Quadratizations of differential equations

Gleb Pogudin,
MAX team, LIX, CNRS, École Polytechnique, Institut Polytechnique de Paris,
joint work with A. Bychkov, O. Issan, and B. Kramer



Computer Algebra for Functional Equations in Combinatorics and Physics
Institute Henri Poincaré, Paris, December 5

In this talk

Plan

1. Quadratization: what, why, and how?
2. Quadratizing systems of varying (sic!) dimension
3. Open problems

Part I
Quadratization: what, why, and how?

Quadratization: what?

Toy example

Consider one-dimensional ODE system:

$$x' = x^4$$

Quadratization: what?

Toy example

Consider one-dimensional ODE system:

$$x' = x^4 \quad \textit{(degree = 4)}$$

Quadratization: what?

Toy example

Consider one-dimensional ODE system:

$$x' = x^4 \quad \textit{(degree = 4)}$$

Goal: add new variables $\implies$ deg $\leqslant 2$

Quadratization: what?

Toy example

Consider one-dimensional ODE system:

$$x' = x^4 \quad \textit{(degree = 4)}$$

Goal: add new variables $\implies$ deg $\leqslant 2$

Solution: introduce $y := x^3$:

$$\begin{cases} x' = x^4 \\ \end{cases}$$

Quadratization: what?

Toy example

Consider one-dimensional ODE system:

$$x' = x^4 \quad \textit{(degree = 4)}$$

Goal: add new variables $\implies$ deg $\leqslant 2$

Solution: introduce $y := x^3$:

$$\begin{cases} x' = \underline{xy} \\ \end{cases}$$

Quadratization: what?

Toy example

Consider one-dimensional ODE system:

$$x' = x^4 \quad \textit{(degree = 4)}$$

Goal: add new variables $\implies$ deg $\leqslant 2$

Solution: introduce $y := x^3$:

$$\begin{cases} x' = \underline{xy} \\ y' = 3x'x^2 \end{cases}$$

Quadratization: what?

Toy example

Consider one-dimensional ODE system:

$$x' = x^4 \quad \textit{(degree = 4)}$$

Goal: add new variables $\implies$ deg $\leqslant 2$

Solution: introduce $y := x^3$:

$$\begin{cases} x' = \underline{xy} \\ y' = 3x'x^2 = 3x^6 \end{cases}$$

Quadratization: what?

Toy example

Consider one-dimensional ODE system:

$$x' = x^4 \quad \textit{(degree = 4)}$$

Goal: add new variables $\implies$ deg $\leqslant 2$

Solution: introduce $y := x^3$:

$$\begin{cases} x' = \underline{xy} \\ y' = 3x'x^2 = 3x^6 = \underline{3y^2} \end{cases}$$

DONE!

Quadratization: what?

Formal definition. Consider a system in $\bar{x} = (x_1, \ldots, x_n)$:

$$\begin{cases} x_1' = f_1(\bar{x}), \\ \ldots \\ x_n' = f_n(\bar{x}), \end{cases} \quad \text{where } f_1, \ldots, f_n \in \mathbb{C}[\bar{x}]. \tag{1}$$

Quadratization: what?

Formal definition. Consider a system in $\bar{x} = (x_1, \ldots, x_n)$:

$$\begin{cases} x_1' = f_1(\bar{x}), \\ \ldots \\ x_n' = f_n(\bar{x}), \end{cases} \quad \text{where } f_1, \ldots, f_n \in \mathbb{C}[\bar{x}]. \tag{1}$$

New variables $y_1 = g_1(\bar{x}), \ldots, y_m = g_m(\bar{x})$ are called **quadratization** if

Quadratization: what?

Formal definition. Consider a system in $\bar{x} = (x_1, \ldots, x_n)$:

$$\begin{cases} x_1' = f_1(\bar{x}), \\ \ldots \\ x_n' = f_n(\bar{x}), \end{cases} \quad \text{where } f_1, \ldots, f_n \in \mathbb{C}[\bar{x}]. \tag{1}$$

New variables $y_1 = g_1(\bar{x}), \ldots, y_m = g_m(\bar{x})$ are called **quadratization** if there exist $h_1, \ldots, h_{m+n} \in \mathbb{C}[\bar{x}, \bar{y}]$, $\deg h_1, \ldots, \deg h_{m+n} \leqslant 2$ such that

Quadratization: what?

Formal definition. Consider a system in $\bar{x} = (x_1, \ldots, x_n)$:

$$\begin{cases} x_1' = f_1(\bar{x}), \\ \ldots \\ x_n' = f_n(\bar{x}), \end{cases} \quad \text{where } f_1, \ldots, f_n \in \mathbb{C}[\bar{x}]. \tag{1}$$

New variables $y_1 = g_1(\bar{x}), \ldots, y_m = g_m(\bar{x})$ are called **quadratization** if there exist $h_1, \ldots, h_{m+n} \in \mathbb{C}[\bar{x}, \bar{y}]$, $\deg h_1, \ldots, \deg h_{m+n} \leqslant 2$ such that

$$\begin{cases} x_1' = h_1(\bar{x}, \bar{y}), \\ \ldots \\ x_n' = h_n(\bar{x}, \bar{y}) \end{cases} \quad \text{and} \quad \begin{cases} y_1' = h_{n+1}(\bar{x}, \bar{y}), \\ \ldots \\ y_m' = h_{n+m}(\bar{x}, \bar{y}). \end{cases}$$

Quadratization: what?

