Symbolic Transformations of Dynamical Models

Gleb Pogudin

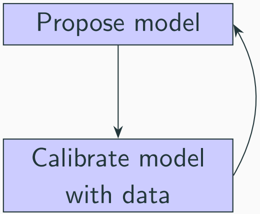
LIX, CNRS, École Polytechnique, IP Paris

HDR defence, December 5th, 2024

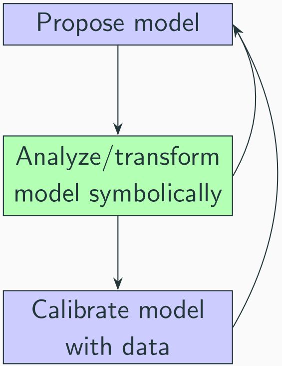
Jury:

François Boulier	Université de Lille
François Fages	INRIA
Joris van der Hoeven	LIX, CNRS
Françoise Point	Université de Mons
Daniel Robertz	RWTH Aachen University
Bernd Sturmfels	Max Planck Institute
Jacques-Arthur Weil	Université de Limoges

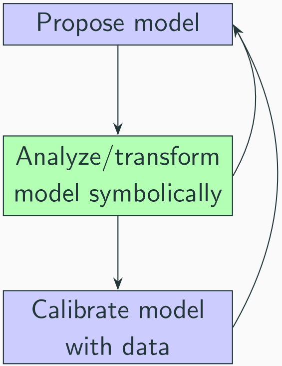
Very simplified workflow



Still simplified workflow



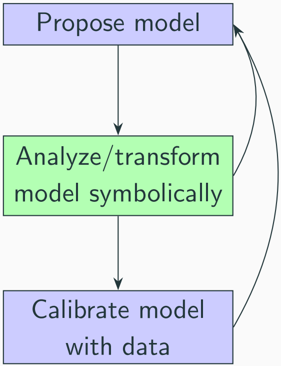
Still simplified workflow



Chapters of the mémoire

1. Introduction
2. Differential and Difference Algebra
3. Differential/Difference Elimination
4. Structural Parameter Identifiability
5. Exact Model Reduction
6. Quadratization
7. Research Project

Still simplified workflow



Chapters of the mémoire

1. Introduction
2. Differential and Difference Algebra
3. Differential/Difference **Elimination**
4. Structural Parameter **Identifiability**
5. Exact Model **Reduction**
6. **Quadratization**
7. Research Project

Brief scientific bio

2012-2016 PhD in (abstract) differential algebra

Brief scientific bio

2012-2016 PhD in (abstract) differential algebra

2016-2017 Postdoc at JKU, Linz (with M. Kauers)

Brief scientific bio

2012-2016 PhD in (abstract) differential algebra

2016-2017 Postdoc at JKU, Linz (with M. Kauers)

- No need to choose between algebra and computation!
- Started working on **Elimination**

Brief scientific bio

2012-2016 PhD in (abstract) differential algebra

2016-2017 Postdoc at JKU, Linz (with M. Kauers)

- No need to choose between algebra and computation!
- Started working on **Elimination**

2017-2019 Postdoc at NYU, New York (with H. Hong, A. Ochinnikov, C. Yap)

Brief scientific bio

2012-2016 PhD in (abstract) differential algebra

2016-2017 Postdoc at JKU, Linz (with M. Kauers)

- No need to choose between algebra and computation!
- Started working on **Elimination**

2017-2019 Postdoc at NYU, New York (with H. Hong, A. Ochinnikov, C. Yap)

- First encounter with modeling and control
- Started working on **Identifiability**

Brief scientific bio

2012-2016 PhD in (abstract) differential algebra

2016-2017 Postdoc at JKU, Linz (with M. Kauers)

- No need to choose between algebra and computation!
- Started working on **Elimination**

2017-2019 Postdoc at NYU, New York (with H. Hong, A. Ochinnikov, C. Yap)

- First encounter with modeling and control
- Started working on **Identifiability**

2019-2019 Assistant professor in HSE (Moscow)

2020-now Assistant professor at École Polytechnique

Brief scientific bio

2012-2016 PhD in (abstract) differential algebra

2016-2017 Postdoc at JKU, Linz (with M. Kauers)

- No need to choose between algebra and computation!
- Started working on **Elimination**

2017-2019 Postdoc at NYU, New York (with H. Hong, A. Ochinnikov, C. Yap)

- First encounter with modeling and control
- Started working on **Identifiability**

2019-2019 Assistant professor in HSE (Moscow)

2020-now Assistant professor at École Polytechnique

- Started working on **Reduction** (supported by DIM RFSI) and **Quadratization** (supported by CNRS INS2I)

Brief scientific bio

2012-2016 PhD in (abstract) differential algebra

2016-2017 Postdoc at JKU, Linz (with M. Kauers)

- No need to choose between algebra and computation!
- Started working on **Elimination**

2017-2019 Postdoc at NYU, New York (with H. Hong, A. Ochinnikov, C. Yap)

- First encounter with modeling and control
- Started working on **Identifiability**

2019-2019 Assistant professor in HSE (Moscow)

2020-now Assistant professor at École Polytechnique

- Started working on **Reduction** (supported by DIM RFSI) and **Quadratization** (supported by CNRS INS2I)
- New developments (including software) in **Identifiability** motivate revisiting **Elimination** (supported by ANR JCJC)

Plan of the talk

1. Elimination
   (focus on theory)
2. Parameter identifiability
   (focus on algorithms and software)
3. Future directions
   (with quadratization sneaking in)

Elimination: what and why?

General formulation

Given a system $\mathbf{f}(\mathbf{x}, \mathbf{y}) = 0$ in $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_m)$

Find nontrivial $g(\mathbf{y}) = 0$ which hold on every solution

Elimination: what and why?

General formulation

Given a system $\mathbf{f}(\mathbf{x}, \mathbf{y}) = 0$ in $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_m)$

Find nontrivial $g(\mathbf{y}) = 0$ which hold on every solution

Linear

$$\begin{cases} 2x + y = 1, \\ x + 2y = 5 \end{cases}$$

Elimination: what and why?

General formulation

Given a system $\mathbf{f}(\mathbf{x}, \mathbf{y}) = 0$ in $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_m)$

Find nontrivial $g(\mathbf{y}) = 0$ which hold on every solution

Linear

$$\begin{cases} 2x + y = 1, \\ x + 2y = 5 \end{cases}$$

$$\implies 3y = 9$$

(Gaussian elimination)

Elimination: what and why?

General formulation

Given a system $\mathbf{f}(\mathbf{x}, \mathbf{y}) = 0$ in $\mathbf{x} = (x_1, \dots, x_n)$, $\mathbf{y} = (y_1, \dots, y_m)$

Find nontrivial $g(\mathbf{y}) = 0$ which hold on every solution

Linear

$$\begin{cases} 2x + y = 1, \\ x + 2y = 5 \end{cases}$$

$$\implies 3y = 9$$

(Gaussian elimination)

Polynomial

$$\begin{cases} x^2 + y^2 = 1, \\ x + y = 0 \end{cases}$$

Elimination: what and why?

