model in animal science? For models constructed for prediction purposes, the question of model adequacy (usefulness) has been traditionally tackled by statistical analysis applied to observed experimental data relative to model-predicted variables. However, little attention has been paid to analytic tools that exploit the mathematical properties of the model equations. For example, in the context of model calibration, before attempting a numerical estimation of the model parameters, we might want to know if we have any chance of success in estimating a unique best value of the model parameters from available measurements. This question of uniqueness is referred to as structural identifiability; a mathematical property that is defined on the sole basis of the model structure within a hypothetical ideal experiment determined by a setting of model inputs (stimuli) and observable variables (measurements). Structural identifiability analysis applied to dynamic models described by

Relaxation of the problem: local identifiability

On this slide

- x can be measured in an experiment
  and, therefore, its derivatives can be estimated
- k_1 and k_2 are unknown scalar parameters

Equation	What happens	Identifiable?
$\dot{x} = x + k_1$	$k_1 = \dot{x} - x$	**YES**
$\dot{x} = x + {k_1}^2$	$k_1 = \pm\sqrt{\dot{x} - x}$	**NO**
$\dot{x} = x + k_1 + k_2$	Infinitely many values for k_1 and k_2	**NO**

Relaxation of the problem: local identifiability

On this slide

- x can be measured in an experiment
  and, therefore, its derivatives can be estimated
- k_1 and k_2 are unknown scalar parameters

Equation	What happens	Identifiable?
$\dot{x} = x + k_1$	$k_1 = \dot{x} - x$	**Globally**
$\dot{x} = x + {k_1}^2$	$k_1 = \pm\sqrt{\dot{x} - x}$	**Locally**
$\dot{x} = x + k_1 + k_2$	Infinitely many values for k_1 and k_2	**NO**

Local identifiability: state of the art

- **Jacobian test**: *Hermann and Krener* (1977)

Local identifiability: state of the art

- **Jacobian test**: *Hermann and Krener* (1977)

- **Efficient software**:
  - ObservabilityTest (2002)
  - IdentifiabilityAnalysis (2012)
  - STRIKE-GOLDD (2016)

Local identifiability: state of the art

- **Jacobian test**: *Hermann and Krener* (1977)
- **Efficient software**:
  - ObservabilityTest (2002)
  - IdentifiabilityAnalysis (2012)
  - STRIKE-GOLDD (2016)
- **Criteria for systems of special form:**
  - *Meshkat, Sullivant, Eisenberg* (2015)
  - *Meshkat, Rosen, Sullivant* (2016)
  - *Baaijens, Draisma* (2016)
  - *Gross, Meshkat, Shiu* (2018)

The importance of being globally identifiable

- Local identifiability does not guarantee the uniqieness of the parameter value.

The importance of being globally identifiable

- Local identifiability does not guarantee the uniqieness of the parameter value.

- Lack of global identifiability is hard to detect using numeric methods.

The importance of being globally identifiable

- Local identifiability does not guarantee the uniqieness of the parameter value.
- Lack of global identifiability is hard to detect using numeric methods.
- It happens!

It happens: epidemiology (SEIR model)

$$\begin{cases} S' = -\beta \frac{SI}{N}, \\ E' = \beta \frac{SI}{N} - \eta E, \\ I' = \eta E - \alpha I, \\ R' = \alpha R, \\ N = S + E + I + R, \end{cases}$$

Susceptible

$\downarrow$

Exposed

$\downarrow$

Infectious

$\downarrow$

Recovered

It happens: epidemiology (SEIR model)

$$\begin{cases} S' = -\beta \frac{SI}{N}, \\ E' = \beta \frac{SI}{N} - \eta E, \\ I' = \eta E - \alpha I, \\ N' = 0, \\ \\ \end{cases}$$

Susceptible
↓
Exposed
↓
Infectious
↓
Recovered

It happens: epidemiology (SEIR model)

$$\begin{cases} S' = -\beta \frac{SI}{N}, \\ E' = \beta \frac{SI}{N} - \eta E, \\ I' = \eta E - \alpha I, \\ N' = 0, \\ y_1 = N, \\ y_2 = \kappa I. \end{cases}$$

Susceptible

$\downarrow$

Exposed

$\downarrow$

Infectious

$\downarrow$

Recovered

It happens: epidemiology (SEIR model)

$$\begin{cases} S' = -\beta \frac{SI}{N}, \\ E' = \beta \frac{SI}{N} - \eta E, \\ I' = \eta E - \alpha I, \\ N' = 0, \\ y_1 = N, \\ y_2 = \kappa I. \end{cases}$$

Susceptible
↓
Exposed
↓
Infectious
↓
Recovered

Turns out:

Only locally identifiable: α, η,

Nonidentifiable: β, κ.

It happens: epidemiology (SEIR model)

$$\begin{cases} S' = -\beta \frac{SI}{N}, \\ E' = \beta \frac{SI}{N} - \eta E, \\ I' = \eta E - \alpha I, \\ N' = 0, \\ y_1 = N, \\ y_2 = \kappa I. \end{cases}$$

Susceptible
↓
Exposed
↓
Infectious
↓
Recovered

Turns out:

Only locally identifiable: α, η,

Nonidentifiable: β, κ.

Furthermore:

An unordered pair $\{\alpha, \eta\}$ is identifiable, so $\alpha + \eta$ and $\alpha\eta$ are identifiable.

Global identifiability: state of the art

Taylor series method	Theory: *Ponjanpalo*, 1978 Software: GenSSI 2.0, 2017 Termination criterion only for special cases
Differential elimination for parameters	Theory: *Diop, Fliess, Ljung, Glad*, 1993 Tackles only small examples
Input-output equations	Theory: *Ollivier*, 1990 Software: DAISY, 2007; COMBOS, 2014 In a few minutes!
Prolongations + symbolc sampling	Theory: *Hong, Ovchinnikov, Pogudin, Yap*, 2020 Software: SIAN, 2019

Definition of identifiability in algebra

Differential fields, polynomials, and ideals

- Differential ring/field K is ring/field with a derivation $'$:
  $\mathbb{C}(x)$ with derivation d/dx.

Differential fields, polynomials, and ideals

- Differential ring/field K is ring/field with a derivation $'$: $\mathbb{C}(x)$ with derivation d/dx.
- Differential polynomials:

$$K\{x, y, z\} = K[x, y, z, x', y', z', \ldots].$$

Differential fields, polynomials, and ideals

- Differential ring/field K is ring/field with a derivation $'$: $\mathbb{C}(x)$ with derivation d/dx.
- Differential polynomials:

$$K\{x, y, z\} = K[x, y, z, x', y', z', \ldots].$$

- Differential ideal I in differential ring R:

$$a \in I \implies a' \in I.$$

Differential fields, polynomials, and ideals

- Differential ring/field K is ring/field with a derivation $'$:
  $\mathbb{C}(x)$ with derivation d/dx.
- Differential polynomials:

$$K\{x, y, z\} = K[x, y, z, x', y', z', \ldots].$$

- Differential ideal I in differential ring R:

$$a \in I \implies a' \in I.$$

- Notation: smallest differential ideal in R containing a, b, c is $[a, b, c]$.

Differential fields, polynomials, and ideals

- Differential ring/field K is ring/field with a derivation $'$: $\mathbb{C}(x)$ with derivation d/dx.
- Differential polynomials:

$$K\{x, y, z\} = K[x, y, z, x', y', z', \ldots].$$

- Differential ideal I in differential ring R:

$$a \in I \implies a' \in I.$$

- Notation: smallest differential ideal in R containing a, b, c is $[a, b, c]$.
- Notation: smallest differential field containing $\mathbb{C}$ and a, b, c is $\mathbb{C}\langle a, b, c\rangle$.

Generic solution

Input

System

$$\begin{cases} x' = f(x, \mu), \\ y = g(x, \mu), \end{cases} \tag{1}$$

where

- x are unknown *state variables*;
- μ are unknown scalar *parameters*;
- y are *outputs* measured in experiment.