Formal definition. Consider a system in $\bar{x} = (x_1, \ldots, x_n)$:

$$\begin{cases} x_1' = f_1(\bar{x}), \\ \ldots \\ x_n' = f_n(\bar{x}), \end{cases} \quad \text{where } f_1, \ldots, f_n \in \mathbb{C}[\bar{x}]. \tag{1}$$

New variables $y_1 = g_1(\bar{x}), \ldots, y_m = g_m(\bar{x})$ are called **quadratization** if there exist $h_1, \ldots, h_{m+n} \in \mathbb{C}[\bar{x}, \bar{y}]$, $\deg h_1, \ldots, \deg h_{m+n} \leqslant 2$ such that

$$\begin{cases} x_1' = h_1(\bar{x}, \bar{y}), \\ \ldots \\ x_n' = h_n(\bar{x}, \bar{y}) \end{cases} \quad \text{and} \quad \begin{cases} y_1' = h_{n+1}(\bar{x}, \bar{y}), \\ \ldots \\ y_m' = h_{n+m}(\bar{x}, \bar{y}). \end{cases}$$

In the example for $x' = x^4$ we had

$n = 1$ & $f_1(x) = x^4$

Quadratization: what?

Formal definition. Consider a system in $\bar{x} = (x_1, \ldots, x_n)$:

$$\begin{cases} x_1' = f_1(\bar{x}), \\ \ldots \\ x_n' = f_n(\bar{x}), \end{cases} \quad \text{where } f_1, \ldots, f_n \in \mathbb{C}[\bar{x}]. \tag{1}$$

New variables $y_1 = g_1(\bar{x}), \ldots, y_m = g_m(\bar{x})$ are called **quadratization** if there exist $h_1, \ldots, h_{m+n} \in \mathbb{C}[\bar{x}, \bar{y}]$, $\deg h_1, \ldots, \deg h_{m+n} \leqslant 2$ such that

$$\begin{cases} x_1' = h_1(\bar{x}, \bar{y}), \\ \ldots \\ x_n' = h_n(\bar{x}, \bar{y}) \end{cases} \quad \text{and} \quad \begin{cases} y_1' = h_{n+1}(\bar{x}, \bar{y}), \\ \ldots \\ y_m' = h_{n+m}(\bar{x}, \bar{y}). \end{cases}$$

In the example for $x' = x^4$ we had

$$n = 1 \;\&\; f_1(x) = x^4 \implies m = 1 \;\&\; g_1(x) = x^3$$

Quadratization: what?

Formal definition. Consider a system in $\bar{x} = (x_1, \ldots, x_n)$:

$$\begin{cases} x_1' = f_1(\bar{x}), \\ \ldots \\ x_n' = f_n(\bar{x}), \end{cases} \quad \text{where } f_1, \ldots, f_n \in \mathbb{C}[\bar{x}]. \tag{1}$$

New variables $y_1 = g_1(\bar{x}), \ldots, y_m = g_m(\bar{x})$ are called **quadratization** if there exist $h_1, \ldots, h_{m+n} \in \mathbb{C}[\bar{x}, \bar{y}]$, $\deg h_1, \ldots, \deg h_{m+n} \leqslant 2$ such that

$$\begin{cases} x_1' = h_1(\bar{x}, \bar{y}), \\ \ldots \\ x_n' = h_n(\bar{x}, \bar{y}) \end{cases} \quad \text{and} \quad \begin{cases} y_1' = h_{n+1}(\bar{x}, \bar{y}), \\ \ldots \\ y_m' = h_{n+m}(\bar{x}, \bar{y}). \end{cases}$$

In the example for $x' = x^4$ we had

$$n = 1 \;\&\; f_1(x) = x^4 \implies m = 1 \;\&\; g_1(x) = x^3 \implies \begin{cases} x' = xy = h_1(x, y), \\ y' = 3y^2 = h_2(x, y) \end{cases}$$

Quadratization: why? Part 1

Quadratization: why? Part 1

- Synthesis of chemical reaction networks:

$$\deg \leqslant 2 \iff \text{bimolecular network}$$

(Hemery,Fages,Soliman'2020)

Quadratization: why? Part 1

- Synthesis of chemical reaction networks:

$$\deg \leqslant 2 \iff \text{bimolecular network}$$

  (Hemery,Fages,Soliman'2020)
- Reachability analysis: explicit error bounds for Carleman linearization in the quadratic case
  (Forets, Schilling ' 2021)

Quadratization: why? Part 1

- Synthesis of chemical reaction networks:

$$\deg \leqslant 2 \iff \text{bimolecular network}$$

(Hemery,Fages,Soliman'2020)

- Reachability analysis: explicit error bounds for Carleman linearization in the quadratic case
(Forets, Schilling ' 2021)
- Solving differential equations numerically
(Cochelin& Vergez'2009, Guillot, Cochelin, Vergez'2019)

Quadratization: why? Part 2

Main target application in this talk: Model Order Reduction.

