Inference of Bifurcations with Differentiable Continuation

Gregory Szep, Attila Csikász-Nagy
King's College London
London, WC2R 2LS
gregory.szep@kcl.ac.uk

Neil Dalchau
Microsoft Research Cambridge
Cambridge, CB1 2FB
ndalchau@microsoft.com

Abstract

In this work we propose a gradient-based semi-supervised approach for matching target bifurcations with parameterised differential equation models. The cost function contains a supervised term that is minimal when predicted bifurcations match the targets and an unsupervised bias that encourages bifurcations by maximising zero crossings in the determinant of the Jacobian. The calculation of gradients with respect to parameters shares the same computational complexity as deflated pseudo-arclength continuation used to calculate the bifurcation diagram. We demonstrate model synthesis with minimal models which explore the space of saddle-node and pitchfork diagrams, a genetic toggle switch from synthetic biology and the FitzHugh-Nagumo model. Furthermore, the cost landscape allows us to organise models in terms of topological and geometric equivalence.

1 Introduction

Backpropagation through differential equation solves has been a breakthrough over the past couple of years [**?** **?**] that enabled scalable parameter inference for differential equations. Determining what model parameters should be for a given set of observations, in the setting of biology and engineering, are known as inverse problems [**?**].

Optimisation targets, however, have mostly been expressed in the spatio-temporal domain. Acquisition of such data can be costly and can often contain over-constraining information generated by processing steps or the measurement device rather than the state variables that underpin the observed mechanism. In microscopy, for example, data is often reported in arbitrary fluoresce units allowing the observer to shift and scale data arbitrarily. Furthermore, such data may also not contain sufficient information about dynamical transients in order to identify kinetic parameters. The emerging picture suggests that identification of the qualitative behaviour – the bifurcation diagram – should precede any attempt at inferring kinetic parameters [**?**]. Techniques for back-propagating through implicit equation solvers have also been developed [**?** **?**] although to the best of the authors' knowledge have not been applied to bifurcation diagrams at the time of writing this paper.

In this work we propose a gradient-based semi-supervised approach that focuses on fitting high-level qualitative constraints, defined by state space structures, rather than kinetics. Drawing inspiration from implicit solvers [**?** **?**] to calculate gradients we find that their computation shares the same complexity as the algorithm used to calculate the bifurcation diagram. We use a predictor-corrector method called deflated pseudo-arclength continuation [**?** **?**], originally developed for partial differential equations. In the case of partial differential equations the computational complexity of calculating a single bifurcation diagram is not bounded since superpositions of localised solutions may give rise to uncountably many branches [**?**]. The complexity of computing a single branch, however, is bounded by the complexities of the chosen eigenvalue solver and corrector, and further decreased by adaptive stepping procedures [**?**].

Preprint. Under review.

We find that the cost function landscape contains basins that not only allow us to synthesise models with a desired bifurcation structure but also allow us to organise models in terms of topological and geometric equivalence. We discuss the relevance of this in model selection. In summary, our paper has the following main contributions:

- Defined a differentiable semi-supervised cost function for encouraging and placing co-dimension one bifurcations to user-specified target locations
- Implementation of method in Julia package `FluxContinuation.jl` leveraging automatic differentiation tooling in `Flux.jl`
- Leveraging the cost landscape for a novel way of organising differential equation models in terms of geometric and topological equivalence

1.1 Related Works | *in progress*

Bifurcation control and *inverse* bifurcation analysis literature applies control theory to enable design and identification of dynamical systems with qualitative constraints. The first works involved the calculation of the minimal distances to bifurcation manifolds [**?** **?** **?**]. Those optimisations did not involve closed form expressions and the challenge of finding the manifolds to being with was left to random sampling.

Matching vector fields From a dynamical systems perspective a given bifurcation curve constrains a state-vector field geometry. A set of target bifurcations may also be represented as a vector field flow and thus inferring bifurcations becomes a problem in matching flows in vector sub-spaces. Note that this would involve matching direction but not the magnitude of the flow.