Generic solution

Input

System

$$\begin{cases} x' = f(x, \mu), \\ y = g(x, \mu), \end{cases} \tag{1}$$

where

- x are unknown *state variables*;
- μ are unknown scalar *parameters*;
- y are *outputs* measured in experiment.

A tuple (x^*, y^*) from a differential field $k \supset \mathbb{C}(\mu)$ is *a generic solution* of (2) if, for every differential polynomial $P \in \mathbb{C}(\mu)\{x, y\}$, we have

$$P(x^*, y^*) = 0 \iff P \in [x' - f(x, \mu), y - g(x, \mu)].$$

Generic solution

Input

System

$$\begin{cases} \mathrm{x}' = \mathrm{f}(\mathrm{x}, \mu), \\ \mathrm{y} = \mathrm{g}(\mathrm{x}, \mu), \end{cases} \tag{1}$$

where

- x are unknown *state variables*;
- μ are unknown scalar *parameters*;
- y are *outputs* measured in experiment.

A tuple (x^*, y^*) from a differential field $k \supset \mathbb{C}(\mu)$ is *a generic solution* of (2) if, for every differential polynomial $P \in \mathbb{C}(\mu)\{x, y\}$, we have

$$P(x^*, y^*) = 0 \iff P \in [x' - f(x, \mu), y - g(x, \mu)].$$

Example: $(0, 0)$ is not generic but (e^t, e^t) is generic for $x' = x, y = x$.

Definition of identifiability

Input

System

$$\begin{cases} x' = f(x, \mu), \\ y = g(x, \mu), \end{cases} \tag{2}$$

where

- x are unknown *state variables*;
- μ are unknown scalar *parameters*;
- y are *outputs* measured in experiment.

Definition of identifiability

Input

System

$$\begin{cases} x' = f(x, \mu), \\ y = g(x, \mu), \end{cases} \tag{2}$$

where

- x are unknown *state variables*;
- μ are unknown scalar *parameters*;
- y are *outputs* measured in experiment.

A rational function $h \in \mathbb{C}(\mu)$ is *globally (resp., locally)* identifiable if, for every generic solution (x^*, y^*) of (2),

$$h \in \mathbb{C}\langle y^* \rangle$$

(resp., h is algebraic over $\mathbb{C}\langle y^* \rangle$).

Definition of identifiability

Input

System

$$\begin{cases} x' = f(x, \mu), \\ y = g(x, \mu), \end{cases} \qquad (2)$$

where

- x are unknown *state variables*;
- μ are unknown scalar *parameters*;
- y are *outputs* measured in experiment.

A rational function $h \in \mathbb{C}(\mu)$ is *globally (resp., locally)* identifiable if, for every generic solution (x^*, y^*) of (2),

$$h \in \mathbb{C}\langle y^* \rangle$$

(resp., h is algebraic over $\mathbb{C}\langle y^* \rangle$).

Example: $x' = x + \mu_1 + \mu_2, y = x$. Then $h = \mu_1 + \mu_2 = y' - y$ is identifiable.

Input-output equations

Specification: what we are after

Input

System

$$\begin{cases} x' = f(x, \mu), \\ y = g(x, \mu), \end{cases}$$

where

- x are unknown *state variables*;
- μ are unknown scalar *parameters*;
- y are *outputs* measured in experiment.

Specification: what we are after

Input

System

$$\begin{cases} x' = f(x, \mu), \\ y = g(x, \mu), \end{cases}$$

where

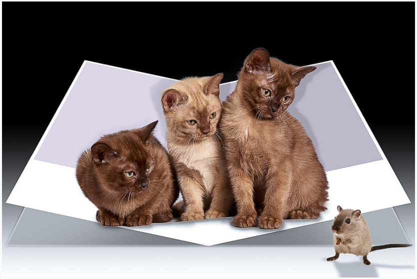
- x are unknown *state variables*;
- μ are unknown scalar *parameters*;
- y are *outputs* measured in experiment.

Output

Generators of the field of identifiable rational functions in μ.

Running example: predator-prey model

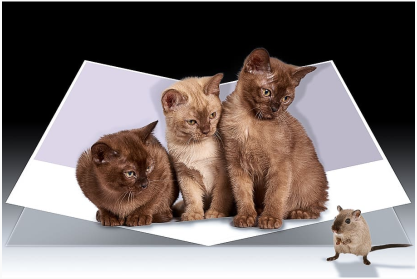
$$\begin{cases} x_1' = k_1 x_1 - k_2 x_1 x_2, \\ x_2' = -k_3 x_2 + k_4 x_1 x_2, \\ y = x_1. \end{cases}$$



- x_1 - prey
- x_2 - predators

Running example: predator-prey model

$$\begin{cases} x_1' = k_1 x_1 - k_2 x_1 x_2, \\ x_2' = -k_3 x_2 + k_4 x_1 x_2, \\ y = x_1. \end{cases}$$



- x_1 - prey
- x_2 - predators

Globally identifiable: k_1, k_3, k_4
Nonidentifiable: k_2
Identifiable functions: $\mathbb{C}(k_1, k_3, k_4)$.

Step 1: Eliminate

Idea: we cannot measure $x_2 \implies$ let us eliminate it!

Step 1: Eliminate

Idea: we cannot measure $x_2 \implies$ let us eliminate it!

$$\begin{cases} x_1' = k_1 x_1 - k_2 x_1 x_2 \\ x_2' = -k_3 x_2 + k_4 x_1 x_2 \\ y = x_1 \end{cases} \implies yy'' - y'^2 - k_4 y^2 y' - k_3 y y' + k_1 k_4 y^3 - k_1 k_3 y^2 = 0$$

Step 1: Eliminate

Idea: we cannot measure $x_2 \implies$ let us eliminate it!

$$\begin{cases} x_1' = k_1 x_1 - k_2 x_1 x_2 \\ x_2' = -k_3 x_2 + k_4 x_1 x_2 \\ y = x_1 \end{cases} \implies yy'' - y'^2 - k_4 y^2 y' - k_3 yy' + k_1 k_4 y^3 - k_1 k_3 y^2 = 0$$

Input-output equation - the “minimal” differential equation for y with coefficients in the parameters.

Step 2: Extract coefficients

Idea: Differentiate the minimal equation $\implies$ linear equations in the coefficients

$$yy'' - y'^2 - k_4 y^2 y' - k_3 yy' + k_1 k_4 y^3 - k_1 k_3 y^2 = 0$$

Step 2: Extract coefficients

Idea: Differentiate the minimal equation $\implies$ linear equations in the coefficients

$$yy'' - y'^2 - k_4 y^2 y' - k_3 yy' + k_1 k_4 y^3 - k_1 k_3 y^2 = 0$$

Wronskian:

$$yy'' - y'^2 = k_4 y^2 y' + k_3 yy' - k_1 k_4 y^3 + k_1 k_3 y^2$$
$$(yy'' - y'^2)' = k_4 (y^2 y')' + k_3 (yy')' - k_1 k_4 (y^3)' + k_1 k_3 (y^2)'$$
$$(yy'' - y'^2)'' = k_4 (y^2 y')'' + k_3 (yy')'' - k_1 k_4 (y^3)'' + k_1 k_3 (y^2)''$$
$$(yy'' - y'^2)''' = k_4 (y^2 y')''' + k_3 (yy')''' - k_1 k_4 (y^3)''' + k_1 k_3 (y^2)'''$$

Step 2: Extract coefficients

Idea: Differentiate the minimal equation $\Longrightarrow$ linear equations in the coefficients

$$yy'' - y'^2 - k_4 y^2 y' - k_3 yy' + k_1 k_4 y^3 - k_1 k_3 y^2 = 0$$

Wronskian:

$$\begin{aligned}
yy'' - y'^2 &= k_4 y^2 y' + k_3 yy' - k_1 k_4 y^3 + k_1 k_3 y^2 \\
(yy'' - y'^2)' &= k_4 (y^2 y')' + k_3 (yy')' - k_1 k_4 (y^3)' + k_1 k_3 (y^2)' \\
(yy'' - y'^2)'' &= k_4 (y^2 y')'' + k_3 (yy')'' - k_1 k_4 (y^3)'' + k_1 k_3 (y^2)'' \\
(yy'' - y'^2)''' &= k_4 (y^2 y')''' + k_3 (yy')''' - k_1 k_4 (y^3)''' + k_1 k_3 (y^2)'''
\end{aligned}$$