Given:

- Learning quadratic reductions is well-understood
- Quadratic reductions are especially natural for quadratic systems
  (in particular projection of a quadratic model is quadratic)

Quadratization: why? Part 2

Main target application in this talk: Model Order Reduction.

Given:

- Learning quadratic reductions is well-understood
- Quadratic reductions are especially natural for quadratic systems
  (in particular projection of a quadratic model is quadratic)

Ergo: Quadratize and then Reduce

- Projection-based MOR
  (Gu'2011, Brenner & Breiten'2015, Kramer & Willcox' 2019)
- Lift & Learn *(Qian, Kramer, Peherstorfer, Willcox'2020)*

What do we know about quadratizations?

Theorem (e.g., Appelroth'1902, Lagutinskii'1911)

Every ODE system has a quadratization.

What do we know about quadratizations?

Theorem (e.g., Appelroth'1902, Lagutinskii'1911)

Every ODE system has a quadratization.

The proof is constructive and the new variables can be chosen to be monomials.

What do we know about quadratizations?

Theorem (e.g., Appelroth'1902, Lagutinskii'1911)

Every ODE system has a quadratization.

The proof is constructive and the new variables can be chosen to be monomials.

BUT:

Theorem (Hemery, Fages, Soliman' 2020)

Computing optimal quadratization is an NP-hard problem.

Can we find quadratizations?

Existing software (monomial quadratizations)

Can we find quadratizations?

Existing software (monomial quadratizations)

- Biocham *(Hemery, Fages, Soliman, 2020)*
  Via encoding as a MAX-SAT problem. Often optimal but not always.

Can we find quadratizations?

Existing software (monomial quadratizations)

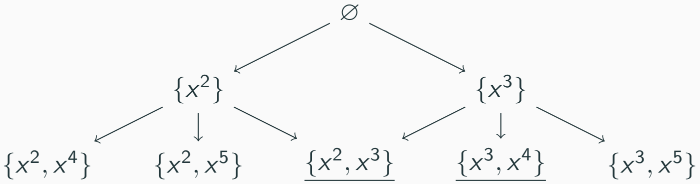
- BIOCHAM *(Hemery, Fages, Soliman, 2020)*
  Via encoding as a MAX-SAT problem. Often optimal but not always.
- QBEE *(Bychkov, P., 2021)*
  Branch & Bound search. Optimality guaranteed.

Can we find quadratizations?

Existing software (monomial quadratizations)

- BIOCHAM *(Hemery, Fages, Soliman, 2020)*
  Via encoding as a MAX-SAT problem. Often optimal but not always.
- QBEE *(Bychkov, P., 2021)*
  Branch & Bound search. Optimality guaranteed.

Example for QBEE: equation $x' = x^4 + x^3$



Can we find quadratizations?

Existing software (monomial quadratizations)

- BIOCHAM *(Hemery, Fages, Soliman, 2020)*
  Via encoding as a MAX-SAT problem. Often optimal but not always.
- QBEE *(Bychkov, P., 2021)*
  Branch & Bound search. Optimality guaranteed.

How far can one go?

Can we find quadratizations?

Existing software (monomial quadratizations)

- BIOCHAM *(Hemery, Fages, Soliman, 2020)*
  Via encoding as a MAX-SAT problem. Often optimal but not always.
- QBEE *(Bychkov, P., 2021)*
  Branch & Bound search. Optimality guaranteed.

How far can one go?

$$\begin{cases} x_1' = x_2^3 + x_7^3, \\ x_2' = x_1^3 + x_3^3, \\ x_3' = x_2^3 + x_4^3, \\ x_4' = x_3^3 + x_5^3, \\ x_5' = x_4^3 + x_6^3, \\ x_6' = x_5^3 + x_7^3, \\ x_7' = x_6^3 + x_1^3. \end{cases}$$

30 sec.

Can we find quadratizations?

Existing software (monomial quadratizations)

- BIOCHAM *(Hemery, Fages, Soliman, 2020)*
  Via encoding as a MAX-SAT problem. Often optimal but not always.
- QBEE *(Bychkov, P., 2021)*
  Branch & Bound search. Optimality guaranteed.

How far can one go?

$$\begin{cases} x_1' = x_2^3 + x_7^3, \\ x_2' = x_1^3 + x_3^3, \\ x_3' = x_2^3 + x_4^3, \\ x_4' = x_3^3 + x_5^3, \\ x_5' = x_4^3 + x_6^3, \\ x_6' = x_5^3 + x_7^3, \\ x_7' = x_6^3 + x_1^3. \end{cases}$$

30 sec.

8 vars $\implies$ 70 sec.

Can we find quadratizations?

Existing software (monomial quadratizations)

- BIOCHAM *(Hemery, Fages, Soliman, 2020)*
  Via encoding as a MAX-SAT problem. Often optimal but not always.
- QBEE *(Bychkov, P., 2021)*
  Branch & Bound search. Optimality guaranteed.

How far can one go?

$$\begin{cases} x_1' = x_2^3 + x_7^3, \\ x_2' = x_1^3 + x_3^3, \\ x_3' = x_2^3 + x_4^3, \\ x_4' = x_3^3 + x_5^3, \\ x_5' = x_4^3 + x_6^3, \\ x_6' = x_5^3 + x_7^3, \\ x_7' = x_6^3 + x_1^3. \end{cases}$$

30 sec.

