The Journal of Open Source Software

TikhonovFenichelReductions.jl: A systematic approach to geometric singular perturbation theory

Johannes Apelt[1]

1 Institute of Mathematics and Computer Science, University of Greifswald, Germany

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Singular perturbation theory is a mathematical toolbox that allows dimensionality reduction of ODE systems whose components evolve on different time scales emanating from the presence of a small parameter ε. Intuitively, the fast components evolve so quickly that they can be approximated with their steady state on the slow time scale for ε sufficiently small. More precisely, we get the limit

$$\begin{aligned} \dot{u} &= g(u,v) \\ \varepsilon\dot{v} &= h(u,v) \end{aligned} \quad \xrightarrow[\varepsilon\to 0]{} \quad \begin{aligned} \dot{u} &= g(u,v) \\ 0 &= h(u,v) \end{aligned} \tag{1}$$

where the *reduced system* on the right is defined on the so-called *slow manifold* $M_0 = \{(u,v)\,|\,h(u,v) = 0\}$ and its dynamics can be described by the slow component u alone (i.e. by local coordinates on M_0). This is known as Tikhonov's (1952) theorem (see Verhulst, 2007 for its present-day form). Dimensionality reduction is very useful for model derivation and analysis, but requiring the system to be in standard form as in Equation 1 can make it difficult to find meaningful reductions, as it relies on the choice of appropriate coordinates. The importance of a coordinate-free approach was already pointed out by Fenichel (1979), who established a geometric singular perturbation theory (GSPT; see Wechselberger (2020) for an overview). More recently, Goeke and Walcher (together with colleagues) developed an algebraic approach to GSPT that allows to find all critical parameters admitting a reduction in the sense of Tikhonov and Fenichel together with their slow manifolds (implicitly given as affine varieties) systematically for polynomial or rational ODE systems (Goeke, 2013; Goeke et al., 2015; Goeke & Walcher, 2013, 2014). These critical parameters define a slow-fast separation of processes instead of components, i.e. we obtain reductions for systems of the form

$$\dot{x} = f(x,\pi) = f^{(0)}(x) + \varepsilon f^{(1)}(x) + \mathcal{O}(\varepsilon^2), \quad x \in \mathbb{R}^n, \pi \in \mathbb{R}^m, \tag{2}$$

which renders this a coordinate-free approach. The main idea is to evaluate necessary conditions for a slow-fast separation as in Equation 2 for the existence of a reduction.

`TikhonovFenichelReductions.jl` is a `Julia` (Bezanson et al., 2017) package implementing this approach for polynomial ODE systems. Apelt & Liebscher (2025) provide a showcasing example and more detailed explanations.

Statement of need

Ad-hoc approaches to singular perturbation theory require prior knowledge about a suitable time scale separation and substantial mathematical effort to compute the reduction. The algebraic approach due to Goeke and Walcher yields algorithmically accessible conditions for the existence of a reduction, which allows us to find all reductions of a given ODE system (Goeke et al., 2015). This relies on methods from computational algebra, such as the computation of Gröbner bases, minimal primary decompositions, normal forms and symbolic matrix operations, which are implemented in many computer algebra systems. However, their usage for the problem at hand is not trivial. `TikhonovFenichelReductions.jl` makes the required computations easily accessible (for non-expert users) by utilizing the package `Oscar.jl` (Decker et al., 2025; The OSCAR Team, 2025), which provides a unified framework combining different computer algebra tools, and the performance and flexibility of `Julia` (Bezanson et al., 2017).

The author is not aware of any publicly available implementation of this particular theory, but there exist an implementation for a related approach for computing invariant manifolds by Roberts (n.d., 1997).

Features

The main features provided by `TikhonovFenichelReductions.jl` are the search for critical parameters admitting a reduction, so-called *Tikhonov-Fenichel Parameter Values (TFPVs)*, and the computation of the corresponding reduced systems. The general procedure is discussed in the following. More detailed explanations and examples are given by Apelt & Liebscher (2025) and can be found in the documentation of the package.

