Discovering Quadratic Representations of PDEs

B. Krämer[1], **A. Olivieri**[1], **and G. Pogudin**[2]

[1] *Department of Mechanical and Aerospace Engineering, University of California San Diego*
[2] *LIX, CNRS, École Polytechnique, Institute Polytechnique de Paris*

Motivation Nonpolynomial and nonquadratic PDEs describe complex dynamical processes in science and engineering; e.g., the cubic FitzHugh-Nagumo model describes the activation and deactivation dynamics of a spiking neuron; the cubic Brusselator model predicts oscillations in chemical reactions; and the quartic model of the nonadiabatic tubular reactor describes the evolution of the species concentration and temperature. Transforming, or lifting, such systems into quadratic form has been integrated into model reduction in [?, ?] and to obtain better variables for model learning [?]. In all of these, the lifting transformation to quadratic form was done by hand on either the ODE or PDE. This is tedious, error-prone, and often results in suboptimal lifted transformations.

Polynomialization Module Polynomialization of a PDE is the process of finding a transformation that turns a nonpolynomial PDE into a system with possibly high-order polynomial drift. We developed a polynomialization module that takes as input the symbolic form of a non-polynomial PDE and generates a polynomial PDE in a new set of variables. The output is the symbolic polynomial form.

Quadratization Module Quadratization turns a PDE system with higher-degree polynomial drift into systems with quadratic drift. The inputs of our quadratization module are the symbolic form of the PDE, and the output is a quadratic PDE alongside a new set of variables. To obtain a quadratic form, it is often required to add new variables to the system. The set of variables introduced is called a quadratization. For illustration, consider the PDE describing the evolution of the space and time-varying function $u(t, x)$ as

$$u_t = u_x u^2 \tag{1}$$

To quadratize (1) we introduce the variable $y := u^2$ and calculate its first derivative in x: $y_x = 2u_x u$, which allows us to write

$$y_t = 2u_x u^3 = 2u_x uy = y_x y \quad \textit{and} \quad u_t = u_x y. \tag{2}$$

This quadratic equation in $u(t, x)$ and $y(t, x)$ is the output of the quadratization module, so that the set $\{u^2\}$ is a quadratization for (1).

Functionalities of the Integrated Discovery System We developed an algorithm and software that finds quadratizations of PDEs. The presented algorithm searches a combinatorial tree of possible transformations, uses branch-and-bound techniques to curb its computational complexity, and outputs a (sometimes minimal) set of variables that effectively quadratize a PDE system. To the best of our knowledge, these are the first results of automated quadratization for PDEs. The module is further being equipped with a functionality to deal with non-autonomous systems with differentiable forcing functions.

References

[1] C. Gu. QLMOR: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(9):1307–1320, 2011.

[2] B. Kramer and K. Willcox. Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. *AIAA Journal*, 57(6):2297–2307, 2019.

[3] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox. Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica D: Nonl. Phen.*, 406:132401, 2020.