Assume nonsingular Wronskian. Then one can prove:

$$\text{identifiable} \iff \text{rational in } k_4, k_3, k_1 k_4, k_1 k_3$$

Step 2: Extract coefficients

Idea: Differentiate the minimal equation $\Longrightarrow$ linear equations in the coefficients

$$yy'' - y'^2 - k_4 y^2 y' - k_3 yy' + k_1 k_4 y^3 - k_1 k_3 y^2 = 0$$

Wronskian:

$$\begin{aligned}
yy'' - y'^2 &= k_4 y^2 y' + k_3 yy' - k_1 k_4 y^3 + k_1 k_3 y^2 \\
(yy'' - y'^2)' &= k_4 (y^2 y')' + k_3 (yy')' - k_1 k_4 (y^3)' + k_1 k_3 (y^2)' \\
(yy'' - y'^2)'' &= k_4 (y^2 y')'' + k_3 (yy')'' - k_1 k_4 (y^3)'' + k_1 k_3 (y^2)'' \\
(yy'' - y'^2)''' &= k_4 (y^2 y')''' + k_3 (yy')''' - k_1 k_4 (y^3)''' + k_1 k_3 (y^2)'''
\end{aligned}$$

Assume nonsingular Wronskian. Then one can prove:

$$\text{identifiable} \iff \text{rational in } k_4, k_3, k_1 k_4, k_1 k_3$$

Remark

- Assumption is not always true

Subtlety: the assumption does not always hold

Subtlety: the assumption does not always hold

Not yet an example (twisted harmonic oscillator)

$$\begin{cases} x_1' = (\omega + \alpha) x_2, \\ x_2' = -\omega x_1, \\ y = x_2 \end{cases}$$

Subtlety: the assumption does not always hold

Not yet an example (twisted harmonic oscillator)

$$\begin{cases} x_1' = (\omega + \alpha) x_2, \\ x_2' = -\omega x_1, \\ y = x_2 \end{cases} \implies y'' + \omega(\omega + \alpha) y = 0$$

Subtlety: the assumption does not always hold

Not yet an example (twisted harmonic oscillator)

$$\begin{cases} x_1' = (\omega + \alpha) x_2, \\ x_2' = -\omega x_1, \\ y = x_2 \end{cases} \implies y'' + \omega(\omega + \alpha) y = 0$$

Example

Assume that α is known

$$\begin{cases} x_1' = (\omega + x_3) x_2, \\ x_2' = -\omega x_1, \\ x_3' = 0, \\ y_1 = x_2,\; y_2 = x_3 \end{cases}$$

Subtlety: the assumption does not always hold

Not yet an example (twisted harmonic oscillator)

$$\begin{cases} x_1' = (\omega + \alpha) x_2, \\ x_2' = -\omega x_1, \\ y = x_2 \end{cases} \implies y'' + \omega(\omega + \alpha) y = 0$$

Example

Assume that α is known

$$\begin{cases} x_1' = (\omega + x_3) x_2, \\ x_2' = -\omega x_1, \\ x_3' = 0, \\ y_1 = x_2, \; y_2 = x_3 \end{cases} \implies y_1'' + \omega^2 y_1 + \omega y_1 y_2 = 0, \; y_2' = 0$$

Subtlety: the assumption does not always hold

Not yet an example (twisted harmonic oscillator)

$$\begin{cases} x_1' = (\omega + \alpha)x_2, \\ x_2' = -\omega x_1, \\ y = x_2 \end{cases} \implies y'' + \omega(\omega + \alpha)y = 0$$

Example

Assume that α is known

$$\begin{cases} x_1' = (\omega + x_3)x_2, \\ x_2' = -\omega x_1, \\ x_3' = 0, \\ y_1 = x_2, \ y_2 = x_3 \end{cases} \implies \begin{array}{l} y_1'' + \omega^2 y_1 + \omega y_1 y_2 = 0, \quad y_2' = 0 \\ y_1''' + \omega^2 y_1' + \omega(y_1 y_2)' = 0 \end{array}$$

Subtlety: the assumption does not always hold

Not yet an example (twisted harmonic oscillator)

$$\begin{cases} x_1' = (\omega + \alpha)x_2, \\ x_2' = -\omega x_1, \\ y = x_2 \end{cases} \implies y'' + \omega(\omega + \alpha)y = 0$$

Example

Assume that α is known

$$\begin{cases} x_1' = (\omega + x_3)x_2, \\ x_2' = -\omega x_1, \\ x_3' = 0, \\ y_1 = x_2,\ y_2 = x_3 \end{cases} \implies \begin{array}{l} y_1'' + \omega^2 y_1 + \omega y_1 y_2 = 0, \\ y_1''' + \omega^2 y_1' + \omega y_1' y_2 = 0 \end{array}$$

Determinant of the Wronskian is $y_1 y_1' y_2 - y_1 y_2 y_1' = 0$.
Only $\omega(\omega + \alpha), \alpha$ known $\implies$ quadratic equation in ω.

Why do we care about this method then?

Why do we care about this method then?

- Used in practice (software: DAISY, COMBOS)

Why do we care about this method then?

- Used in practice (software: DAISY, COMBOS)
- If the assumption is true, finds all identifiable functions

Our algorithm

Algorithm Computing all identifiable functions

Input System $\Sigma = \begin{cases} x' = f(x, \mu) \\ y = g(x, \mu) \end{cases}$

Our algorithm

Algorithm Computing all identifiable functions

Input System $\Sigma = \begin{cases} x' = f(x, \mu) \\ y = g(x, \mu) \end{cases}$

Output Generators of the field of identifiable functions of Σ

Our algorithm

Algorithm Computing all identifiable functions

Input System $\Sigma = \begin{cases} x' = f(x, \mu) \\ y = g(x, \mu) \end{cases}$

Output Generators of the field of identifiable functions of Σ

1. Compute a set $\bar{p}$ of input-output equations of Σ (differential alg.).

Our algorithm

Algorithm Computing all identifiable functions

Input System $\Sigma = \begin{cases} \mathsf{x}' = \mathsf{f}(\mathsf{x}, \mu) \\ \mathsf{y} = \mathsf{g}(\mathsf{x}, \mu) \end{cases}$

Output Generators of the field of identifiable functions of Σ

1. Compute a set $\bar{p}$ of input-output equations of Σ (differential alg.).
2. For each $p \in \bar{p}$, compute W_p: compute the Wronskian of the monomials of p and apply reduction modulo the equations of Σ.

Our algorithm

Algorithm Computing all identifiable functions

Input System $\Sigma = \begin{cases} \mathsf{x}' = \mathsf{f}(\mathsf{x}, \mu) \\ \mathsf{y} = \mathsf{g}(\mathsf{x}, \mu) \end{cases}$

Output Generators of the field of identifiable functions of Σ

1. Compute a set $\bar{p}$ of input-output equations of Σ (differential alg.).
2. For each $p \in \bar{p}$, compute W_p: compute the Wronskian of the monomials of p and apply reduction modulo the equations of Σ.
3. For each $p \in \bar{p}$, calculate the reduced row echelon form of the matrix W_p and let $F(\bar{p})$ be the field generated over $\mathbb{C}$ by all non-leading coefficients of all matrices W_p.