General formulation

Given a system $\mathbf{f}(\mathbf{x}, \mathbf{y}) = 0$ in $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_m)$

Find nontrivial $g(\mathbf{y}) = 0$ which hold on every solution

Linear

$$\begin{cases} 2x + y = 1, \\ x + 2y = 5 \end{cases}$$

$$\implies 3y = 9$$

(Gaussian elimination)

Polynomial

$$\begin{cases} x^2 + y^2 = 1, \\ x + y = 0 \end{cases}$$

$$\implies 2y^2 = 1$$

(resultants, GB,...)

Elimination: what and why?

General formulation

Given a system $\mathbf{f}(\mathbf{x}, \mathbf{y}) = 0$ in $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_m)$

Find nontrivial $g(\mathbf{y}) = 0$ which hold on every solution

Linear

$$\begin{cases} 2x + y = 1, \\ x + 2y = 5 \end{cases}$$

$$\implies 3y = 9$$

(Gaussian elimination)

Polynomial

$$\begin{cases} x^2 + y^2 = 1, \\ x + y = 0 \end{cases}$$

$$\implies 2y^2 = 1$$

(resultants, GB,...)

Differential

$$\begin{cases} x' = -y, \\ y' = x \end{cases}$$

Elimination: what and why?

General formulation

Given a system $\mathbf{f}(\mathbf{x}, \mathbf{y}) = 0$ in $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_m)$

Find nontrivial $g(\mathbf{y}) = 0$ which hold on every solution

Linear

$$\begin{cases} 2x + y = 1, \\ x + 2y = 5 \end{cases}$$

$$\implies 3y = 9$$

(Gaussian elimination)

Polynomial

$$\begin{cases} x^2 + y^2 = 1, \\ x + y = 0 \end{cases}$$

$$\implies 2y^2 = 1$$

(resultants, GB,...)

Differential

$$\begin{cases} x' = -y, \\ y' = x \end{cases}$$

$$\implies (y' - x)' + (x' + y) = 0$$

Elimination: what and why?

General formulation

Given a system $\mathbf{f}(\mathbf{x}, \mathbf{y}) = 0$ in $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_m)$

Find nontrivial $g(\mathbf{y}) = 0$ which hold on every solution

Linear

$$\begin{cases} 2x + y = 1, \\ x + 2y = 5 \end{cases}$$

$$\implies 3y = 9$$

(Gaussian elimination)

Polynomial

$$\begin{cases} x^2 + y^2 = 1, \\ x + y = 0 \end{cases}$$

$$\implies 2y^2 = 1$$

(resultants, GB,...)

Differential

$$\begin{cases} x' = -y, \\ y' = x \end{cases}$$

$$\implies (y' - x)' + (x' + y) = 0$$
$$\implies y'' + y = 0$$

(characteristic sets)

Elimination: what and why?

General formulation

Given a system $\mathbf{f}(\mathbf{x}, \mathbf{y}) = 0$ in $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_m)$

Find nontrivial $g(\mathbf{y}) = 0$ which hold on every solution

Linear

$$\begin{cases} 2x + y = 1, \\ x + 2y = 5 \end{cases}$$

$$\implies 3y = 9$$

(Gaussian elimination)

Polynomial

$$\begin{cases} x^2 + y^2 = 1, \\ x + y = 0 \end{cases}$$

$$\implies 2y^2 = 1$$

(resultants, GB,...)

Differential

$$\begin{cases} x' = -y, \\ y' = x \end{cases}$$

$$\implies (y' - x)' + (x' + y) = 0$$
$$\implies y'' + y = 0$$

(characteristic sets)

Why?

- Get simpler equations
- Remove *latent* variables

Difference (discrete-time) equations

Setup

Solution space: two-sided sequences (signals) $(\ldots, a_{-1}, a_0, a_1, \ldots) \in \mathbb{C}^{\mathbb{Z}}$.

Difference (discrete-time) equations

Setup

Solution space: two-sided sequences (signals) $(\ldots, a_{-1}, a_0, a_1, \ldots) \in \mathbb{C}^{\mathbb{Z}}$.

Equations considered: involve arithmetic operations and *shift* σ
i.e. $\sigma(\ldots, a_{-1}, a_0, a_1, \ldots) = (\ldots, a_0, a_1, a_2, \ldots)$

Difference (discrete-time) equations

Setup

Solution space: two-sided sequences (signals) $(\ldots, a_{-1}, a_0, a_1, \ldots) \in \mathbb{C}^{\mathbb{Z}}$.

Equations considered: involve arithmetic operations and *shift* σ
i.e. $\sigma(\ldots, a_{-1}, a_0, a_1, \ldots) = (\ldots, a_0, a_1, a_2, \ldots)$

Example

$$f_{n+2} = f_{n+1} + f_n \rightsquigarrow \sigma^2(F) = \sigma(F) + F$$

Difference (discrete-time) equations

Setup

Solution space: two-sided sequences (signals) $(\ldots, a_{-1}, a_0, a_1, \ldots) \in \mathbb{C}^{\mathbb{Z}}$.

Equations considered: involve arithmetic operations and *shift* σ
i.e. $\sigma(\ldots, a_{-1}, a_0, a_1, \ldots) = (\ldots, a_0, a_1, a_2, \ldots)$

Example

$$f_{n+2} = f_{n+1} + f_n \rightsquigarrow \sigma^2(F) = \sigma(F) + F$$

Not an easy setup! (Hrushovski, Point, 2007)

Undecidable: check if a first order statement in this language is true.

Difference (discrete-time) equations

Setup

Solution space: two-sided sequences (signals) $(\ldots, a_{-1}, a_0, a_1, \ldots) \in \mathbb{C}^{\mathbb{Z}}$.

Equations considered: involve arithmetic operations and *shift* σ
i.e. $\sigma(\ldots, a_{-1}, a_0, a_1, \ldots) = (\ldots, a_0, a_1, a_2, \ldots)$

Example

$$f_{n+2} = f_{n+1} + f_n \rightsquigarrow \sigma^2(F) = \sigma(F) + F$$

Not an easy setup! (Hrushovski, Point, 2007)

Undecidable: check if a first order statement in this language is true.
(sharp contrast with the differential case!)

Difference elimination: example

Illustrating example

$$\begin{cases} x_{n+1} = x_n + y_n \\ x_{n+2} = x_n + z_n \end{cases}$$

Difference elimination: example

Illustrating example

$$\begin{cases} x_{n+1} = x_n + y_n \\ x_{n+2} = x_n + z_n \end{cases} \iff \begin{cases} \sigma(X) = X + Y \\ \sigma^2(X) = X + Z \end{cases}$$

Difference elimination: example

Illustrating example

$$\begin{cases} x_{n+1} = x_n + y_n \\ x_{n+2} = x_n + z_n \end{cases} \iff \begin{cases} \sigma(X) = X + Y \\ \sigma^2(X) = X + Z \end{cases} \overset{?}{\Rightarrow} g(Y, Z, \sigma(Y), \sigma(Z), \ldots) = 0$$

Difference elimination: example

Illustrating example

$$\begin{cases} x_{n+1} = x_n + y_n \\ x_{n+2} = x_n + z_n \end{cases} \iff \begin{cases} \sigma(X) = X + Y \\ \sigma^2(X) = X + Z \end{cases} \overset{?}{\Rightarrow} g(Y, Z, \sigma(Y), \sigma(Z), \ldots) = 0$$