8 vars $\Longrightarrow$ 70 sec.

$$\begin{cases} x_1' = x_2^2 + (x_1x_2x_3)^2, \\ x_2' = x_3^2 + (x_1x_2x_3)^2, \\ x_3' = x_1^2 + (x_1x_2x_3)^2. \end{cases}$$

100 sec.

Can we find quadratizations?

Existing software (monomial quadratizations)

- BIOCHAM *(Hemery, Fages, Soliman, 2020)*
  Via encoding as a MAX-SAT problem. Often optimal but not always.
- QBEE *(Bychkov, P., 2021)*
  Branch & Bound search. Optimality guaranteed.

How far can one go?

$$\begin{cases} x_1' = x_2^3 + x_7^3, \\ x_2' = x_1^3 + x_3^3, \\ x_3' = x_2^3 + x_4^3, \\ x_4' = x_3^3 + x_5^3, \\ x_5' = x_4^3 + x_6^3, \\ x_6' = x_5^3 + x_7^3, \\ x_7' = x_6^3 + x_1^3. \end{cases}$$

30 sec.

8 vars $\implies$ 70 sec.

$$\begin{cases} x_1' = x_2^2 + (x_1x_2x_3)^2, \\ x_2' = x_3^2 + (x_1x_2x_3)^2, \\ x_3' = x_1^2 + (x_1x_2x_3)^2. \end{cases}$$

100 sec.

4 vars $\implies \infty$

Part II
Quadratizing systems of varying dimension

Where can I get a high-dimensional ODE system?

Where can I get a high-dimensional ODE system?

Recipy

Running example

Take one small PDE system and one large integer N

Where can I get a high-dimensional ODE system?

Recipy

Take one small PDE system and one large integer N

Running example

$$\frac{\partial v}{\partial t} = v + v^2 \frac{\partial v}{\partial \xi}, \quad \text{where } v = v(t, \xi)$$

$$v(t, 0) = \frac{\partial v}{\partial \xi}(t, 1) = 0, \quad N = 100$$

Where can I get a high-dimensional ODE system?

Recipy

Take one small PDE system and one large integer N

Running example

$$\frac{\partial v}{\partial t} = v + v^2 \frac{\partial v}{\partial \xi}, \quad \text{where } v = v(t, \xi)$$

$$v(t, 0) = \frac{\partial v}{\partial \xi}(t, 1) = 0, \quad N = 100$$

Semidiscretize the system

Where can I get a high-dimensional ODE system?

Recipy

Take one small PDE system and one large integer N

Running example

$$\frac{\partial v}{\partial t} = v + v^2 \frac{\partial v}{\partial \xi}, \quad \text{where } v = v(t, \xi)$$

$$v(t, 0) = \frac{\partial v}{\partial \xi}(t, 1) = 0, \quad N = 100$$

Semidiscretize the system

$$x_i(t) = v(t, i/N),$$

$$\frac{\partial v}{\partial \xi}(t, i/N) \approx \frac{x_i(t) - x_{i-1}(t)}{1/N}$$

Where can I get a high-dimensional ODE system?

Recipy

Take one small PDE system and one large integer N

Running example

$$\frac{\partial v}{\partial t} = v + v^2 \frac{\partial v}{\partial \xi}, \quad \text{where } v = v(t, \xi)$$

$$v(t, 0) = \frac{\partial v}{\partial \xi}(t, 1) = 0, \quad N = 100$$

Semidiscretize the system

$$x_i(t) = v(t, i/N),$$

$$\frac{\partial v}{\partial \xi}(t, i/N) \approx \frac{x_i(t) - x_{i-1}(t)}{1/N}$$

$$\Downarrow$$

$$\underbrace{x_i' = x_i + N x_i^2 (x_i - x_{i-1}) \quad i = 1, \ldots, N}_{N\text{-dimensional system}}$$

The Problem

Goal: Quadratize efficiently systems appearing as discretizations

The Problem

Goal: Quadratize efficiently systems appearing as discretizations

Challenges

- The dimension is large (infeasible for QBEE);
- Changing $N \implies$ changing the dimension (*varying dimension*)

The Problem

Goal: Quadratize efficiently systems appearing as discretizations

Challenges

- The dimension is large (infeasible for QBEE);
- Changing $N \implies$ changing the dimension (*varying dimension*)

Idea: equations are abundant but not so different!

Approach: Motivating example

Running example restated

$$\frac{\partial v}{\partial t} = v + v^2 \frac{\partial v}{\partial \xi} \implies x_i' = x_i + N x_i^2 (x_i - x_{i-1}),\; i = 1, \ldots, N$$

Approach: Motivating example

Running example restated

$$\frac{\partial v}{\partial t} = v + v^2 \frac{\partial v}{\partial \xi} \implies \mathbf{x}' = \mathbf{x} + \mathbf{x}^2 \odot (\mathbf{D}\mathbf{x})$$

Approach: Motivating example

Running example restated

$$\frac{\partial v}{\partial t} = v + v^2 \frac{\partial v}{\partial \xi} \implies \mathbf{x}' = \mathbf{x} + \mathbf{x}^2 \odot (\mathbf{D}\mathbf{x})$$