Our algorithm

Algorithm Computing all identifiable functions

Input System $\Sigma = \begin{cases} \mathrm{x}' = \mathrm{f}(\mathrm{x}, \mu) \\ \mathrm{y} = \mathrm{g}(\mathrm{x}, \mu) \end{cases}$

Output Generators of the field of identifiable functions of Σ

1. Compute a set $\bar{p}$ of input-output equations of Σ (differential alg.).
2. For each $p \in \bar{p}$, compute W_p: compute the Wronskian of the monomials of p and apply reduction modulo the equations of Σ.
3. For each $p \in \bar{p}$, calculate the reduced row echelon form of the matrix W_p and let $F(\bar{p})$ be the field generated over $\mathbb{C}$ by all non-leading coefficients of all matrices W_p.
4. Find generators of $\mathbb{C}(\mu) \cap F(\bar{p})$.

Our algorithm

Algorithm Computing all identifiable functions

Input System $\Sigma = \begin{cases} \mathrm{x}' = \mathrm{f}(\mathrm{x}, \mu) \\ \mathrm{y} = \mathrm{g}(\mathrm{x}, \mu) \end{cases}$

Output Generators of the field of identifiable functions of Σ

1. Compute a set $\bar{p}$ of input-output equations of Σ (differential alg.).
2. For each $p \in \bar{p}$, compute W_p: compute the Wronskian of the monomials of p and apply reduction modulo the equations of Σ.
3. For each $p \in \bar{p}$, calculate the reduced row echelon form of the matrix W_p and let $F(\bar{p})$ be the field generated over $\mathbb{C}$ by all non-leading coefficients of all matrices W_p.
4. Find generators of $\mathbb{C}(\mu) \cap F(\bar{p})$. Return these generators.

Implementation is available here:

`https://github.com/pogudingleb/AllIdentifiableFunctions`

Our algorithm: example

$$\Sigma = \begin{cases} x' = 0 \\ y_1 = ax + b \\ y_2 = x \end{cases}$$

Our algorithm: example

$$\Sigma = \begin{cases} x' = 0 \\ y_1 = ax + b \\ y_2 = x \end{cases}$$

1. We eliminate x and find the following input-output equations:

$$y_1 - ay_2 - b = 0,\ y_2' = 0.$$

Our algorithm: example

$$\Sigma = \begin{cases} x' = 0 \\ y_1 = ax + b \\ y_2 = x \end{cases}$$

1. We eliminate x and find the following input-output equations:

$$y_1 - ay_2 - b = 0,\ y_2' = 0.$$

   Therefore, $\bar{p} = (p_1, p_2)$, where $p_1 = y_1 - ay_2 - b$ and $p_2 = y_2'$.

Our algorithm: example

$$\Sigma = \begin{cases} x' = 0 \\ y_1 = ax + b \\ y_2 = x \end{cases}$$

1. We eliminate x and find the following input-output equations:

$$y_1 - ay_2 - b = 0,\ y_2' = 0.$$

   Therefore, $\bar{p} = (p_1, p_2)$, where $p_1 = y_1 - ay_2 - b$ and $p_2 = y_2'$.

2.

$$W_{p_1} = \begin{pmatrix} 1 & y_1 & y_2 \\ 0 & y_1' & y_2' \\ 0 & y_1'' & y_2'' \end{pmatrix} \mod \Sigma = \begin{pmatrix} 1 & ax+b & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Our algorithm: example

$$\Sigma = \begin{cases} x' = 0 \\ y_1 = ax + b \\ y_2 = x \end{cases}$$

1. We eliminate x and find the following input-output equations:

$$y_1 - ay_2 - b = 0,\ y_2' = 0.$$

   Therefore, $\bar{p} = (p_1, p_2)$, where $p_1 = y_1 - ay_2 - b$ and $p_2 = y_2'$.

2.

$$W_{p_1} = \begin{pmatrix} 1 & y_1 & y_2 \\ 0 & y_1' & y_2' \\ 0 & y_1'' & y_2'' \end{pmatrix} \mod \Sigma = \begin{pmatrix} 1 & ax+b & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$W_{p_2} = \begin{pmatrix} y_2' \end{pmatrix} \mod \Sigma = \begin{pmatrix} 0 \end{pmatrix}.$$

Our algorithm: example

$$\Sigma = \begin{cases} x' = 0 \\ y_1 = ax + b \\ y_2 = x \end{cases}$$

1. We eliminate x and find the following input-output equations:

$$y_1 - ay_2 - b = 0,\ y_2' = 0.$$

   Therefore, $\bar{p} = (p_1, p_2)$, where $p_1 = y_1 - ay_2 - b$ and $p_2 = y_2'$.
2.

$$W_{p_1} = \begin{pmatrix} 1 & y_1 & y_2 \\ 0 & y_1' & y_2' \\ 0 & y_1'' & y_2'' \end{pmatrix} \mod \Sigma = \begin{pmatrix} 1 & ax + b & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$W_{p_2} = \begin{pmatrix} y_2' \end{pmatrix} \mod \Sigma = \begin{pmatrix} 0 \end{pmatrix}.$$

3. The corresponding reduced row echelon forms are the same.
   Therefore, $F(\bar{p}) = \mathbb{C}(ax + b, x)$.

Our algorithm: example

$$\Sigma = \begin{cases} x' = 0 \\ y_1 = ax + b \\ y_2 = x \end{cases}$$

1. We eliminate x and find the following input-output equations:

$$y_1 - ay_2 - b = 0,\ y_2' = 0.$$

   Therefore, $\bar{p} = (p_1, p_2)$, where $p_1 = y_1 - ay_2 - b$ and $p_2 = y_2'$.

2.

$$W_{p_1} = \begin{pmatrix} 1 & y_1 & y_2 \\ 0 & y_1' & y_2' \\ 0 & y_1'' & y_2'' \end{pmatrix} \mod \Sigma = \begin{pmatrix} 1 & ax+b & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$W_{p_2} = \begin{pmatrix} y_2' \end{pmatrix} \mod \Sigma = \begin{pmatrix} 0 \end{pmatrix}.$$

3. The corresponding reduced row echelon forms are the same.
   Therefore, $F(\bar{p}) = \mathbb{C}(ax + b, x)$.
4. The field of identifiable functions is $\mathbb{C}(a, b) \cap \mathbb{C}(ax + b, x) = ?$.

Our algorithm: another example

$$\Sigma = \begin{cases} x_1' = (\omega + x_3)x_2, \\ x_2' = -\omega x_1, \\ x_3' = 0, \\ y_1 = x_2,\ y_2 = x_3 \end{cases}$$

Our algorithm: another example

$$\Sigma = \begin{cases} x_1' = (\omega + x_3)x_2, \\ x_2' = -\omega x_1, \\ x_3' = 0, \\ y_1 = x_2,\ y_2 = x_3 \end{cases}$$

1. We eliminate x_1, x_2, x_3 and find these input-output equations:

$$y_1'' + \omega^2 y_1 + \omega y_1 y_2 = 0,\ y_2' = 0.$$

Our algorithm: another example

$$\Sigma = \begin{cases} x_1' = (\omega + x_3)x_2, \\ x_2' = -\omega x_1, \\ x_3' = 0, \\ y_1 = x_2,\ y_2 = x_3 \end{cases}$$

1. We eliminate x_1, x_2, x_3 and find these input-output equations:

$$y_1'' + \omega^2 y_1 + \omega y_1 y_2 = 0,\ y_2' = 0.$$

Therefore, $\bar{p} = (p_1, p_2)$, where $p_1 = y_1'' + \omega^2 y_1 + \omega y_1 y_2,\ p_2 = y_2'$.

Our algorithm: another example

$$\Sigma = \begin{cases} x_1' = (\omega + x_3)x_2, \\ x_2' = -\omega x_1, \\ x_3' = 0, \\ y_1 = x_2,\ y_2 = x_3 \end{cases}$$

1. We eliminate x_1, x_2, x_3 and find these input-output equations:

$$y_1'' + \omega^2 y_1 + \omega y_1 y_2 = 0,\ y_2' = 0.$$

   Therefore, $\bar{p} = (p_1, p_2)$, where $p_1 = y_1'' + \omega^2 y_1 + \omega y_1 y_2,\ p_2 = y_2'$.