Take the equations **and** a shift:

$$0 = (\underline{\sigma^2(X)} - \underset{\sim}{X} - Z) - \sigma(\underline{\sigma(X)} - \underline{\underline{X}} - Y) - (\underline{\underline{\sigma(X)}} - \underset{\sim}{X} - Y)$$

Difference elimination: example

Illustrating example

$$
\begin{cases} x_{n+1} = x_n + y_n \\ x_{n+2} = x_n + z_n \end{cases} \iff \begin{cases} \sigma(X) = X + Y \\ \sigma^2(X) = X + Z \end{cases} \overset{?}{\Rightarrow} g(Y, Z, \sigma(Y), \sigma(Z), \ldots) = 0
$$

Take the equations **and** a shift:

$$
0 = (\underline{\sigma^2(X)} - \underset{\sim}{X} - Z) - \sigma(\underline{\sigma(X)} - \underline{\underline{X}} - Y) - (\underline{\underline{\underline{\sigma(X)}}} - \underset{\sim}{X} - Y)
$$

$$
= \sigma(Y) + Y - Z = 0
$$

Difference elimination: example

Illustrating example

$$\begin{cases} x_{n+1} = x_n + y_n \\ x_{n+2} = x_n + z_n \end{cases} \iff \begin{cases} \sigma(X) = X + Y \\ \sigma^2(X) = X + Z \end{cases} \overset{?}{\Rightarrow} g(Y, Z, \sigma(Y), \sigma(Z), \ldots) = 0$$

Take the equations **and** a shift:

$$\begin{aligned} 0 &= (\underline{\sigma^2(X)} - \underset{\sim}{X} - Z) - \sigma(\underline{\sigma(X)} - \underline{\underline{X}} - Y) - (\underline{\underline{\underline{\sigma(X)}}} - \underset{\sim}{X} - Y) \\ &= \sigma(Y) + Y - Z = 0 \end{aligned}$$

Idea

1. Take several shifts of the original equations
2. Perform polynomial elimination

Difference elimination: example

Illustrating example

$$\begin{cases} x_{n+1} = x_n + y_n \\ x_{n+2} = x_n + z_n \end{cases} \iff \begin{cases} \sigma(X) = X + Y \\ \sigma^2(X) = X + Z \end{cases} \overset{?}{\Rightarrow} g(Y, Z, \sigma(Y), \sigma(Z), \ldots) = 0$$

Take the equations **and** a shift:

$$0 = (\underline{\sigma^2(X)} - \underset{\sim}{X} - Z) - \sigma(\underline{\sigma(X)} - \underline{\underline{X}} - Y) - (\underline{\underline{\sigma(X)}} - \underset{\sim}{X} - Y)$$

$$= \sigma(Y) + Y - Z = 0$$

Idea

1. Take several shifts of the original equations
2. Perform polynomial elimination

For differential case, done in (Ovchinnikov, P., Vo, 2019).

Difference elimination: Results

Theorem (Ovchinnikov, P., Scanlon, 2020)

There is an (explicit) function $B(n, d)$ such that,
for a system of dimension n and degree d:

Difference elimination possible $\iff$ *Polynomial* elimination possible after $B(n, d)$ shifts

Difference elimination: Results

Theorem (Ovchinnikov, P., Scanlon, 2020)

There is an (explicit) function $B(n, d)$ such that,
for a system of dimension n and degree d:

Difference elimination possible $\iff$ *Polynomial* elimination possible after $B(n, d)$ shifts

$\implies$ first algorithm for discrete-time elimination.

Difference elimination: Results

Theorem (Ovchinnikov, P., Scanlon, 2020)

There is an (explicit) function $B(n, d)$ such that,
for a system of dimension n and degree d:

Difference elimination possible $\iff$ *Polynomial* elimination possible after $B(n, d)$ shifts

$\implies$ first algorithm for discrete-time elimination.

Theorem (OPS, 2020)

For a system of dimension n and degree d:

has a solution in $\mathbb{C}^{\mathbb{Z}}$ $\iff$ has a solution in $\mathbb{C}^{B(n,d)}$

Difference elimination: Results

Theorem (Ovchinnikov, P., Scanlon, 2020)

There is an (explicit) function $B(n, d)$ such that,
for a system of dimension n and degree d:

Difference elimination possible $\iff$ *Polynomial* elimination possible after $B(n, d)$ shifts

$\implies$ first algorithm for discrete-time elimination.

Theorem (OPS, 2020)

For a system of dimension n and degree d:

has a solution in $\mathbb{C}^{\mathbb{Z}}$ $\iff$ has a solution in $\mathbb{C}^{B(n,d)}$

$\implies$ decidability of existence of a solution.

Difference elimination: Results

Theorem (Ovchinnikov, P., Scanlon, 2020)

There is an (explicit) function $B(n, d)$ such that,
for a system of dimension n and degree d:

Difference elimination possible $\iff$ *Polynomial* elimination possible after $B(n, d)$ shifts

$\implies$ first algorithm for discrete-time elimination.

Theorem (OPS, 2020)

For a system of dimension n and degree d:

has a solution in $\mathbb{C}^{\mathbb{Z}}$ $\iff$ has a solution in $\mathbb{C}^{B(n,d)}$

$\implies$ decidability of existence of a solution.

Idea: $\exists$ solution $\iff$ $\exists$ "periodic" solution
but need a correct notion of "periodic" !

On the frontier of decidability

Corollary (OPS, 2020)

Decidable: check if a system $f_1 = \ldots = f_\ell = 0$ has a solution $\mathbb{C}^{\mathbb{Z}}$.

On the frontier of decidability

Corollary (OPS, 2020)

Decidable: check if a system $f_1 = \ldots = f_\ell = 0$ has a solution $\mathbb{C}^{\mathbb{Z}}$.

Theorem (P., Scanlon, Wibmer, 2020)

Undecidable:

- check if a system $f_1 = \ldots = f_\ell = 0$ has a solution $\mathbb{R}^{\mathbb{Z}}$

On the frontier of decidability

Corollary (OPS, 2020)

Decidable: check if a system $f_1 = \ldots = f_\ell = 0$ has a solution $\mathbb{C}^{\mathbb{Z}}$.

Theorem (P., Scanlon, Wibmer, 2020)

Undecidable:

- check if a system $f_1 = \ldots = f_\ell = 0$ has a solution $\mathbb{R}^{\mathbb{Z}}$
- check if a system $f_1 = \ldots = f_\ell = 0$, $g \neq 0$ has a solution $\mathbb{C}^{\mathbb{Z}}$
  i.e. radical ideal membership

On the frontier of decidability

Corollary (OPS, 2020)

Decidable: check if a system $f_1 = \ldots = f_\ell = 0$ has a solution $\mathbb{C}^{\mathbb{Z}}$.