- smooth and match estimator methods [**?**]
- robots learning limit cycles [**?**]

Non-differentiable methods A large body of work is dedicated to finding conditions for fold-bifurcations, multistability and other

- bifurcation inference using Mixed Integer Nonlinear Programming [**?**]
- Random sampling and genetic algorithms [**?** **?**]

1.2 Preliminaries

Suppose we parameterise a set of differential equations for states $u \in \mathbb{R}^N$ with a function F_θ in an unknown parameter space $\theta \in \mathbb{R}^M$. We would like these differential equations to obey a set of target bifurcations $\mathcal{D} := \{p_1 \dots p_K\}$ along a known bifurcation parameter $p \in \mathbb{R}$. Let the differential equations be defined as

$$\partial_t u = F_\theta(z) \qquad \text{where} \quad z := (u, p) \quad F_\theta : \mathbb{R}^{N+1} \to \mathbb{R}^N \tag{1.1}$$

For a given set of parameters θ one could compute the set of predicted bifurcations $\mathcal{P}(\theta)$ using parameter continuation methods [**?** **?**]. Our goal is to find optimal parameters θ^* that match predictions $\mathcal{P}(\theta^*)$ to specified targets $\mathcal{D}$. We must design a suitable cost function L so that

$$\theta^* := \text{argmin}_\theta L(\theta|\mathcal{D}) \tag{1.2}$$

The optimal θ^* is not expected to always be unique, but is in general a manifold representing the space of qualitatively equivalent models. The cost function L must have some distance measure between the two sets $\mathcal{P}(\theta)$ and $\mathcal{D}$ with a bias that encourages equivalent cardinality $|\mathcal{P}(\theta)| = |\mathcal{D}|$. This is especially important in the case where there are no predictions $|\mathcal{P}(\theta)| = 0$.

For clarity, we guide the reader through the methods with the following minimal models that explore the space of saddle-nodes $F_\theta(z) = p + \theta_1 u + \theta_2 u^3$ and pitchforks $F_\theta(z) = \theta_1 + pu + \theta_2 u^3$. These minimal models are shown with targets $\mathcal{D}$ in Figure **??** and Figure **??** respectively. These figures show that the directional derivative of the determinant $\frac{d}{ds}\left|\frac{\partial F_\theta}{\partial u}\right|$ along the bifurcation curve s increases in the vicinity of bifurcations and that $\left|\frac{\partial F_\theta}{\partial u}\right| = 0$ at the bifurcations. This gives us our first hint of what should be optimised to increase the number of predictions $|\mathcal{P}(\theta)|$.

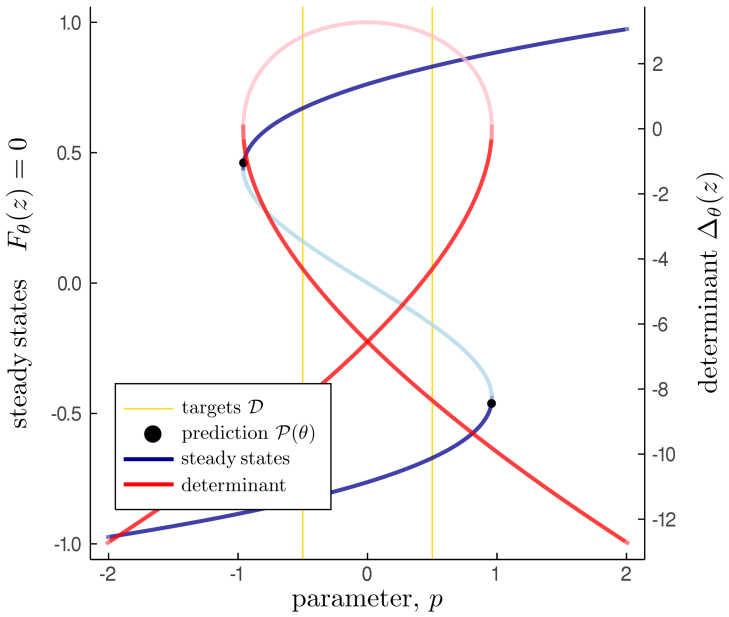

Figure 1.1: Saddle-node model $F_\theta(z) = p + \theta_1 u + \theta_2 u^3$ with set $\theta = (2, -1)$ and targets $\mathcal{D} = \{-1/2, 1/2\}$. Lighter shades indicate the determinant crossing zero for unstable solutions