2.

$$W_{p_1} = \begin{pmatrix} y_1 & y_1'' & y_1 y_2 \\ y_1' & y_1''' & (y_1 y_2)' \\ y_1'' & y_1'''' & (y_1 y_2)'' \end{pmatrix} \bmod \Sigma = \begin{pmatrix} x_2 & -(\omega + x_3)\omega x_2 & x_2 x_3 \\ -\omega x_1 & x_1 \omega^2 (\omega + x_3) & -x_3 x_1 \omega \\ -(\omega + x_3)\omega x_2 & x_2 \omega^2 (\omega + x_3)^2 & -(\omega + x_3)\omega x_2 x_3 \end{pmatrix}$$

Our algorithm: another example

$$\Sigma = \begin{cases} x_1' = (\omega + x_3)x_2, \\ x_2' = -\omega x_1, \\ x_3' = 0, \\ y_1 = x_2,\ y_2 = x_3 \end{cases}$$

1. We eliminate x_1, x_2, x_3 and find these input-output equations:

$$y_1'' + \omega^2 y_1 + \omega y_1 y_2 = 0,\ y_2' = 0.$$

2. Therefore, $\bar{p} = (p_1, p_2)$, where $p_1 = y_1'' + \omega^2 y_1 + \omega y_1 y_2,\ p_2 = y_2'$.

$$W_{p_1} = \begin{pmatrix} y_1 & y_1'' & y_1 y_2 \\ y_1' & y_1''' & (y_1 y_2)' \\ y_1'' & y_1'''' & (y_1 y_2)'' \end{pmatrix} \bmod \Sigma = \begin{pmatrix} x_2 & -(\omega + x_3)\omega x_2 & x_2 x_3 \\ -\omega x_1 & x_1 \omega^2 (\omega + x_3) & -x_3 x_1 \omega \\ -(\omega + x_3)\omega x_2 & x_2 \omega^2 (\omega + x_3)^2 & -(\omega + x_3)\omega x_2 x_3 \end{pmatrix}$$

$$W_{p_2} = \begin{pmatrix} y_2' \end{pmatrix} \mod \Sigma = \begin{pmatrix} 0 \end{pmatrix}.$$

Our algorithm: another example

$$\Sigma = \begin{cases} x_1' = (\omega + x_3)x_2, \\ x_2' = -\omega x_1, \\ x_3' = 0, \\ y_1 = x_2,\ y_2 = x_3 \end{cases}$$

1. We eliminate x_1, x_2, x_3 and find these input-output equations:
$$y_1'' + \omega^2 y_1 + \omega y_1 y_2 = 0,\ y_2' = 0.$$
2. Therefore, $\bar{p} = (p_1, p_2)$, where $p_1 = y_1'' + \omega^2 y_1 + \omega y_1 y_2,\ p_2 = y_2'$.

$$W_{p_1} = \begin{pmatrix} y_1 & y_1'' & y_1 y_2 \\ y_1' & y_1''' & (y_1 y_2)' \\ y_1'' & y_1'''' & (y_1 y_2)'' \end{pmatrix} \bmod \Sigma = \begin{pmatrix} x_2 & -(\omega + x_3)\omega x_2 & x_2 x_3 \\ -\omega x_1 & x_1 \omega^2 (\omega + x_3) & -x_3 x_1 \omega \\ -(\omega + x_3)\omega x_2 & x_2 \omega^2 (\omega + x_3)^2 & -(\omega + x_3)\omega x_2 x_3 \end{pmatrix}$$

$$W_{p_2} = \begin{pmatrix} y_2' \end{pmatrix} \quad \bmod \Sigma = \begin{pmatrix} 0 \end{pmatrix}.$$

3. The corresponding reduced row echelon forms are

$$\begin{pmatrix} 1 & -(\omega + x_3)\omega & x_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 \end{pmatrix}$$

Our algorithm: another example

$$\Sigma = \begin{cases} x_1' = (\omega + x_3)x_2, \\ x_2' = -\omega x_1, \\ x_3' = 0, \\ y_1 = x_2,\ y_2 = x_3 \end{cases}$$

1. We eliminate x_1, x_2, x_3 and find these input-output equations:

$$y_1'' + \omega^2 y_1 + \omega y_1 y_2 = 0,\ y_2' = 0.$$

   Therefore, $\bar{p} = (p_1, p_2)$, where $p_1 = y_1'' + \omega^2 y_1 + \omega y_1 y_2,\ p_2 = y_2'$.

2. The corresponding reduced row echelon forms are

$$\begin{pmatrix} 1 & -(\omega + x_3)\omega & x_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 \end{pmatrix}$$

   . Therefore, $F(\bar{p}) = \mathbb{C}(\omega(\omega + x_3), x_3)$.

3. The field of identifiable functions is $\mathbb{C}(\omega) \cap \mathbb{C}(\omega(\omega + x_3), x_3) = ?$.

Intersection of fields: an attempt

ACM SIGSAM Bulletin Volume 32, Issue 2, p. 62
(from abstract of ISSAC 1998 poster):

Computing the Intersection of Finitely Generated Fields

JÖRN MÜLLER-QUADE and THOMAS BETH

Institut für Algorithmen und Kognitive Systeme
Fakultät für Informatik, Universität Karlsruhe, Germany.

For the problem of computing the intersection of fields only partial solutions were known. For fields generated by single polynomials in one variable a construction was given by Binder [B96]. Another approach was a spin-off of an algorithm capable of deciding if two finitely generated fields are linear disjoint [MR98]. For two fields being linear disjoint an algorithm for the computation of the intersection is given there.

In this note we introduce the first algorithm for computing the intersection $k(\mathbf{f}) \cap k(\mathbf{g})$ in the general case of two subfields $k(\mathbf{f}) = k(f_1, \ldots, f_r)$ and $k(\mathbf{g}) = k(g_1, \ldots, g_s)$ of a function field $k(X) = \mathrm{Quot}(k[X_1, \ldots, X_n]/\mathrm{I}(X))$ which is finitely generated over a field k of constants.

Intersection of fields: mistake found

3. A (counter-)example: Intersecting fields

As described in Müller-Quade and Beth (1998a), an ideal restriction can be used to compute generators of the intersection $k(\vec{g}) \cap k(\vec{h})$ of two subfields $k(\vec{g}), k(\vec{h}) \subseteq k(\vec{x})$: it is sufficient to find a basis of the ideal

$$\underbrace{\mathfrak{P}_{(\vec{x})/k(\vec{g})}}_{\subseteq k(\vec{g})[\vec{X}]} \cap k(\vec{h})[\vec{X}] \subseteq (k(\vec{g}) \cap k(\vec{h}))[\vec{X}]. \tag{3}$$

Unfortunately, the method discussed in the previous section does not allow the computation of the intersection (3), as in general $k(\vec{h})$ is not a subfield of $k(\vec{g})$. In Müller-Quade and Beth (1998a) an algorithm for accomplishing this task was proposed, but a more detailed analysis shows that it actually computes the ideal $\mathfrak{P}_{(\vec{x})/k(\vec{g})} \cdot k(\vec{x})[\vec{X}] \cap k(\vec{h})[X]$ which in general does not coincide with the ideal (3).

Example. Consider the two subfields $k(\vec{g}) := \mathbb{Q}(x^3 + x^2)$ and $k(\vec{h}) := \mathbb{Q}(x^2)$ of $k(\vec{x}) := \mathbb{Q}(x)$. Then we know from the first example in the previous section that

$$\mathfrak{P}_{(\vec{x})/k(\vec{g})} \cdot k(\vec{x})[\vec{X}] \cap k(\vec{h})[X] = \langle X^6 + 2 \cdot X^5 + X^4 - 2x^2 \cdot X^3 - 2x^2 \cdot X^2 - x^6 + x^4 \rangle.$$

T. Beth et al. / Journal of Symbolic Computation 41 (2006) 372–380 379

As adjoining the coefficients of a reduced Gröbner basis of this ideal to $\mathbb{Q}$ yields the field $\mathbb{Q}(x^2)$, the algorithm from Müller-Quade and Beth (1998a) yields $\mathbb{Q}(x^3 + x^2) \cap \mathbb{Q}(x^2) = \mathbb{Q}(x^2)$, which is clearly wrong.