Finding TFPVs

To find TFPVs, we first construct an instance of type `ReductionProblem`, which creates all symbolic objects needed and parses the system given as a `Julia` function. For this, we specify the desired dimension s of the slow manifold. Then, there are two methods for finding TFPVs.

1. `tfpvs_and_varieties` returns all TFPV candidates for which some parameters are set to zero, which we call *slow-fast separations of rates*, together with the corresponding potential slow manifolds given implicitly as affine varieties (i.e. the common zeros of a set of polynomials) and their dimensions. Note that such a TFPV can admit multiple slow manifolds corresponding to the irreducible components of the variety $\mathcal{V}(f^{(0)})$.
2. `tfpvs_groebner` computes a Gröbner basis G with an elimination ordering for the state variables of an ideal obtained from necessary conditions for the existence of a reduction. Then, every TFPV is contained in $\mathcal{V}(G)$. Note that this can be a computationally intensive task. If it is feasible to compute G for the problem at hand, it may reveal more complex TFPVs determined by expressions in the original parameters that can be considered small. Introducing new parameters from these expressions typically allows us to use the methods for slow-fast separations of rates, in particular the computation of the reduced system as implemented in `TikhonovFenichelReductions.jl`.

Computing reductions

In order to compute a reduction for a slow-fast separation of rates as in Theorem 1 in Goeke & Walcher (2014), we construct an instance of type `Reduction`, which holds all the relevant information. Let Y be the irreducible component of $\mathcal{V}(f^{(0)})$ with dimension s corresponding to the slow manifold M_0 (as returned by `tfpvs_and_varieties`).

First, we have to find an explicit description of M_0. In most cases this can be obtained automatically with the heuristic `get_explicit_manifold`. Alternatively, we have to parameterize Y explicitly w.r.t. r state variables (the local coordinates). Then we call `set_manifold!` to store the results in the `Reduction` instance.

Next, we need to find a product decomposition $P(x)\psi(x) = f^{(0)}(x)$, where $P(x)$ is a $n \times r$ matrix of rational functions and $\psi(x)$ is a vector of polynomials, such that locally $\mathcal{V}(\psi) = \mathcal{V}(f^{(0)})$ and $\operatorname{rank} P(x) = \operatorname{rank} D\psi(x) = r$ hold. ψ can be constructed from r independent entries of $f^{(0)}$ or the generators of Y (if the corresponding ideal has exactly r generators). The package contains a heuristic to compute P automatically from ψ, which may fail if the r entries for ψ cannot be set automatically. In this case, P and ψ need to be computed manually. The function `set_decomposition!` dispatches on the appropriate method and can take P and ψ together, ψ, or an instance of `Variety` as arguments.

If the slow manifold and product decomposition are set correctly, `compute_reduction!` yields the reduced system, which is given as

$$\dot{x} = \left[1_n - P(x)(D\psi(x)P(x))^{-1}D\psi(x)\right] f^{(1)}(x)$$

(Goeke & Walcher, 2014).

For complex systems, there may exist many TFPV candidates, which makes the analysis of interesting cases intractable. To circumvent this problem, we can use the function `unique_varieties` to find all the potential slow manifolds with dimension s that exist for the input system. Typically, there are much fewer unique varieties than TFPV candidates and it suffices to find explicit descriptions of the corresponding manifolds (e.g. with `get_explicit_manifold`) for these cases. Then, we can compute all reductions grouped by slow manifolds with `compute_reductions`. Note that this relies on the heuristic to set the product decomposition for $f^{(0)}$. Overall, this means we can compute all or at least most reductions fully automatically in most cases.

Integration with the `Julia` ecosystem

The initialization of a `ReductionProblem` can be done directly from a reaction network defined with `Catalyst.jl` (Loman et al., 2023). This is particularly useful as singular perturbation is commonly used in modelling of (bio)chemical reactions – often in the form of quasi-steady state reductions (Goeke & Walcher, 2013).

In modelling applications, the search for critical parameters and computation of the reduction is only the first step followed by model analysis. Because the reduced systems obtained with `TikhonovFenichelReductions.jl` are already parsed to the appropriate types from `Oscar.jl`, one can use functions from the latter package that aid the symbolic analysis (e.g. computing a minimal primary decomposition to find fixed points, factor polynomials, etc.).