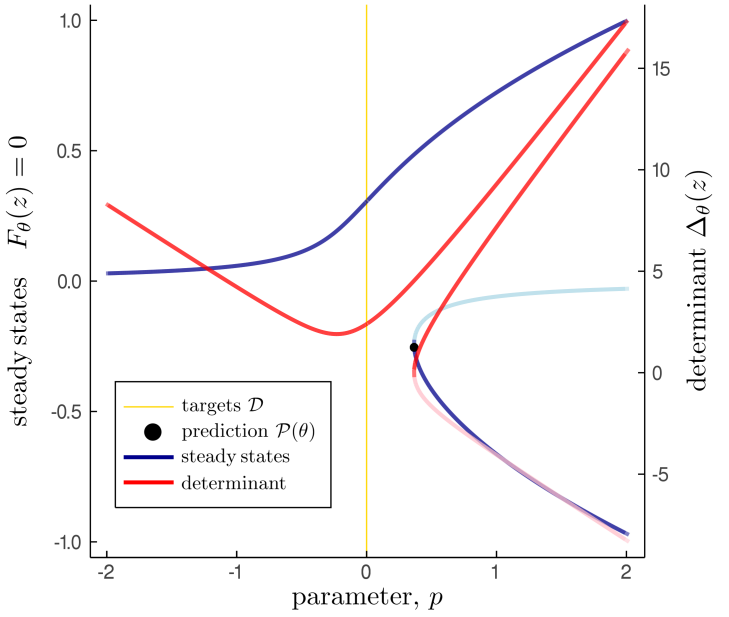

Figure 1.2: Pitchfork model $F_\theta(z) = \theta_1 + pu + \theta_2 u^3$ with set $\theta = (-1/10, -1)$ and target $\mathcal{D} = \{0\}$. Lighter shades indicate the determinant crossing zero for unstable solutions

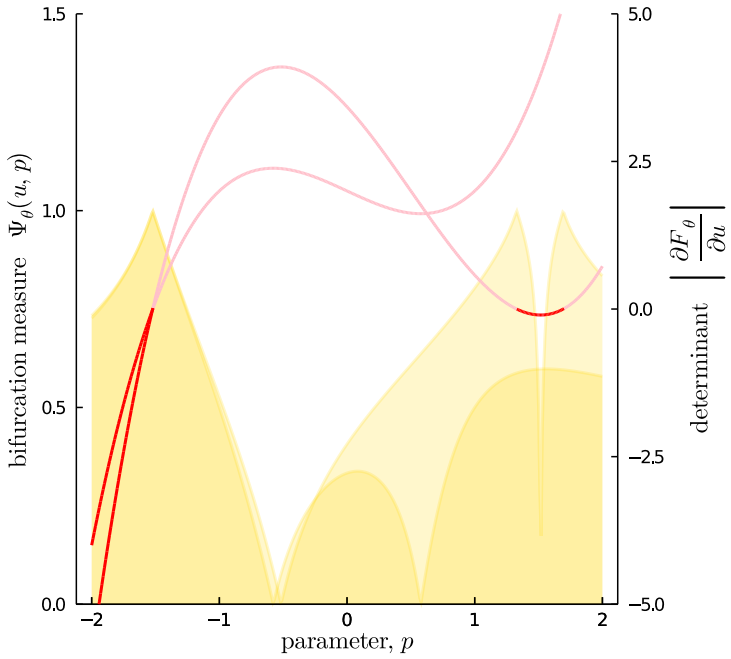

Figure 2.1: Measure $\Psi_\theta(z) \to 1$ as bifurcations form where $\left|\frac{\partial F_\theta}{\partial u}\right| = 0$

2 Proposed Method

2.1 Semi-supervised Cost Function

In order for predicted bifurcations $p \in \mathcal{P}(\theta)$ to match targets $p' \in \mathcal{D}$ we need to evaluate some form of error term $|p - p'|$ on the bifurcation curve defined by $F_\theta(z) = 0$. We expect the supervised term to be some mean over the targets $\mathcal{D}$ and predictions $\mathcal{P}(\theta)$. We choose a geometric mean over the predictions and an arithmetic mean over targets to account for cases where $|\mathcal{P}| \neq |\mathcal{D}|$ and discourage multiple predictions matching the same target

$$\frac{1}{|\mathcal{D}|} \sum_{p' \in \mathcal{D}} \left(\prod_{p \in \mathcal{P}(\theta)} |p - p'| \right)^{\frac{1}{|\mathcal{P}|+1}} \tag{2.1}$$