So it remains an interesting open question whether the techniques described here can be extended in such a way that they allow the computation of a system of generators of the intersection of arbitrary finitely generated extension fields.

Intersection of fields: towards solution

A solution was given in 2009 with a restriction: the fields that are being intersected are algebraically closed in the ambient field.

TECHNISCHE UNIVERSITÄT MÜNCHEN
Zentrum Mathematik

Algorithms for Fields and an Application to a Problem in Computer Vision

Anna Katharina Binder

Intersection of fields: towards solution

A solution was given in 2009 with a restriction: the fields that are being intersected are algebraically closed in the ambient field.

TECHNISCHE UNIVERSITÄT MÜNCHEN
Zentrum Mathematik

Algorithms for Fields and an Application to a Problem in Computer Vision

Anna Katharina Binder

This result is good but is not good enough for our purpose.

Intersection of fields: algorithm

Algorithm Intersection of fields

Input Tuples $\bar{f} := (f_1, \ldots, f_s)$ and $\bar{g} := (g_1, \ldots, g_\ell)$ such that $f_1, \ldots, f_s, g_1, \ldots, g_\ell \in K(\bar{x})$, where $\bar{x} := (x_1, \ldots, x_n)$;

Output If terminates, returns generators of $K(\bar{f}) \cap K(\bar{g})$.

Intersection of fields: algorithm

Algorithm Intersection of fields

Input Tuples $\bar{f} := (f_1, \ldots, f_s)$ and $\bar{g} := (g_1, \ldots, g_\ell)$ such that $f_1, \ldots, f_s, g_1, \ldots, g_\ell \in K(\bar{x})$, where $\bar{x} := (x_1, \ldots, x_n)$;

Output If terminates, returns generators of $K(\bar{f}) \cap K(\bar{g})$.

Notation: Introduce new variables $\overline{Z} := (Z_1, \ldots, Z_n)$. In the algorithm, for $S \subset K(\bar{x})[\overline{Z}]$, $\langle S \rangle$ is the ideal generated by S in $K(\bar{x})[\overline{Z}]$.

Intersection of fields: algorithm

Algorithm Intersection of fields

Input Tuples $\bar{f} := (f_1, \ldots, f_s)$ and $\bar{g} := (g_1, \ldots, g_\ell)$ such that $f_1, \ldots, f_s, g_1, \ldots, g_\ell \in K(\bar{x})$, where $\bar{x} := (x_1, \ldots, x_n)$;

Output If terminates, returns generators of $K(\bar{f}) \cap K(\bar{g})$.

Notation: Introduce new variables $\overline{Z} := (Z_1, \ldots, Z_n)$. In the algorithm, for $S \subset K(\bar{x})[\overline{Z}]$, $\langle S \rangle$ is the ideal generated by S in $K(\bar{x})[\overline{Z}]$.

1. For every $1 \leqslant i \leqslant s$, write $f_i(\bar{x}) = \frac{n_i(\bar{x})}{d_i(\bar{x})}$ so that $n_i, d_i \in K[\bar{x}]$, and set $D(\bar{x}) := d_1 \cdot \ldots \cdot d_s$;

Intersection of fields: algorithm

Algorithm Intersection of fields

Input Tuples $\bar{f} := (f_1, \ldots, f_s)$ and $\bar{g} := (g_1, \ldots, g_\ell)$ such that $f_1, \ldots, f_s, g_1, \ldots, g_\ell \in K(\bar{x})$, where $\bar{x} := (x_1, \ldots, x_n)$;

Output If terminates, returns generators of $K(\bar{f}) \cap K(\bar{g})$.

1. For every $1 \leqslant i \leqslant s$, write $f_i(\bar{x}) = \frac{n_i(\bar{x})}{d_i(\bar{x})}$ so that $n_i, d_i \in K[\bar{x}]$, and set $D(\bar{x}) := d_1 \cdot \ldots \cdot d_s$;
2. Set $i := 1$, $I_1 := \langle 1 \rangle$ and

$$J_1 := \left\langle n_1(\overline{Z}) - f_1(\bar{x})d_1(\overline{Z}), \ldots, n_s(\overline{Z}) - f_s(\bar{x})d_s(\overline{X})\right\rangle \colon D(\overline{Z})^{\infty};$$

Intersection of fields: algorithm

Algorithm Intersection of fields

Input Tuples $\bar{f} := (f_1, \dots, f_s)$ and $\bar{g} := (g_1, \dots, g_\ell)$ such that $f_1, \dots, f_s, g_1, \dots, g_\ell \in K(\bar{x})$, where $\bar{x} := (x_1, \dots, x_n)$;

Output If terminates, returns generators of $K(\bar{f}) \cap K(\bar{g})$.

1. For every $1 \leqslant i \leqslant s$, write $f_i(\bar{x}) = \frac{n_i(\bar{x})}{d_i(\bar{x})}$ so that $n_i, d_i \in K[\bar{x}]$, and set $D(\bar{x}) := d_1 \cdot \ldots \cdot d_s$;
2. Set $i := 1$, $I_1 := \langle 1 \rangle$ and
$$J_1 := \left\langle n_1(\overline{Z}) - f_1(\bar{x})d_1(\overline{Z}), \dots, n_s(\overline{Z}) - f_s(\bar{x})d_s(\overline{X}) \right\rangle : D(\overline{Z})^\infty;$$
3. While $I_i \neq J_i$ do
   - 3.1 $I_{i+1} := \langle J_i \cap K(\bar{g})[\overline{Z}] \rangle$;
   - 3.2 $J_{i+1} := \langle I_{i+1} \cap K(\bar{f})[\overline{Z}] \rangle$;
   - 3.3 $i := i + 1$;

Intersection of fields: algorithm

Algorithm Intersection of fields

Input Tuples $\bar{f} := (f_1, \ldots, f_s)$ and $\bar{g} := (g_1, \ldots, g_\ell)$ such that $f_1, \ldots, f_s, g_1, \ldots, g_\ell \in K(\bar{x})$, where $\bar{x} := (x_1, \ldots, x_n)$;

Output If terminates, returns generators of $K(\bar{f}) \cap K(\bar{g})$.

1. For every $1 \leqslant i \leqslant s$, write $f_i(\bar{x}) = \frac{n_i(\bar{x})}{d_i(\bar{x})}$ so that $n_i, d_i \in K[\bar{x}]$, and set $D(\bar{x}) := d_1 \cdot \ldots \cdot d_s$;
2. Set $i := 1$, $I_1 := \langle 1 \rangle$ and
$$J_1 := \left\langle n_1(\overline{Z}) - f_1(\bar{x})d_1(\overline{Z}), \ldots, n_s(\overline{Z}) - f_s(\bar{x})d_s(\overline{X}) \right\rangle \colon D(\overline{Z})^\infty;$$
3. While $I_i \neq J_i$ do
   - 3.1 $I_{i+1} := \langle J_i \cap K(\bar{g})[\overline{Z}] \rangle$;
   - 3.2 $J_{i+1} := \langle I_{i+1} \cap K(\bar{f})[\overline{Z}] \rangle$;
   - 3.3 $i := i + 1$;
4. Compute any reduced Gröbner basis of J_i and return its coefficients.

Intersection of fields: algorithm

Algorithm Intersection of fields

Input Tuples $\bar{f} := (f_1, \ldots, f_s)$ and $\bar{g} := (g_1, \ldots, g_\ell)$ such that $f_1, \ldots, f_s, g_1, \ldots, g_\ell \in K(\bar{x})$, where $\bar{x} := (x_1, \ldots, x_n)$;

Output If terminates, returns generators of $K(\bar{f}) \cap K(\bar{g})$.