Theorem (P., Scanlon, Wibmer, 2020)

Undecidable:

- check if a system $f_1 = \ldots = f_\ell = 0$ has a solution $\mathbb{R}^{\mathbb{Z}}$
- check if a system $f_1 = \ldots = f_\ell = 0$, $g \neq 0$ has a solution $\mathbb{C}^{\mathbb{Z}}$ *i.e. radical ideal membership*
- check if a system $f_1 = \ldots = f_\ell = 0$ has a solution $\mathbb{C}^{\mathbb{Z}^2}$ *i.e. two shifts like in PDEs*

Concluding remarks

Summary

- first decidability results for consistency and elimination in discrete-time

Concluding remarks

Summary

- first decidability results for consistency and elimination in discrete-time
- basis for further results for delay-differential equations and delay-PDEs in (Li, Ovchinnikov, P., Scanlon, 2020, 2021)

Concluding remarks

Summary

- first decidability results for consistency and elimination in discrete-time
- basis for further results for delay-differential equations and delay-PDEs in (Li, Ovchinnikov, P., Scanlon, 2020, 2021)

Not the end of the story

Concluding remarks

Summary

- first decidability results for consistency and elimination in discrete-time
- basis for further results for delay-differential equations and delay-PDEs in (Li, Ovchinnikov, P., Scanlon, 2020, 2021)

Not the end of the story

- Deciding $f_1 = \ldots = f_\ell = 0 \implies g = 0$ is important.
  Add restrictions to make decidable (e.g. existence of solutions)?

Concluding remarks

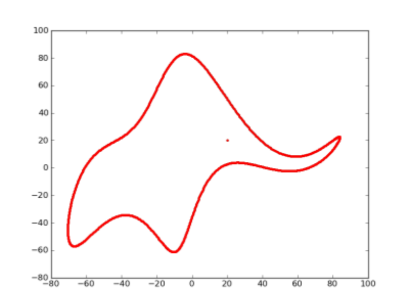
Summary

- first decidability results for consistency and elimination in discrete-time
- basis for further results for delay-differential equations and delay-PDEs in (Li, Ovchinnikov, P., Scanlon, 2020, 2021)

Not the end of the story

- Deciding $f_1 = \ldots = f_\ell = 0 \implies g = 0$ is important.
  Add restrictions to make decidable (e.g. existence of solutions)?
- Known hard cases come from "linear systems with switches".
  Study systematically and obtain lower bounds?

Parameter identifiability



from Mayer et al., 2009

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.

John von Neumann

Toy examples of non-identifiability

Consider a scalar ODE model:

$$x'(t) = (a + b)x(t)$$

with unknown parameters a and b.

Toy examples of non-identifiability

Consider a scalar ODE model:

$$x'(t) = (a + b)x(t)$$

with unknown parameters a and b.

Problem: Given time series for $x(t)$,
find the values of a and b.

Toy examples of non-identifiability

Consider a scalar ODE model:

$$x'(t) = (a + b)x(t)$$

with unknown parameters a and b.

Problem: Given time series for $x(t)$,
find the values of a and b.

Mission impossible: for every c

$$(a_0, b_0) \sim (a_0 - c, b_0 + c)$$

Toy examples of non-identifiability

Consider a scalar ODE model:

$$x'(t) = (a + b)x(t)$$

with unknown parameters a and b.

Problem: Given time series for $x(t)$, find the values of a and b.

Mission impossible: for every c

$$(a_0, b_0) \sim (a_0 - c, b_0 + c)$$

Consider a scalar ODE model:

$$x'(t) = (a + b)x(t) + ab$$

with unknown parameters a and b.

Problem: Same

Toy examples of non-identifiability

Consider a scalar ODE model:

$$x'(t) = (a + b)x(t)$$

with unknown parameters a and b.

Problem: Given time series for $x(t)$, find the values of a and b.

Mission impossible: for every c

$$(a_0, b_0) \sim (a_0 - c, b_0 + c)$$

Consider a scalar ODE model:

$$x'(t) = (a + b)x(t) + ab$$

with unknown parameters a and b.

Problem: Same

Mission still impossible:

$$(a_0, b_0) \sim (b_0, a_0)$$

Toy examples of non-identifiability

Consider a scalar ODE model:

$$x'(t) = (a+b)x(t)$$

with unknown parameters a and b.

Problem: Given time series for $x(t)$, find the values of a and b.

Mission impossible: for every c

$$(a_0, b_0) \sim (a_0 - c, b_0 + c)$$

Consider a scalar ODE model:

$$x'(t) = (a+b)x(t) + ab$$

with unknown parameters a and b.

Problem: Same

Mission still impossible:

$$(a_0, b_0) \sim (b_0, a_0)$$

Can such things naturally occur?

More realistic example

Simple ODE model:

$$\begin{cases} x_1'(t) = -ax_1(t) + bx_2(t), \\ x_2'(t) = -bx_2(t) \end{cases}$$

More realistic example

Simple ODE model:

$$\begin{cases} x_1'(t) = -ax_1(t) + bx_2(t), \\ x_2'(t) = -bx_2(t) \end{cases}$$

We have time series for x_1:

t	0.0	0.2	...	1.0
x_1	1.0	1.1	...	0.3

More realistic example

Simple ODE model:

$$\begin{cases} x_1'(t) = -ax_1(t) + bx_2(t), \\ x_2'(t) = -bx_2(t) \end{cases}$$

We have time series for x_1:

t	0.0	0.2	...	1.0
x_1	1.0	1.1	...	0.3

Want fit a and b to the data

More realistic example

Simple ODE model:

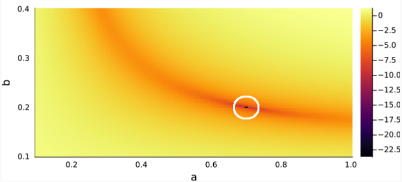
$$\begin{cases} x_1'(t) = -ax_1(t) + bx_2(t), \\ x_2'(t) = -bx_2(t) \end{cases}$$

We have time series for x_1:

t	0.0	0.2	...	1.0
x_1	1.0	1.1	...	0.3

Want fit a and b to the data

Heatmap of log of the minimal L^2 error for $(a, b) \in [0.1, 1.0] \times [0.1, 0.3]$



More realistic example

Simple ODE model:

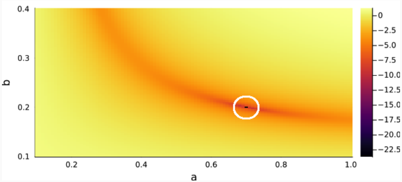
$$\begin{cases} x_1'(t) = -ax_1(t) + bx_2(t), \\ x_2'(t) = -bx_2(t) \end{cases}$$

We have time series for x_1:

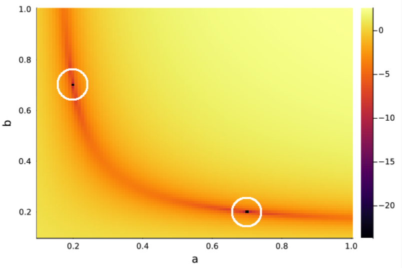
t	0.0	0.2	...	1.0
x_1	1.0	1.1	...	0.3

Want fit a and b to the data

Heatmap of log of the minimal L^2 error for $(a, b) \in [0.1, 1.0] \times [0.1, 0.3]$



Zoom out:



Two perfect fits!