We can see from Figures **??** and **??** that predictions $\mathcal{P}(\theta)$ can be identified by looking for points p along the curve where the determinant $\left|\frac{\partial F_\theta}{\partial u}\right| = 0$. The supervised term can be re-written as an explicit integration over the curve $\int_{F_\theta(z)=0} \mathrm{dz}$ subject to constraint $\left|\frac{\partial F_\theta}{\partial u}\right| = 0$

$$\frac{1}{|\mathcal{D}|} \sum_{p' \in \mathcal{D}} \exp \left[\frac{1}{|\mathcal{P}|+1} \int_{F_\theta(z)=0,\, \left|\frac{\partial F_\theta}{\partial u}\right|=0} \mathrm{d}u\mathrm{d}p \log |p - p'| \right] \tag{2.2}$$

The supervised term is equal to one when there are no predictions $|\mathcal{P}| = 0$; the exponent is zero because the constraint $\left|\frac{\partial F_\theta}{\partial u}\right| = 0$ along the curve $F_\theta(z) = 0$ is never satisfied in the integration. It is equal to zero when $|\mathcal{P}| \geq |\mathcal{D}|$ and all targets are matched by at least one prediction. Note that we dropped the explicit dependency on θ in counting the predictions $|\mathcal{P}|$. While the dependence still exists, gradients of this term with respect to θ are zero everywhere, and not defined when the number of predictions changes; the smooth dependency on θ is now picked up by the constraint $\left|\frac{\partial F_\theta}{\partial u}\right| = 0$. We note that at bifurcation points the determinant go to zero $\left|\frac{\partial F_\theta}{\partial u}\right| = 0$ while its derivative with respect to the bifurcation curve s is finite $\frac{d}{ds}\left|\frac{\partial F_\theta}{\partial u}\right| \neq 0$. We can use these quantities to create a scale-invariant measure of the zero crossing density of the determinant. Figure **??** shows an example

of such a measure. Let us define the bifurcation density as

$$P(\theta) := \frac{1}{\int_{F_\theta(z)=0} \mathrm{d}u\mathrm{d}p} \int_{F_\theta(z)=0} \mathrm{d}u\mathrm{d}p \left(1 + \left| \frac{\left|\frac{\partial F_\theta}{\partial u}\right|}{\frac{d}{ds}\left|\frac{\partial F_\theta}{\partial u}\right|} \right| \right)^{-1} \tag{2.3}$$

This curvature can be maximised to encourage bifurcations and should be used in the unsupervised term to allow an optimiser to follow non-zero gradients in parameter regimes θ far away from a bifurcating regime $|\mathcal{P}| = 0$. The number of bifurcations must be reduced however, in cases where $|\mathcal{P}| > |\mathcal{D}|$. A term of the following form emerges

$$\big(|\mathcal{D}| - |\mathcal{P}|\big) f \circ K(\theta) \quad \text{where} \quad f(x) > 0 \quad f(-x) = f(x) \quad \lim_{x\to\infty} f(x) = 0 \tag{2.4}$$

In makes sense compose the curvature K with a positive definite symmetric function f that goes to zero as the curvature diverges. The pre-factor ensures that the gradients are always pushing optimisers towards a state where $|\mathcal{D}| = |\mathcal{P}|$. The remaining questions are: what should $f(0)$ be? How fast do we want the function to decay? These are implementation details that are discussed in section **??**. In order to compute the curvature a differentiable representation of the bifurcation curve is needed. The following sections **??** and **??** outline how this can be done.

2.2 Bifurcation Curves as Tangent Fields

Let each component of the vector function F_θ in the model (**??**) implicitly define a surface embedded in $\mathbb{R}^{N+1}$. Let's assume that the intersection of these N surfaces exists and is not null or degenerate, then the steady states of (**??**) must be a set of one dimensional space curves in $z \in \mathbb{R}^{N+1}$ defined by

$$F_\theta(z) = 0 \tag{2.5}$$

An expression for the field $T_\theta(z)$ tangent to the set of curves would allow us to take derivatives and integrals along them. Fortunately the tangent field can be constructed by ensuring it is perpendicular to the gradient ∂_z of each component of F_θ as illustrated by an example two component system in Figure **??**. The tangent field $T_\theta(z)$ can be constructed perpendicular to all gradient vectors using the properties of the determinant [**?**]

$$T_\theta(z) := \left| \begin{matrix} \hat{z} \\ \partial_z F_\theta \end{matrix} \right| \qquad T_\theta : \mathbb{R}^{N+1} \to \mathbb{R}^{N+1} \tag{2.6}$$

$$= \sum_{i=1}^{N+1} \hat{z}_i (-1)^{i+1} \left| \frac{\partial F_\theta}{\partial (z \setminus z_i)} \right| \tag{2.7}$$