Because metaprogramming is supported in `Julia`, it is also possible to build `Julia` functions from the symbolically given reduction. This allows to easily get an overview of the behaviour of the reductions by using numerical simulations and avoids having to parse code between languages and frameworks. See e.g. the small package `TFRSimulations.jl`, that yields a responsive GUI showing simulations of a reduced system directly from an instance of `Reduction`.

For convenience, there are several methods for displaying the output, including printing as LaTeX source code via `Latexify.jl`.

Acknowledgements

This work was supported by a scholarship awarded by the University of Greifswald according to the "Landesgraduiertenförderungsgesetz (LGFG) MV". I like to thank Volkmar Liebscher for his supervision and discussing the maths behind this software package, Sebastian Walcher and Alexandra Goeke for the development of the Tikhonov-Fenichel reduction theory and for making me aware of its many advantages, Leonard Schmitz for the helpful exchanges about the computational algebra used in this paper, and the OSCAR team for their quick and helpful replies to my questions and remarks.

References

Apelt, J., & Liebscher, V. (2025). Tikhonov-Fenichel reductions and their application to a novel modelling approach for mutualism. *Theoretical Population Biology*, 16–35. https:

//doi.org/10.1016/j.tpb.2025.08.004

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Decker, W., Eder, C., Fieker, C., Horn, M., & Joswig, M. (Eds.). (2025). *The Computer Algebra System OSCAR: Algorithms and Examples*. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-62127-7

Fenichel, N. (1979). Geometric singular perturbation theory for ordinary differential equations. *Journal of Differential Equations*, *31*(1), 53–98. https://doi.org/10.1016/0022-0396(79)90152-9

Goeke, A. (2013). *Reduktion und asymptotische Reduktion von Reaktionsgleichungen* [RWTH Aachen]. https://publications.rwth-aachen.de/record/229008

Goeke, A., & Walcher, S. (2013). Quasi-Steady State: Searching for and Utilizing Small Parameters. In A. Johann, H.-P. Kruse, F. Rupp, & S. Schmitz (Eds.), *Recent Trends in Dynamical Systems* (Vol. 35, pp. 153–178). Springer Basel. https://doi.org/10.1007/978-3-0348-0451-6_8

Goeke, A., & Walcher, S. (2014). A constructive approach to quasi-steady state reductions. *Journal of Mathematical Chemistry*, *52*(10), 2596–2626. https://doi.org/10.1007/s10910-014-0402-5

Goeke, A., Walcher, S., & Zerz, E. (2015). Determining "small parameters" for quasi-steady state. *Journal of Differential Equations*, *259*(3), 1149–1180. https://doi.org/10.1016/j.jde.2015.02.038

Loman, T. E., Ma, Y., Ilin, V., Gowda, S., Korsbo, N., Yewale, N., Rackauckas, C., & Isaacson, S. A. (2023). Catalyst: Fast and flexible modeling of reaction networks. *PLOS Computational Biology*, *19*(10), e1011530. https://doi.org/10.1371/journal.pcbi.1011530

Roberts, A. J. (n.d.). *Construct invariant manifolds*. https://tuck.adelaide.edu.au/gencm.php

Roberts, A. J. (1997). Low-dimensional modelling of dynamics via computer algebra. *Computer Physics Communications*, *100*(3), 215–230. https://doi.org/10.1016/s0010-4655(96)00162-2

The OSCAR Team. (2025). *OSCAR – Open Source Computer Algebra Research system*. https://www.oscar-system.org

Tikhonov, A. N. (1952). Systems of differential equations containing small parameters in the derivatives. *Matematicheskii sbornik*, *73*(3), 575–586.

Verhulst, F. (2007). Singular perturbation methods for slow–fast dynamics. *Nonlinear Dynamics*, *50*(4), 747–753. https://doi.org/10.1007/s11071-007-9236-z

Wechselberger, M. (2020). *Geometric Singular Perturbation Theory Beyond the Standard Form*. Springer International Publishing. https://doi.org/10.1007/978-3-030-36399-4