Where:

- $\mathbf{x} = [x_1, \dots, x_N]^T$; $[a_1, a_2, \dots]^T \odot [b_1, b_2, \dots]^T = [a1 b_1, a_2 b_2, \dots]$;
- $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a matrix with $\mathbf{D}_{i,j} = \begin{cases} N, \text{ if } i = j, \\ -N, \text{ if } i = j + 1, \\ 0, \text{ otherwise.} \end{cases}$

Approach: Motivating example

Running example restated

$$\frac{\partial v}{\partial t} = v + v^2 \frac{\partial v}{\partial \xi} \implies \mathbf{x}' = \mathbf{x} + \mathbf{x}^2 \odot (\mathbf{D}\mathbf{x})$$

Where:

- $\mathbf{x} = [x_1, \ldots, x_N]^T$; $[a_1, a_2, \ldots]^T \odot [b_1, b_2, \ldots]^T = [a1 b_1, a_2 b_2, \ldots]$;
- $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a matrix with $\mathbf{D}_{i,j} = \begin{cases} N, & \text{if } i = j, \\ -N, & \text{if } i = j + 1, \\ 0, & \text{otherwise.} \end{cases}$

Quadratization

The following works for every N:

$$\mathbf{w}_1 = \mathbf{x}^2 \quad \text{and} \quad \mathbf{w}_2 = \mathbf{x} \odot \mathbf{S}\mathbf{x}$$

where $\mathbf{S}$ is a shift operator: $\mathbf{S}\mathbf{x} = [0, x_1, \ldots, x_{N-1}]^T$.

Approach: Motivating example

Running example restated

$$\frac{\partial v}{\partial t} = v + v^2 \frac{\partial v}{\partial \xi} \implies \mathbf{x}' = \mathbf{x} + \mathbf{x}^2 \odot (\mathbf{D}\mathbf{x})$$

Where:

- $\mathbf{x} = [x_1, \ldots, x_N]^T$; $[a_1, a_2, \ldots]^T \odot [b_1, b_2, \ldots]^T = [a1b_1, a_2b_2, \ldots]$;
- $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a matrix with $\mathbf{D}_{i,j} = \begin{cases} N, \text{ if } i = j, \\ -N, \text{ if } i = j+1, \\ 0, \text{ otherwise.} \end{cases}$

Quadratization

The following works for every N:

$$\mathbf{w}_1 = \underbrace{\mathbf{x}^2}_{\text{uncoupled}} \quad \text{and} \quad \mathbf{w}_2 = \underbrace{\mathbf{x} \odot \mathbf{S}\mathbf{x}}_{\textit{coupled}}$$

where $\mathbf{S}$ is a shift operator: $\mathbf{S}\mathbf{x} = [0, x_1, \ldots, x_{N-1}]^T$.

Approach: Formal statement

Consider scalar case only to keep notation simple

Input family of ODE systems indexed by number N and matrix $\mathbf{D} \in \mathbb{C}^{N \times N}$:

$$\mathbf{x}' = p_0(\mathbf{x}) + p_1(\mathbf{x}) \odot (\mathbf{D}\mathbf{x})$$

where $\mathbf{x} = [x_1, \ldots, x_N]^T$ and $p_0, p_1 \in \mathbb{C}[x]$ are polynomials.

Approach: Formal statement

Consider scalar case only to keep notation simple

Input family of ODE systems indexed by number N and matrix $\mathbf{D} \in \mathbb{C}^{N \times N}$:

$$\mathbf{x}' = p_0(\mathbf{x}) + p_1(\mathbf{x}) \odot (\mathbf{D}\mathbf{x})$$

where $\mathbf{x} = [x_1, \ldots, x_N]^T$ and $p_0, p_1 \in \mathbb{C}[x]$ are polynomials.

Output family of quadratizations valid of **all** N and $\mathbf{D}$ consisting of

- *uncoupled*: blocks of new variables of the form $\{q(x_1), \ldots, q(x_N)\}$ (like $\mathbf{x}^2$)

Approach: Formal statement

Consider scalar case only to keep notation simple

Input family of ODE systems indexed by number N and matrix $\mathbf{D} \in \mathbb{C}^{N \times N}$:

$$\mathbf{x}' = p_0(\mathbf{x}) + p_1(\mathbf{x}) \odot (\mathbf{D}\mathbf{x})$$

where $\mathbf{x} = [x_1, \ldots, x_N]^T$ and $p_0, p_1 \in \mathbb{C}[x]$ are polynomials.

Output family of quadratizations valid of **all** N and $\mathbf{D}$ consisting of

- *uncoupled*: blocks of new variables of the form $\{q(x_1), \ldots, q(x_N)\}$ (like $\mathbf{x}^2$)
- *coupled*: blocks of new variables of the form $\{q(x_i, x_j) \mid x_j \text{ appears in } x_i'\}$ (like $\mathbf{x} \odot \mathbf{S}\mathbf{x}$).

Approach: Formal statement

Consider scalar case only to keep notation simple

Input family of ODE systems indexed by number N and matrix $\mathbf{D} \in \mathbb{C}^{N \times N}$:

$$\mathbf{x}' = p_0(\mathbf{x}) + p_1(\mathbf{x}) \odot (\mathbf{D}\mathbf{x})$$

where $\mathbf{x} = [x_1, \ldots, x_N]^T$ and $p_0, p_1 \in \mathbb{C}[x]$ are polynomials.