Binder (2009): the algorithm terminates if $K(\bar{f})$ and $K(\bar{g})$ are algebraically closed in $K(\bar{x})$.

.

Intersection of fields: algorithm

Algorithm Intersection of fields

Input Tuples $\bar{f} := (f_1, \ldots, f_s)$ and $\bar{g} := (g_1, \ldots, g_\ell)$ such that $f_1, \ldots, f_s, g_1, \ldots, g_\ell \in K(\bar{x})$, where $\bar{x} := (x_1, \ldots, x_n)$;

Output If terminates, returns generators of $K(\bar{f}) \cap K(\bar{g})$.

Binder (2009): the algorithm terminates if $K(\bar{f})$ and $K(\bar{g})$ are algebraically closed in $K(\bar{x})$.

Our contribution: proved that the algorithm terminates if at least one of $K(\bar{f})$ and $K(\bar{g})$ is algebraically closed in $K(\bar{x})$.

Intersection of fields: algorithm

Algorithm Intersection of fields

Input Tuples $\bar{f} := (f_1, \ldots, f_s)$ and $\bar{g} := (g_1, \ldots, g_\ell)$ such that $f_1, \ldots, f_s, g_1, \ldots, g_\ell \in K(\bar{x})$, where $\bar{x} := (x_1, \ldots, x_n)$;

Output If terminates, returns generators of $K(\bar{f}) \cap K(\bar{g})$.

Binder (2009): the algorithm terminates if $K(\bar{f})$ and $K(\bar{g})$ are algebraically closed in $K(\bar{x})$.

Our contribution: proved that the algorithm terminates if at least one of $K(\bar{f})$ and $K(\bar{g})$ is algebraically closed in $K(\bar{x})$.

Implementation is available here:
`https://github.com/pogudingleb/AllIdentifiableFunctions`

Intersection of fields: algorithm

Algorithm Intersection of fields

Input Tuples $\bar{f} := (f_1, \ldots, f_s)$ and $\bar{g} := (g_1, \ldots, g_\ell)$ such that $f_1, \ldots, f_s, g_1, \ldots, g_\ell \in K(\bar{x})$, where $\bar{x} := (x_1, \ldots, x_n)$;

Output If terminates, returns generators of $K(\bar{f}) \cap K(\bar{g})$.

Binder (2009): the algorithm terminates if $K(\bar{f})$ and $K(\bar{g})$ are algebraically closed in $K(\bar{x})$.

Our contribution: proved that the algorithm terminates if at least one of $K(\bar{f})$ and $K(\bar{g})$ is algebraically closed in $K(\bar{x})$.

Implementation is available here:
`https://github.com/pogudingleb/AllIdentifiableFunctions`

More particular case used in our identifiability algorithm:

$$\mathbb{C}(x_1, \ldots, x_s) \cap \mathbb{C}(\bar{g})$$

in $\mathbb{C}(x_1, \ldots, x_n)$.

Intersection of fields: example

$$K = \mathbb{C}, \quad \bar{f} = (a, b), \quad \bar{g} = (ax + b, x), \quad \mathbb{C}(\bar{f}) \cap \mathbb{C}(\bar{g}) =? \text{ in } \mathbb{C}(a, b, x).$$

Intersection of fields: example

$K = \mathbb{C}, \quad \bar{f} = (a, b), \quad \bar{g} = (ax + b, x), \quad \mathbb{C}(\bar{f}) \cap \mathbb{C}(\bar{g}) = ?$ in $\mathbb{C}(a, b, x)$.

We have $I_1 = \langle 1 \rangle$

Intersection of fields: example

$$K = \mathbb{C}, \quad \bar{f} = (a, b), \quad \bar{g} = (ax + b, x), \quad \mathbb{C}(\bar{f}) \cap \mathbb{C}(\bar{g}) = ? \text{ in } \mathbb{C}(a, b, x).$$

We have $I_1 = \langle 1 \rangle$ and

$$J_1 = \langle Z_1 - a, Z_2 - b \rangle \subset \mathbb{C}(a, b, x)[Z_1, Z_2, Z_3].$$

Intersection of fields: example

$$K = \mathbb{C}, \quad \bar{f} = (a, b), \quad \bar{g} = (ax + b, x), \quad \mathbb{C}(\bar{f}) \cap \mathbb{C}(\bar{g}) = ? \text{ in } \mathbb{C}(a, b, x).$$

We have $I_1 = \langle 1 \rangle$ and

$$J_1 = \langle Z_1 - a, Z_2 - b \rangle \subset \mathbb{C}(a, b, x)[Z_1, Z_2, Z_3].$$

Compute $J_1 \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3]$.

Intersection of fields: example

$$K = \mathbb{C}, \quad \bar{f} = (a, b), \quad \bar{g} = (ax + b, x), \quad \mathbb{C}(\bar{f}) \cap \mathbb{C}(\bar{g}) =? \text{ in } \mathbb{C}(a, b, x).$$

We have $I_1 = \langle 1 \rangle$ and

$$J_1 = \langle Z_1 - a, Z_2 - b \rangle \subset \mathbb{C}(a, b, x)[Z_1, Z_2, Z_3].$$

Compute $J_1 \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3]$. For this, first consider the ideal

$$I := \langle Z_1 - A, Z_2 - B, AX + B - ax - b, X - x \rangle$$

in $\mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3, A, B, X]$,

Intersection of fields: example

$$K = \mathbb{C}, \quad \bar{f} = (a, b), \quad \bar{g} = (ax + b, x), \quad \mathbb{C}(\bar{f}) \cap \mathbb{C}(\bar{g}) = ? \text{ in } \mathbb{C}(a, b, x).$$

We have $I_1 = \langle 1 \rangle$ and

$$J_1 = \langle Z_1 - a, Z_2 - b \rangle \subset \mathbb{C}(a, b, x)[Z_1, Z_2, Z_3].$$

Compute $J_1 \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3]$. For this, first consider the ideal

$$I := \langle Z_1 - A, Z_2 - B, AX + B - ax - b, X - x \rangle$$

in $\mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3, A, B, X]$, and now we compute

$$I_2 := I \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3] = \langle Z_1 x + Z_2 - ax - b \rangle.$$

Intersection of fields: example

$$K = \mathbb{C}, \quad \bar{f} = (a, b), \quad \bar{g} = (ax + b, x), \quad \mathbb{C}(\bar{f}) \cap \mathbb{C}(\bar{g}) =? \text{ in } \mathbb{C}(a, b, x).$$

We have $I_1 = \langle 1 \rangle$ and

$$J_1 = \langle Z_1 - a, Z_2 - b \rangle \subset \mathbb{C}(a, b, x)[Z_1, Z_2, Z_3].$$

Compute $J_1 \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3]$. For this, first consider the ideal

$$I := \langle Z_1 - A, Z_2 - B, AX + B - ax - b, X - x \rangle$$

in $\mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3, A, B, X]$, and now we compute

$$I_2 := I \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3] = \langle Z_1 x + Z_2 - ax - b \rangle.$$

We now compute $J_2 := I_2 \cap \mathbb{C}(a, b)[Z_1, Z_2, Z_3]$.