General setup

ODE model with rational rhs

$$\begin{cases} \mathbf{x}'(t) = \mathbf{f}(\mathbf{k}, \mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{y}(t) = \mathbf{g}(\mathbf{k}, \mathbf{x}(t), \mathbf{u}(t)) \end{cases}$$

$$\left.\begin{aligned} \mathbf{x}(t) &= \text{state variables} \\ \mathbf{k} &= \text{parameters} \end{aligned}\right\} \quad - \quad \text{unknown}$$

$$\left.\begin{aligned} \mathbf{y}(t) &= \text{output variables} \\ \mathbf{u}(t) &= \text{input variables} \end{aligned}\right\} \quad - \quad \text{known}$$

General setup

ODE model with rational rhs

$$\begin{cases} \mathbf{x}'(t) = \mathbf{f}(\mathbf{k}, \mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{y}(t) = \mathbf{g}(\mathbf{k}, \mathbf{x}(t), \mathbf{u}(t)) \end{cases}$$

$$\left.\begin{array}{l} \mathbf{x}(t) = \text{state variables} \\ \mathbf{k} = \text{parameters} \end{array}\right\} \quad - \quad \text{unknown}$$

$$\left.\begin{array}{l} \mathbf{y}(t) = \text{output variables} \\ \mathbf{u}(t) = \text{input variables} \end{array}\right\} \quad - \quad \text{known}$$

Structural identifiability problem

A parameter k is identifiable if its value can be determined from $\mathbf{y}(t)$ and $\mathbf{u}(t)$ for generic values of $\mathbf{k}$ and initial conditions

General setup

ODE model with rational rhs

$$\begin{cases} \mathbf{x}'(t) = \mathbf{f}(\mathbf{k}, \mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{y}(t) = \mathbf{g}(\mathbf{k}, \mathbf{x}(t), \mathbf{u}(t)) \end{cases}$$

$$\left.\begin{aligned} \mathbf{x}(t) &= \text{state variables} \\ \mathbf{k} &= \text{parameters} \end{aligned}\right\} \quad - \quad \text{unknown}$$

$$\left.\begin{aligned} \mathbf{y}(t) &= \text{output variables} \\ \mathbf{u}(t) &= \text{input variables} \end{aligned}\right\} \quad - \quad \text{known}$$

Structural identifiability problem

A parameter k is identifiable if its value can be determined from $\mathbf{y}(t)$ and $\mathbf{u}(t)$ for generic values of $\mathbf{k}$ and initial conditions

Lack of identifiability $\implies$ impossibility of reliable estimation

How to check: Taylor series approach

Goes back to Pohjanpalo in the 1970-s.

How to check: Taylor series approach

Goes back to Pohjanpalo in the 1970-s.

Idea

k is identifiable $\iff$ k can be expressed via $\mathbf{y}(0), \mathbf{u}(0), \mathbf{y}'(0), \mathbf{u}'(0), \ldots$

How to check: Taylor series approach

Goes back to Pohjanpalo in the 1970-s.

Idea

k is identifiable $\iff$ k can be expressed via $\mathbf{y}(0), \mathbf{u}(0), \mathbf{y}'(0), \mathbf{u}'(0), \ldots$

Example

$$\begin{cases} x' = x + a, \\ y = x \end{cases} \implies a = y'(0) - y(0) \ \checkmark$$

How to check: Taylor series approach

Goes back to Pohjanpalo in the 1970-s.

Idea

k is identifiable $\iff$ k can be expressed via $\mathbf{y}(0), \mathbf{u}(0), \mathbf{y}'(0), \mathbf{u}'(0), \ldots$

Example

$$\begin{cases} x' = x + a, \\ y = x \end{cases} \implies a = y'(0) - y(0) \ ✓$$

Missing pieces

- Why correct?

How to check: Taylor series approach

Goes back to Pohjanpalo in the 1970-s.

Idea

k is identifiable $\iff$ k can be expressed via $\mathbf{y}(0), \mathbf{u}(0), \mathbf{y}'(0), \mathbf{u}'(0), \ldots$

Example

$$\begin{cases} x' = x + a, \\ y = x \end{cases} \implies a = y'(0) - y(0) \ \checkmark$$

Missing pieces

- Why correct?
- How many Taylor coefficients to take?

How to check: Taylor series approach

Goes back to Pohjanpalo in the 1970-s.

Idea

k is identifiable $\iff$ k can be expressed via $\mathbf{y}(0), \mathbf{u}(0), \mathbf{y}'(0), \mathbf{u}'(0), \ldots$

Example

$$\begin{cases} x' = x + a, \\ y = x \end{cases} \implies a = y'(0) - y(0) \ \checkmark$$

Missing pieces

- Why correct?
- How many Taylor coefficients to take?
- How to check "can be expressed" efficiently?

How to check: Taylor series approach

Goes back to Pohjanpalo in the 1970-s.

Idea

k is identifiable $\iff$ k can be expressed via $\mathbf{y}(0), \mathbf{u}(0), \mathbf{y}'(0), \mathbf{u}'(0), \ldots$

Example

$$\begin{cases} x' = x + a, \\ y = x \end{cases} \implies a = y'(0) - y(0) \ \checkmark$$

Missing pieces

- Why correct?
- How many Taylor coefficients to take?
- How to check "can be expressed" efficiently?

Results (Hong, Ovchinnikov, P., Yap, 2020)

How to check: Taylor series approach

Goes back to Pohjanpalo in the 1970-s.

Idea

k is identifiable $\iff$ k can be expressed via $\mathbf{y}(0), \mathbf{u}(0), \mathbf{y}'(0), \mathbf{u}'(0), \ldots$

Example

$$\begin{cases} x' = x + a, \\ y = x \end{cases} \implies a = y'(0) - y(0) \;\checkmark$$

Missing pieces

- Why correct?
- How many Taylor coefficients to take?
- How to check "can be expressed" efficiently?

Results (Hong, Ovchinnikov, P., Yap, 2020)

- Theorem: Correct!

How to check: Taylor series approach

Goes back to Pohjanpalo in the 1970-s.

Idea

k is identifiable $\iff$ k can be expressed via $\mathbf{y}(0), \mathbf{u}(0), \mathbf{y}'(0), \mathbf{u}'(0), \ldots$

Example

$$\begin{cases} x' = x + a, \\ y = x \end{cases} \implies a = y'(0) - y(0) \ \checkmark$$

Missing pieces

- Why correct?
- How many Taylor coefficients to take?
- How to check "can be expressed" efficiently?

Results (Hong, Ovchinnikov, P., Yap, 2020)

- Theorem: Correct!
- Algorithm to find how many

How to check: Taylor series approach

Goes back to Pohjanpalo in the 1970-s.

Idea

k is identifiable $\iff$ k can be expressed via $\mathbf{y}(0), \mathbf{u}(0), \mathbf{y}'(0), \mathbf{u}'(0), \ldots$

Example

$$\begin{cases} x' = x + a, \\ y = x \end{cases} \implies a = y'(0) - y(0) \ \checkmark$$

Missing pieces

- Why correct?
- How many Taylor coefficients to take?
- How to check "can be expressed" efficiently?

Results (Hong, Ovchinnikov, P., Yap, 2020)

- Theorem: Correct!
- Algorithm to find how many
- Randomized algorithm to check

How to check: Taylor series approach

Goes back to Pohjanpalo in the 1970-s.

Idea

k is identifiable $\iff$ k can be expressed via $\mathbf{y}(0), \mathbf{u}(0), \mathbf{y}'(0), \mathbf{u}'(0), \ldots$

Example

$$\begin{cases} x' = x + a, \\ y = x \end{cases} \implies a = y'(0) - y(0) \ \checkmark$$

Missing pieces

- Why correct?
- How many Taylor coefficients to take?
- How to check "can be expressed" efficiently?