Output family of quadratizations valid of **all** N and $\mathbf{D}$ consisting of

- *uncoupled*: blocks of new variables of the form $\{q(x_1), \ldots, q(x_N)\}$ (like $\mathbf{x}^2$)
- *coupled*: blocks of new variables of the form $\{q(x_i, x_j) \mid x_j \text{ appears in } x_i'\}$ (like $\mathbf{x} \odot \mathbf{S}\mathbf{x}$).

We will call this dimension-agnostic quadratization.

Approach: Underlying theory

One quadratization to rule them all . . . too much to ask?

Approach: Underlying theory

One quadratization to rule them all . . . too much to ask?

Theorem (Bychkov, Issan, P., Kramer, 2023)

No, it's okay

Approach: Underlying theory

One quadratization to rule them all ... too much to ask?

Theorem (Bychkov, Issan, P., Kramer, 2023)

Every family of the form $\mathbf{x}' = p_0(\mathbf{x}) + p_1(\mathbf{x}) \odot (\mathbf{D}\mathbf{x})$
has a (monomial) dimension-agnostic quadratization.

Approach: Underlying theory

One quadratization to rule them all ... too much to ask?

Theorem (Bychkov, Issan, P., Kramer, 2023)

Every family of the form $\mathbf{x}' = p_0(\mathbf{x}) + p_1(\mathbf{x}) \odot (\mathbf{D}\mathbf{x})$
has a (monomial) dimension-agnostic quadratization.

Idea

Fix N and $\mathbf{D}$ and use QBee to search for
a quadratization of such *uncoupled+coupled* form.

Approach: Underlying theory

One quadratization to rule them all ... too much to ask?

Theorem (Bychkov, Issan, P., Kramer, 2023)

Every family of the form $\mathbf{x}' = p_0(\mathbf{x}) + p_1(\mathbf{x}) \odot (\mathbf{D}\mathbf{x})$
has a (monomial) dimension-agnostic quadratization.

Idea

Fix N and $\mathbf{D}$ and use QBEE to search for
a quadratization of such *uncoupled+coupled* form.

Wanted: N_0 and $\mathbf{D}_0$ such that

$$\text{quadratization for } N_0, \mathbf{D}_0 \implies \text{quadratization for } \textbf{any } N, \mathbf{D}$$

Approach: Underlying theory

One quadratization to rule them all ... too much to ask?

Theorem (Bychkov, Issan, P., Kramer, 2023)

Every family of the form $\mathbf{x}' = p_0(\mathbf{x}) + p_1(\mathbf{x}) \odot (\mathbf{D}\mathbf{x})$
has a (monomial) dimension-agnostic quadratization.

Idea

Fix N and $\mathbf{D}$ and use QBEE to search for
a quadratization of such *uncoupled+coupled* form.

Wanted: N_0 and $\mathbf{D}_0$ such that

$$\text{quadratization for } N_0, \mathbf{D}_0 \implies \text{quadratization for } \textbf{any } N, \mathbf{D}$$

Theorem (Bychkov, Issan, P., Kramer, 2023)

$$N_0 = 4, \qquad \mathbf{D}_0 = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Algorithm

Summary

Input Family of the form $\mathbf{x}' = p_0(\mathbf{x}) + p_1(\mathbf{x}) \odot (\mathbf{D}\mathbf{x})$ (not necessarily a single x!)

Output dimension-agnostic quadratization

1. Consider a specific ODE system in the family using $N_0, \mathbf{D}_0$
2. Find a quadratization matching the *uncoupled+coupled* pattern (using QBee)
3. Return the corresponding dimension-agnostic quadratization

Example: Tubular reactor

Model

$$\boldsymbol{\psi}' = \mathbf{b}_\psi - \mathcal{D}\boldsymbol{\psi} \odot (\mathbf{c}_0 + \mathbf{c}_1 \odot \boldsymbol{\theta} + \mathbf{c}_2 \odot \boldsymbol{\theta}^2 + \mathbf{c}_3 \odot \boldsymbol{\theta}^3) + \mathbf{A}_\psi \boldsymbol{\psi},$$
$$\boldsymbol{\theta}' = \mathbf{b}_\theta + \mathbf{b}u + \mathcal{B}\mathcal{D}\boldsymbol{\psi} \odot (\mathbf{c}_0 + \mathbf{c}_1 \odot \boldsymbol{\theta} + \mathbf{c}_2 \odot \boldsymbol{\theta}^2 + \mathbf{c}_3 \odot \boldsymbol{\theta}^3) + \mathbf{A}_\theta \boldsymbol{\theta}$$

where

- $\boldsymbol{\theta}$, $\boldsymbol{\psi}$ — N-dimensional vectors of variables;
- $\mathbf{A}_\psi, \mathbf{A}_\theta$ — $N \times N$ matrices;
- u — external input;
- the rest are constant parameters.