Intersection of fields: example

$$K = \mathbb{C}, \quad \bar{f} = (a, b), \quad \bar{g} = (ax + b, x), \quad \mathbb{C}(\bar{f}) \cap \mathbb{C}(\bar{g}) = ? \text{ in } \mathbb{C}(a, b, x).$$

We have $I_1 = \langle 1 \rangle$ and

$$J_1 = \langle Z_1 - a, Z_2 - b \rangle \subset \mathbb{C}(a, b, x)[Z_1, Z_2, Z_3].$$

Compute $J_1 \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3]$. For this, first consider the ideal

$$I := \langle Z_1 - A, Z_2 - B, AX + B - ax - b, X - x \rangle$$

in $\mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3, A, B, X]$, and now we compute

$$I_2 := I \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3] = \langle Z_1 x + Z_2 - ax - b \rangle.$$

We now compute $J_2 := I_2 \cap \mathbb{C}(a, b)[Z_1, Z_2, Z_3]$. For this, we consider

$$I' := \langle Z_1 X + Z_2 - AX - B, A - a, B - b \rangle$$

in $\mathbb{C}(a, b)[Z_1, Z_2, Z_3, A, B, X]$

Intersection of fields: example

$$K = \mathbb{C}, \quad \bar{f} = (a, b), \quad \bar{g} = (ax + b, x), \quad \mathbb{C}(\bar{f}) \cap \mathbb{C}(\bar{g}) = ? \text{ in } \mathbb{C}(a, b, x).$$

We have $I_1 = \langle 1 \rangle$ and

$$J_1 = \langle Z_1 - a, Z_2 - b \rangle \subset \mathbb{C}(a, b, x)[Z_1, Z_2, Z_3].$$

Compute $J_1 \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3]$. For this, first consider the ideal

$$I := \langle Z_1 - A, Z_2 - B, AX + B - ax - b, X - x \rangle$$

in $\mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3, A, B, X]$, and now we compute

$$I_2 := I \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3] = \langle Z_1 x + Z_2 - ax - b \rangle.$$

We now compute $J_2 := I_2 \cap \mathbb{C}(a, b)[Z_1, Z_2, Z_3]$. For this, we consider

$$I' := \langle Z_1 X + Z_2 - AX - B, A - a, B - b \rangle$$

in $\mathbb{C}(a, b)[Z_1, Z_2, Z_3, A, B, X]$, and we compute

$$J_2 := I' \cap \mathbb{C}(a, b)[Z_1, Z_2, Z_3] = \langle 0 \rangle,$$

Intersection of fields: example

$$K = \mathbb{C}, \quad \bar{f} = (a, b), \quad \bar{g} = (ax + b, x), \quad \mathbb{C}(\bar{f}) \cap \mathbb{C}(\bar{g}) =? \text{ in } \mathbb{C}(a, b, x).$$

We have $I_1 = \langle 1 \rangle$ and

$$J_1 = \langle Z_1 - a, Z_2 - b \rangle \subset \mathbb{C}(a, b, x)[Z_1, Z_2, Z_3].$$

Compute $J_1 \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3]$. For this, first consider the ideal

$$I := \langle Z_1 - A, Z_2 - B, AX + B - ax - b, X - x \rangle$$

in $\mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3, A, B, X]$, and now we compute

$$I_2 := I \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3] = \langle Z_1 x + Z_2 - ax - b \rangle.$$

We now compute $J_2 := I_2 \cap \mathbb{C}(a, b)[Z_1, Z_2, Z_3]$. For this, we consider

$$I' := \langle Z_1 X + Z_2 - AX - B, A - a, B - b \rangle$$

in $\mathbb{C}(a, b)[Z_1, Z_2, Z_3, A, B, X]$, and we compute

$$J_2 := I' \cap \mathbb{C}(a, b)[Z_1, Z_2, Z_3] = \langle 0 \rangle,$$

which implies $I_3 = J_3 = \langle 0 \rangle$,

Intersection of fields: example

$K = \mathbb{C}, \quad \bar{f} = (a, b), \quad \bar{g} = (ax + b, x), \quad \mathbb{C}(\bar{f}) \cap \mathbb{C}(\bar{g}) =?$ in $\mathbb{C}(a, b, x)$.

We have $I_1 = \langle 1 \rangle$ and

$$J_1 = \langle Z_1 - a, Z_2 - b \rangle \subset \mathbb{C}(a, b, x)[Z_1, Z_2, Z_3].$$

Compute $J_1 \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3]$. For this, first consider the ideal

$$I := \langle Z_1 - A, Z_2 - B, AX + B - ax - b, X - x \rangle$$

in $\mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3, A, B, X]$, and now we compute

$$I_2 := I \cap \mathbb{C}(ax + b, x)[Z_1, Z_2, Z_3] = \langle Z_1 x + Z_2 - ax - b \rangle.$$

We now compute $J_2 := I_2 \cap \mathbb{C}(a, b)[Z_1, Z_2, Z_3]$. For this, we consider

$$I' := \langle Z_1 X + Z_2 - AX - B, A - a, B - b \rangle$$

in $\mathbb{C}(a, b)[Z_1, Z_2, Z_3, A, B, X]$, and we compute

$$J_2 := I' \cap \mathbb{C}(a, b)[Z_1, Z_2, Z_3] = \langle 0 \rangle,$$

which implies $I_3 = J_3 = \langle 0 \rangle$, and so we stop and conclude

$$\mathbb{C}(a, b) \cap \mathbb{C}(ax + b, x) = \mathbb{C}.$$

Intersection of fields: example

$$K = \mathbb{Q}, \quad \bar{f} = X^2, \quad \bar{g} = X^2 + X, \quad \bar{X} = X$$

Intersection of fields: example

$$K = \mathbb{Q}, \quad \bar{f} = X^2, \quad \bar{g} = X^2 + X, \quad \bar{X} = X$$

So, we compute

$$\mathbb{Q}(X^2 + X) \cap \mathbb{Q}(X^2).$$

Intersection of fields: example

$$K = \mathbb{Q}, \quad \bar{f} = X^2, \quad \bar{g} = X^2 + X, \quad \bar{X} = X$$

So, we compute

$$\mathbb{Q}(X^2 + X) \cap \mathbb{Q}(X^2).$$

Binder proved:

$$I_{n+1} = \left(\prod_{i=0}^{n-1} \left(Z_1^2 - X_1^2 + 2iX_1 - i^2 \right) \cdot \prod_{i=0}^{n-1} \left(Z_1^2 - X_1^2 - 2(i+1)X_1 - (i+1)^2 \right) \right)$$

and

$$J_{n+1} = \left(\prod_{i=0}^{n} \left(Z_1^2 - X_1^2 + 2iX_1 - i^2 \right) \cdot \prod_{i=0}^{n-1} \left(Z_1^2 - X_1^2 - 2(i+1)X_1 - (i+1)^2 \right) \right)$$

Intersection of fields: example

$$K = \mathbb{Q}, \quad \bar{f} = X^2, \quad \bar{g} = X^2 + X, \quad \bar{X} = X$$

So, we compute

$$\mathbb{Q}(X^2 + X) \cap \mathbb{Q}(X^2).$$

Binder proved:

$$I_{n+1} = \left(\prod_{i=0}^{n-1} \left(Z_1^2 - X_1^2 + 2iX_1 - i^2 \right) \cdot \prod_{i=0}^{n-1} \left(Z_1^2 - X_1^2 - 2(i+1)X_1 - (i+1)^2 \right) \right)$$

and

$$J_{n+1} = \left(\prod_{i=0}^{n} \left(Z_1^2 - X_1^2 + 2iX_1 - i^2 \right) \cdot \prod_{i=0}^{n-1} \left(Z_1^2 - X_1^2 - 2(i+1)X_1 - (i+1)^2 \right) \right)$$

and so

$$I_1 \supsetneq J_1 \supsetneq I_2 \supsetneq \ldots,$$

and the algorithm never stops.

Open problems

Open problems

Improve the efficiency of our algorithm:

Open problems

Improve the efficiency of our algorithm:

- Improve the efficiency of computing input-output equations by a better choice of ordering variables

Open problems

Improve the efficiency of our algorithm:

- Improve the efficiency of computing input-output equations by a better choice of ordering variables
- Improve the computation of the Wronskian and its reduction modulo the equations (problem: derivatives of high order $\implies$ large expressions)

Open problems

Improve the efficiency of our algorithm:

- Improve the efficiency of computing input-output equations by a better choice of ordering variables
- Improve the computation of the Wronskian and its reduction modulo the equations (problem: derivatives of high order $\implies$ large expressions)
- Improve the efficiency of the intersection of fields algorithm (problem: decomposition into prime components is used; can we do this factorization-free, e.g., using regular chains?)