Results (Hong, Ovchinnikov, P., Yap, 2020)

- Theorem: Correct!
- Algorithm to find how many
- Randomized algorithm to check

$\implies$ SIAN software

How to check: Input-output equations approach

Goes back to PhD thesis of F. Ollivier (1990).

How to check: Input-output equations approach

Goes back to PhD thesis of F. Ollivier (1990).

Example

Step 1 Eliminate states:

$$\begin{cases} x_1' = -ax_1 + bx_2, \\ x_2' = -bx_2, \\ y = x_1 \end{cases} \implies \underbrace{y'' + (a+b)y' + aby = 0}_{\text{input-output equation}}.$$

How to check: Input-output equations approach

Goes back to PhD thesis of F. Ollivier (1990).

Example

Step 1 Eliminate states:

$$\begin{cases} x_1' = -ax_1 + bx_2, \\ x_2' = -bx_2, \\ y = x_1 \end{cases} \implies \underbrace{y'' + (a+b)y' + aby = 0}_{\text{input-output equation}}.$$

Step 2 Check if parameter/expression of interest belongs to $\mathbb{C}(a+b, ab)$.

How to check: Input-output equations approach

Goes back to PhD thesis of F. Ollivier (1990).

Example

Step 1 Eliminate states:

$$\begin{cases} x_1' = -ax_1 + bx_2, \\ x_2' = -bx_2, \\ y = x_1 \end{cases} \implies \underbrace{y'' + (a+b)y' + aby = 0}_{\text{input-output equation}}.$$

Step 2 Check if parameter/expression of interest belongs to $\mathbb{C}(a+b, ab)$.

Missing pieces

- Why correct?

How to check: Input-output equations approach

Goes back to PhD thesis of F. Ollivier (1990).

Example

Step 1 Eliminate states:

$$\begin{cases} x_1' = -ax_1 + bx_2, \\ x_2' = -bx_2, \\ y = x_1 \end{cases} \implies \underbrace{y'' + (a+b)y' + aby = 0}_{\text{input-output equation}}.$$

Step 2 Check if parameter/expression of interest belongs to $\mathbb{C}(a+b, ab)$.

Missing pieces

- Why correct?
- How eliminate efficiently?
- How check field membership efficiently?

Input-output equations: assembling the puzzle

Outline of the approach

1. Eliminate states
2. Check if the parameter belongs to the field of the coefficients

Missing piece #1: Why is this approach correct?

Input-output equations: assembling the puzzle

Outline of the approach

1. Eliminate states
2. Check if the parameter belongs to the field of the coefficients

Missing piece #1: Why is this approach correct?

Answers

- (Hong, Ovchinnikov, P., Yap, 2020) Actually, it is not
  (coefficients of the input-output equations may be not identifiable)

Input-output equations: assembling the puzzle

Outline of the approach

1. Eliminate states
2. Check if the parameter belongs to the field of the coefficients

Missing piece #1: Why is this approach correct?

Answers

- (Hong, Ovchinnikov, P., Yap, 2020) Actually, it is not
  (coefficients of the input-output equations may be not identifiable)
- (Ovchinnikov, Pillay, P., Scanlon, 2021)
  Correct but for a different problem *(multi-experiment identifiability)*

Input-output equations: assembling the puzzle

Outline of the approach

1. Eliminate states
2. Check if the parameter belongs to the field of the coefficients

Missing piece #1: Why is this approach correct?

Answers

- (Hong, Ovchinnikov, P., Yap, 2020) Actually, it is not
  (coefficients of the input-output equations may be not identifiable)
- (Ovchinnikov, Pillay, P., Scanlon, 2021)
  Correct but for a different problem *(multi-experiment identifiability)*
- (Dong, Goodbrake, Harrington, P., 2023)
  Efficient correctness check $\Longrightarrow$ in practice, almost always correct

Input-output equations: assembling the puzzle

Outline of the approach

1. Eliminate states
2. Check if the parameter belongs to the field of the coefficients

Missing piece #2: How eliminate efficiently?

Input-output equations: assembling the puzzle

Outline of the approach

1. Eliminate states
2. Check if the parameter belongs to the field of the coefficients

Missing piece #2: How eliminate efficiently?

Answer (Dong, Goodbrake, Harrington, P., 2023)

New differential elimination algorithm tailored to the systems $\mathbf{x}' = f(\mathbf{x}, \mathbf{u}),\ \mathbf{y} = g(\mathbf{x}, \mathbf{u})$:

Input-output equations: assembling the puzzle

Outline of the approach

1. Eliminate states
2. Check if the parameter belongs to the field of the coefficients

Missing piece #2: How eliminate efficiently?

Answer (Dong, Goodbrake, Harrington, P., 2023)

New differential elimination algorithm tailored to the systems $\mathbf{x}' = f(\mathbf{x}, \mathbf{u}),\ \mathbf{y} = g(\mathbf{x}, \mathbf{u})$:

- change-of-ordering point of view
- ideal membership testing with power series solutions

Input-output equations: assembling the puzzle

Outline of the approach

1. Eliminate states
2. Check if the parameter belongs to the field of the coefficients

Missing piece #2: How eliminate efficiently?

Answer (Dong, Goodbrake, Harrington, P., 2023)

New differential elimination algorithm tailored to the systems $\mathbf{x}' = f(\mathbf{x}, \mathbf{u}),\ \mathbf{y} = g(\mathbf{x}, \mathbf{u})$:

- change-of-ordering point of view
- ideal membership testing with power series solutions

$\Longrightarrow$ outperforms general purpose elimination algorithms by orders

Input-output equations: assembling the puzzle

Outline of the approach

1. Eliminate states
2. Check if the parameter belongs to the field of the coefficients

Missing piece #2: How eliminate efficiently?

Answer (Dong, Goodbrake, Harrington, P., 2023)

New differential elimination algorithm tailored to the systems
$\mathbf{x}' = f(\mathbf{x}, \mathbf{u}),\ \mathbf{y} = g(\mathbf{x}, \mathbf{u})$:

- change-of-ordering point of view
- ideal membership testing with power series solutions

$\implies$ outperforms general purpose elimination algorithms by orders

Missing piece #3: How check membership efficiently?

Answer (DGHP, 2023): Randomized algorithm to check

Voilà: StructuralIdentifiability.jl

Resulted in Julia package StructuralIdentifiability.jl.

Model	DAISY	SIAN	SI.jl
SIWR model	OOM	> 5 h.	18 s.
SIWR model - 2	OOM	213 s.	0.7 s.
Pharmacokinetics	> 5 h.	> 5 h.	406 s.
MAPK pathway - 1	OOM	31 s.	39.5 s.
MAPK pathway - 2	> 5 h.	> 5 h.	58 s.
MAPK pathway - 3	> 5 h.	35 s.	1084 s.
SEAIJRC model	OOM	> 5 h.	131.3 s.
Akt pathway	182 s.	28 s.	5 s.
NFκB	> 5 h.	2018 s.	> 5 h.
Mass-action	> 5 h.	3 s.	0.5 s.
SIRS w. forcing	OOM	5 s.	30.3 s.

Ça marche !