Example: Tubular reactor

Model

$$\psi' = \mathbf{b}_\psi - \mathcal{D}\psi \odot (\mathbf{c}_0 + \mathbf{c}_1 \odot \theta + \mathbf{c}_2 \odot \theta^2 + \mathbf{c}_3 \odot \theta^3) + \mathbf{A}_\psi \psi,$$
$$\theta' = \mathbf{b}_\theta + \mathbf{b}u + \mathcal{B}\mathcal{D}\psi \odot (\mathbf{c}_0 + \mathbf{c}_1 \odot \theta + \mathbf{c}_2 \odot \theta^2 + \mathbf{c}_3 \odot \theta^3) + \mathbf{A}_\theta \theta$$

where

- θ, ψ — N-dimensional vectors of variables;
- $\mathbf{A}_\psi, \mathbf{A}_\theta$ — $N \times N$ matrices;
- u — external input;
- the rest are constant parameters.

Previous work: $7N$-dimensional quadratization found by hand.

Example: Tubular reactor

Model

$$\boldsymbol{\psi}' = \mathbf{b}_{\psi} - \mathcal{D}\boldsymbol{\psi} \odot (\mathbf{c}_0 + \mathbf{c}_1 \odot \boldsymbol{\theta} + \mathbf{c}_2 \odot \boldsymbol{\theta}^2 + \mathbf{c}_3 \odot \boldsymbol{\theta}^3) + \mathbf{A}_{\psi}\boldsymbol{\psi},$$
$$\boldsymbol{\theta}' = \mathbf{b}_{\theta} + \mathbf{b}u + \mathcal{B}\mathcal{D}\boldsymbol{\psi} \odot (\mathbf{c}_0 + \mathbf{c}_1 \odot \boldsymbol{\theta} + \mathbf{c}_2 \odot \boldsymbol{\theta}^2 + \mathbf{c}_3 \odot \boldsymbol{\theta}^3) + \mathbf{A}_{\theta}\boldsymbol{\theta}$$

where

- $\boldsymbol{\theta}$, $\boldsymbol{\psi}$ — N-dimensional vectors of variables;
- $\mathbf{A}_{\psi}, \mathbf{A}_{\theta}$ — $N \times N$ matrices;
- u — external input;
- the rest are constant parameters.

Previous work: $7N$-dimensional quadratization found by hand.

We find $6N$-dimensional automatically.

New variables: $\boldsymbol{\theta}^2$, $\boldsymbol{\theta}^3$, $\boldsymbol{\psi} \odot \boldsymbol{\theta}$, $\boldsymbol{\psi} \odot \boldsymbol{\theta}$.

Case study: Solar wind model (quadratization)

Model

$$\frac{\mathrm{d}\mathbf{v}(r)}{\mathrm{d}r} = \mathbf{D}\ln(\mathbf{v}(r)) - c\mathbf{D}\mathbf{v}(r)$$

Case study: Solar wind model (quadratization)

Model

$$\frac{\mathrm{d}\mathbf{v}(r)}{\mathrm{d}r} = \mathbf{D}\ln(\mathbf{v}(r)) - c\mathbf{D}\mathbf{v}(r) \implies \mathbf{w} := \ln(\mathbf{v}) \implies \begin{cases} \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}r} = \mathbf{D}\mathbf{w} - C_1\mathbf{D}\mathbf{v}, \\ \frac{\mathrm{d}\mathbf{w}}{\mathrm{d}r} = \frac{1}{\mathbf{v}}\mathbf{D}\mathbf{w} - \frac{C_1}{\mathbf{v}}\mathbf{D}\mathbf{v}. \end{cases}$$

Case study: Solar wind model (quadratization)

Model

$$\frac{\mathrm{d}\mathbf{v}(r)}{\mathrm{d}r} = \mathbf{D}\ln(\mathbf{v}(r)) - c\mathbf{D}\mathbf{v}(r) \implies \mathbf{w} := \ln(\mathbf{v}) \implies \begin{cases} \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}r} = \mathbf{D}\mathbf{w} - C_1\mathbf{D}\mathbf{v}, \\ \frac{\mathrm{d}\mathbf{w}}{\mathrm{d}r} = \frac{1}{\mathbf{v}}\mathbf{D}\mathbf{w} - \frac{C_1}{\mathbf{v}}\mathbf{D}\mathbf{v}. \end{cases}$$

Result

- Uncoupled: $\frac{1}{\mathbf{v}}, \ \frac{\mathbf{w}}{\mathbf{v}}$;
- Coupled: $\{\frac{v_j}{v_i}, \frac{w_j}{v_i} \mid (i,j) \in \mathbf{D}\}$

Case study: Solar wind model (quadratization)

Model

$$\frac{\mathrm{d}\mathbf{v}(r)}{\mathrm{d}r} = \mathbf{D}\ln(\mathbf{v}(r)) - c\mathbf{D}\mathbf{v}(r) \implies \mathbf{w} := \ln(\mathbf{v}) \implies \begin{cases} \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}r} = \mathbf{D}\mathbf{w} - C_1\mathbf{D}\mathbf{v}, \\ \frac{\mathrm{d}\mathbf{w}}{\mathrm{d}r} = \frac{1}{\mathbf{v}}\mathbf{D}\mathbf{w} - \frac{C_1}{\mathbf{v}}\mathbf{D}\mathbf{v}. \end{cases}$$