PNAS nexus

PNAS Nexus, 2024, 3, 1–15
https://doi.org/10.1093/pnasnexus/pgae031
Advance access publication 20 January 2024
Biological, Health, and Medical Sciences

Signaling-biophysical modeling unravels mechanistic control of red blood cell phagocytosis by macrophages in sickle cell disease

Yu Zhang[a,*,1], Yuhao Qiang[b,1], He Li[c], Guansheng Li[d], Lu Lu[e], Ming Dao[b], George E. Karniadakis[d], Aleksander S. Popel[a,*] and Chen Zhao[f,*]

[a]Department of Biomedical Engineering, School of Medicine, Johns Hopkins University, Baltimore, MD 21205, USA
[b]Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[c]School of Chemical, Materials and Biomedical Engineering, University of Georgia, Athens, GA 30602, USA
[d]School of Engineering, Brown University, Providence, RI 02912, USA
[e]Department of Statistics and Data Science, Yale University, New Haven, CT 06520, USA
[f]School of Pharmacy, Nanjing Medical University, Nanjing, Jiangsu 210029, China
*To whom correspondence should be addressed: Email: zhangyu@jhmi.edu (Y.Z.); Email: apopel@jhu.edu (A.S.P); Email: zcshinchon4677@outlook.com (C.Z.)
[1]Co-first authors.
Edited By: Dennis Discher

Neural Networks 180 (2024) 106732

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

ELSEVIER

Full Length Article

Coagulo-Net: Enhancing the mathematical modeling of blood coagulation using physics-informed neural networks

Ying Qian[a,1], Ge Zhu[b,1], Zhen Zhang[c,1], Susree Modepalli[d], Yihao Zheng[e], Xiaoning Zheng[f], Galit Frydman[g,h], He Li[a,*]

[a] School of Chemical, Materials and Biomedical Engineering, University of Georgia, Athens, USA
[b] Department of Biomedical Engineering, Worcester Polytechnic Institute, Worcester, USA
[c] Division of Applied Mathematics, Brown University, Providence, RI, USA
[d] School of Medicine, Georgetown University, Washington DC, USA
[e] Department of Mechanical and Material Engineering, Worcester Polytechnic Institute, Worcester, USA
[f] Department of Mathematics, College of Information Science & Technology, Jinan University, Guangzhou, Guangdong, 510632, China
[g] Division of Trauma, Emergency Surgery and Surgical Critical Care at the Massachusetts General Hospital, Boston, MA, USA
[h] Division of Comparative Medicine, Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA

Contents lists available at ScienceDirect

Journal of Theoretical Biology

journal homepage: www.elsevier.com/locate/yjtbi

ELSEVIER

Journal of Theoretical Biology

Modeling the CD8+ T cell immune response to influenza infection in adult and aged mice

Benjamin Whipple[a,b], Tanya A. Miura[b,c,d], Esteban A. Hernandez-Vargas[a,b,d,*]

[a] Department of Mathematics and Statistical Science, University of Idaho, Moscow, ID, 83844, United States
[b] Bioinformatics and Computational Biology Program, University of Idaho, Moscow, ID, 83844, United States
[c] Department of Biological Sciences, University of Idaho, Moscow, ID, 83844, United States
[d] Institute for Modeling Collaboration and Innovation, University of Idaho, Moscow, ID, 83844, United States

ARTICLE INFO

Keywords:
Mathematical model
Influenza
Aging
CD8+ T cells
Murine model

ABSTRACT

The CD8+ T cell response is the main determinant of viral clearance during influenza infection. However, influenza viral dynamics and the respective immune responses are affected by the host's age. To investigate age-related differences in the CD8+ T cell immune response dynamics, we propose 16 ordinary differential equation models of existing experimental data. These data consist of viral titer and CD8+ T cell counts collected periodically over a period of 19 days from adult and aged mice infected with influenza A/Puerto Rico/8/34 (H1N1). We use the corrected Akaike Information Criterion to identify the models which best represent the

Chernikova *et al. Applied Network Science* (2023) 8:52
https://doi.org/10.1007/s41109-023-00578-z

Applied Network Science

RESEARCH Open Access

Modeling self-propagating malware with epidemiological models

Alesia Chernikova[1*], Nicolò Gozzi[2], Nicola Perra[3], Simona Boboila[1], Tina Eliassi-Rad[1] and Alina Oprea[1]

*Correspondence: chernikova.a@northeastern.edu
[1] Northeastern University, Boston, MA, USA
[2] ISI Foundation, Turin, Italy
[3] School of Mathematical Sciences, Queen Mary University of London, London, UK

Abstract

Self-propagating malware (SPM) is responsible for large financial losses and major data breaches with devastating social impacts that cannot be understated. Well-known campaigns such as WannaCry and Colonial Pipeline have been able to propagate rapidly on the Internet and cause widespread service disruptions. To date, the propagation behavior of SPM is still not well understood. As result, our ability to defend against these cyber threats is still limited. Here, we address this gap by performing a comprehensive analysis of a newly proposed epidemiological-inspired model for SPM propagation, the *Susceptible-Infected-Infected Dormant-Recovered* (SIIDR) model. We

Concluding remarks

Summary

- Algebra in action $\implies$ two state-of-the-art software tools

Concluding remarks

Summary

- Algebra in action $\Longrightarrow$ two state-of-the-art software tools
- Clarifying foundations gives algorithmic insights

Concluding remarks

Summary

- Algebra in action $\implies$ two state-of-the-art software tools
- Clarifying foundations gives algorithmic insights
- Pushing further the limits of differential elimination
  (and pushing even further $\rightsquigarrow$ *upcoming paper with Y. Mukhina)*

Concluding remarks

Summary

- Algebra in action $\implies$ two state-of-the-art software tools
- Clarifying foundations gives algorithmic insights
- Pushing further the limits of differential elimination
  (and pushing even further $\rightsquigarrow$ *upcoming paper with Y. Mukhina)*

Not the end of the story

- Software is in the process of continuous development
  (including student contributions)

Concluding remarks

Summary

- Algebra in action $\implies$ two state-of-the-art software tools
- Clarifying foundations gives algorithmic insights
- Pushing further the limits of differential elimination
  (and pushing even further $\rightsquigarrow$ *upcoming paper with Y. Mukhina)*

Not the end of the story

- Software is in the process of continuous development
  (including student contributions)
- With nonidentifiability the fun just begins $\rightsquigarrow$ reparametrization/reduction
  (with A. Demin and C. Rackauckas, code is already available)

Concluding remarks

Summary

- Algebra in action $\implies$ two state-of-the-art software tools
- Clarifying foundations gives algorithmic insights
- Pushing further the limits of differential elimination
  (and pushing even further $\rightsquigarrow$ upcoming paper with Y. Mukhina)

Not the end of the story

- Software is in the process of continuous development
  (including student contributions)
- With nonidentifiability the fun just begins $\rightsquigarrow$ reparametrization/reduction
  (with A. Demin and C. Rackauckas, code is already available)
- Efficient elimination via change of representation
  (using Wronskians, integrals, numerical evaluations, etc.)

Research project

Transformation for learning

Scientific Machine Learning

model calibration $\rightsquigarrow$ learning (a part of) model from data

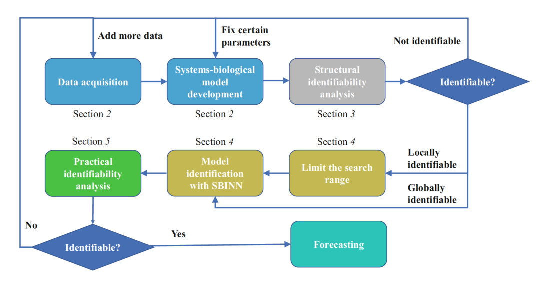
Transformation for learning

Scientific Machine Learning

model calibration $\rightsquigarrow$ learning (a part of) model from data

Use-case #1: structural parameter identifiability

Identifiability assessment as a preprocessing for PINNs.



Systems Biology: Identifiability Analysis and Parameter Identification via Systems-Biology-Informed Neural Networks, Daneker et al., 2023

Transformation for learning

Use-case #2: quadratization for model reduction

Model order reduction: replace an ODE model with a smaller one.

Transformation for learning

Use-case #2: quadratization for model reduction

Model order reduction: replace an ODE model with a smaller one.

- Learning quadratic (degree $\leqslant 2$) reductions is well-understood
- Quadratic reductions are especially natural for quadratic systems

Transformation for learning

Use-case #2: quadratization for model reduction

Model order reduction: replace an ODE model with a smaller one.

- Learning quadratic (degree $\leqslant 2$) reductions is well-understood
- Quadratic reductions are especially natural for quadratic systems

$\Longrightarrow$ Lift & Learn (Qian, Kramer, Peherstorfer, Willcox, 2020)

Transformation for learning

Use-case #2: quadratization for model reduction

Model order reduction: replace an ODE model with a smaller one.

- Learning quadratic (degree $\leqslant 2$) reductions is well-understood
- Quadratic reductions are especially natural for quadratic systems

$\Longrightarrow$ Lift (how?) & Learn (Qian, Kramer, Peherstorfer, Willcox, 2020)

Transformation for learning

Use-case #2: quadratization for model reduction

Model order reduction: replace an ODE model with a smaller one.

- Learning quadratic (degree $\leqslant 2$) reductions is well-understood
- Quadratic reductions are especially natural for quadratic systems

$\implies$ Lift (how?) & Learn (Qian, Kramer, Peherstorfer, Willcox, 2020)

$$x' = x^4 \xRightarrow{y := x^3} \begin{cases} x' = xy, \\ y' = 3x^2 x' = 3x^6 = 3y^2 \end{cases}$$

Transformation for learning

Use-case #2: quadratization for model reduction

Model order reduction: replace an ODE model with a smaller one.

- Learning quadratic (degree $\leqslant 2$) reductions is well-understood
- Quadratic reductions are especially natural for quadratic systems

$\implies$ Lift (how?) & Learn (Qian, Kramer, Peherstorfer, Willcox, 2020)

$$x' = x^4 \xRightarrow{y := x^3} \begin{cases} x' = xy, \\ y' = 3x^2 x' = 3x^6 = 3y^2 \end{cases}$$

- (Bychkov, P., 2021)
  Algorithm for ODEs
- (Bychkov, Issan, P., Kramer, 2024)
  Varying-dimension ODEs

Transformation for learning

Use-case #2: quadratization for model reduction

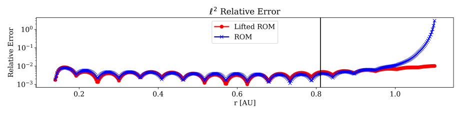
Model order reduction: replace an ODE model with a smaller one.

- Learning quadratic (degree $\leqslant 2$) reductions is well-understood
- Quadratic reductions are especially natural for quadratic systems

$\implies$ Lift (how?) & Learn (Qian, Kramer, Peherstorfer, Willcox, 2020)

$$x' = x^4 \xRightarrow{y := x^3} \begin{cases} x' = xy, \\ y' = 3x^2 x' = 3x^6 = 3y^2 \end{cases}$$

- (Bychkov, P., 2021)
  Algorithm for ODEs
- (Bychkov, Issan, P., Kramer, 2024)
  Varying-dimension ODEs



Transformation for learning

Use-case #2: quadratization for model reduction

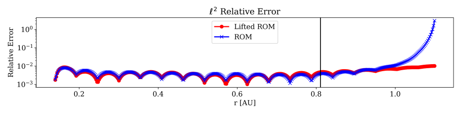
Model order reduction: replace an ODE model with a smaller one.

- Learning quadratic (degree $\leqslant 2$) reductions is well-understood
- Quadratic reductions are especially natural for quadratic systems

$\implies$ Lift (how?) & Learn (Qian, Kramer, Peherstorfer, Willcox, 2020)

$$x' = x^4 \xRightarrow{y := x^3} \begin{cases} x' = xy, \\ y' = 3x^2x' = 3x^6 = 3y^2 \end{cases}$$

- (Bychkov, P., 2021)
  Algorithm for ODEs
- (Bychkov, Issan, P., Kramer, 2024)
  Varying-dimension ODEs
- (Olivieri, P., Kramer, 2024+)
  Algorithm for PDEs



Transformation for learning

Scientific Machine Learning

model calibration $\rightsquigarrow$ learning (a part of) model from data

General research question

How to compute "convenient" coordinates for learning?
(e.g., identifiable reparametrizations, quadratizations for PDEs, etc.)

Calculus of variations

Typical problem: Optimize a functional $\int_a^b F(t, x, x')\,\mathrm{dt}$.

Calculus of variations

Typical problem: Optimize a functional $\int_a^b F(t, x, x')\,\mathrm{dt}$.

Research program

- Perform transformations on the level of functional insead of ODEs.

Calculus of variations

Typical problem: Optimize a functional $\int_a^b F(t, x, x')\,\mathrm{d}t$.

Research program

- Perform transformations on the level of functional insead of ODEs.
- Many problems of strong algebraic flavor: null Lagrangians, homogeneous Lagrangians, inverse problem.

Calculus of variations

Typical problem: Optimize a functional $\int_a^b F(t, x, x')\,\mathrm{dt}$.

Research program

- Perform transformations on the level of functional insead of ODEs.
- Many problems of strong algebraic flavor: null Lagrangians, homogeneous Lagrangians, inverse problem.
- Fundamental object — variation ("shift by a very small function ε").
    - Weak variation: $0 = \varepsilon^2 = (\varepsilon')^2 = (\varepsilon'')^2 = \dots$
    - Strong variation: only $\varepsilon^2 = 0$.

Calculus of variations

Typical problem: Optimize a functional $\int\limits_a^b F(t, x, x')\,\mathrm{dt}$.

Research program

- Perform transformations on the level of functional insead of ODEs.
- Many problems of strong algebraic flavor: null Lagrangians, homogeneous Lagrangians, inverse problem.
- Fundamental object — variation ("shift by a very small function ε").
  - Weak variation: $0 = \varepsilon^2 = (\varepsilon')^2 = (\varepsilon'')^2 = \ldots$
  - Strong variation: only $\varepsilon^2 = 0$.

  Something in the middle?

  Based on algebraic version from (Ait El Manssour, P., 2022, 2024)

All these adventures made possible by

- my teachers
- my postdoc mentors
- my collaborators
- my students
- my colleagues
- and my family

Thank you!