Result

- Uncoupled: $\frac{1}{\mathbf{v}}, \frac{\mathbf{w}}{\mathbf{v}}$;
- Coupled: $\{\frac{v_j}{v_i}, \frac{w_j}{v_i} \mid (i,j) \in \mathbf{D}\}$

Using extra observation + specific form of $\mathbf{D}$

$\implies$ $3N$-dimensional quadratization

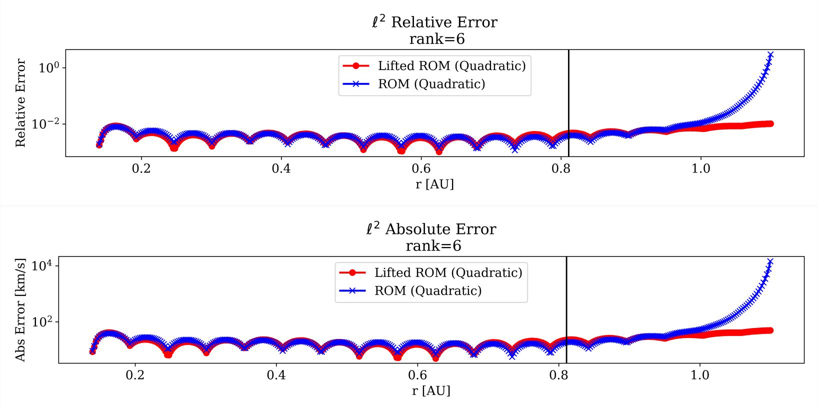
Case study: Solar wind model (order reduction)

Reduction

- $N = 129$;
- reduced to $\ell = 6$.

Learned models

- Blue: learn from original data;
- Red: learn from quadratized data.



Part III
Quadratizing systems of varying dimension

Size of the output

Theorem (Hemery, Fages, Soliman' 2020)

Computing optimal quadratization is an NP-hard problem.

Size of the output

Theorem (Hemery, Fages, Soliman' 2020)

Computing optimal quadratization is an NP-hard problem.

Conjecture (Hemery, Fages, Soliman' 2020)

Minimal number of new variables may be exponential in the size of input.
(particular proposed system $x_i' = x_{i+1}^2 + \prod_{j=1}^{n} x_j^2$*; values:* $3, 10, \leqslant 25$*)*

Size of the output

Theorem (Hemery, Fages, Soliman' 2020)

Computing optimal quadratization is an NP-hard problem.

Conjecture (Hemery, Fages, Soliman' 2020)

Minimal number of new variables may be exponential in the size of input.
(particular proposed system $x_i' = x_{i+1}^2 + \prod_{j=1}^{n} x_j^2$*; values:* $3, 10, \leqslant 25$*)*

Proposition (Bychkov, P.' 2021)

Conjecture is false if new variables can be Laurent monomials.
(the size is in fact linear in this case)

Size of the output

Theorem (Hemery, Fages, Soliman' 2020)

Computing optimal quadratization is an NP-hard problem.

Conjecture (Hemery, Fages, Soliman' 2020)

Minimal number of new variables may be exponential in the size of input.
(particular proposed system $x_i' = x_{i+1}^2 + \prod_{j=1}^{n} x_j^2$*; values:* $3, 10, \leqslant 25$*)*

Proposition (Bychkov, P.' 2021)

Conjecture is false if new variables can be Laurent monomials.
(the size is in fact linear in this case)

Problem

Find algorithm for finding optimal Laurent monomial quadratizations.

Quadratizing PDEs directly

Work in progress to find quadratizations on the level of PDEs (good news — always exists!).

Quadratizing PDEs directly

Work in progress to find quadratizations on the level of PDEs (good news — always exists!).

Inspired open problem

Input differential polynomials $p(x, x', x'', \ldots), q_1(x, x', \ldots), q_n(x, x', \ldots)$ and integer d;

Ouptut TRUE if p can be written as a differential polynomial in $q_1, \ldots, q_n$ of degree at most d, otherwise FALSE.

Quadratizing PDEs directly

Work in progress to find quadratizations on the level of PDEs (good news — always exists!).

Inspired open problem

Input differential polynomials $p(x, x', x'', \ldots), q_1(x, x', \ldots), q_n(x, x', \ldots)$ and integer d;

Ouptut $\textsc{True}$ if p can be written as a differential polynomial in $q_1, \ldots, q_n$ of degree at most d, otherwise $\textsc{False}$.

Remarks

- I think I can solve $d = 1$;
- For quadratization, $d = 2$ is needed.

Conclusions

- Quadratization is a symbolic transformation appearing in many situations;

Conclusions

- Quadratization is a symbolic transformation appearing in many situations;
- Was relatively understood in the simplest finite-dimensional case.

Conclusions

- Quadratization is a symbolic transformation appearing in many situations;
- Was relatively understood in the simplest finite-dimensional case.
- And now settled for coupled families of systems.

Conclusions

- Quadratization is a symbolic transformation appearing in many situations;
- Was relatively understood in the simplest finite-dimensional case.
- And now settled for coupled families of systems.
- Many things left: PDEs, Laurent monomials, arbitrary polynomials, etc.

Thank you!

Partially supported by the PANTOMIME project (AAP INS2I CNRS) and Paris Ile-de-France region.