where $\hat{z}$ is a collection of unit basis vectors in the $\mathbb{R}^{N+1}$ space and $\partial_z F_\theta$ is an $N \times (N+1)$ rectangular Jacobian matrix of partial derivatives and $z \setminus z_i$ denotes the N dimensional vector z with component z_i removed. This construction ensures that the dot product of this field any gradient of a component of F_θ

$$T_\theta(z) \cdot \partial_z f_\theta = \left| \begin{matrix} \partial_z f_\theta \\ \partial_z F_\theta \end{matrix} \right| \quad = 0 \quad \forall f_\theta \in F_\theta \tag{2.8}$$

since the determinant of any matrix with two identical rows or columns is zero. Note that the tangent field $T_\theta(z)$ is actually defined for all values of z where adjacent field lines trace out other level sets where $F_\theta(z) \neq 0$. Furthermore deformations with respect to θ are always orthogonal to the tangent

$$T_\theta(z) \cdot \frac{dT_\theta}{d\theta} = 0 \tag{2.9}$$

Figure **??** shows how the bifurcation curve defined by $F_\theta(z) = 0$ picks out one of many traces in tangent field $T_\theta(z)$ for the saddle and pitchfork. The tangent field $T_\theta(z)$ can always be analytically

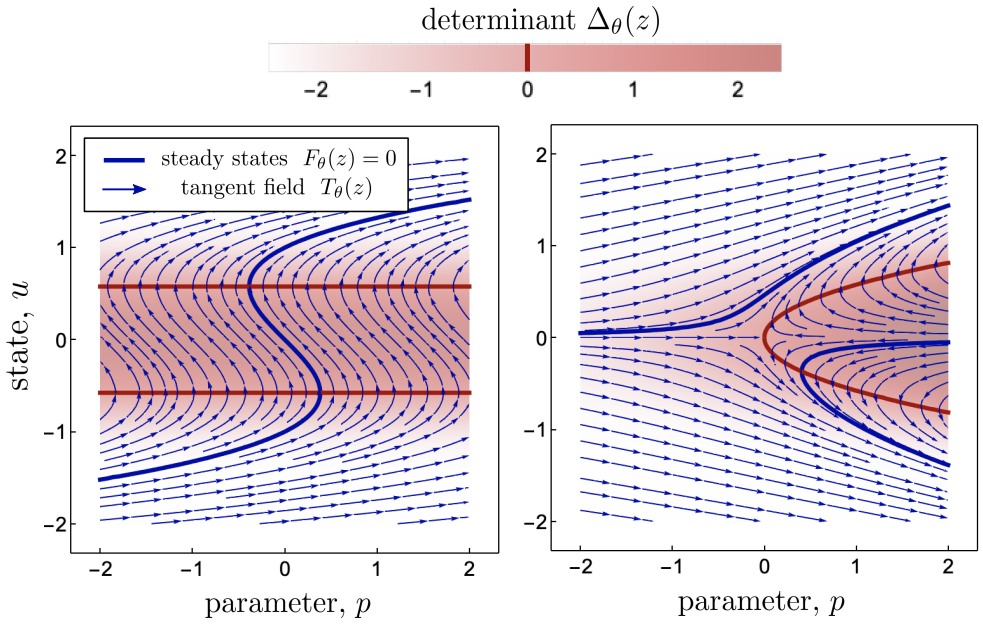

Figure 2.2: Left/Right : Determinant $\left|\frac{\partial F_\theta}{\partial u}\right|$ and tangent field $T_\theta(z)$ for the saddle-node/pitchfork models for some set values of θ revealing that $\left|\frac{\partial F_\theta}{\partial u}\right| = 0$ defines bifurcations

evaluated by taking the determinant in (**??**). We will proceed with calculations on $T_\theta(z)$ in the whole space z and pick out a single trace by solving $F_\theta(z) = 0$ later. For our two models

$$\underset{\text{saddle-node model}}{T_\theta(z) = \hat{u} - (\,3\theta_2 u^2 + \theta_1\,)\,\hat{p}} \qquad \underset{\text{pitchfork model}}{T_\theta(z) = u\hat{u} - (\,3\theta_2 u^2 + p\,)\,\hat{p}} \tag{2.10}$$

2.3 Evaluating Curvature of the Determinant

Bifurcation points are defined as values of z along the bifurcation curve where eigenvalues λ of the Jacobian $\partial_u F_\theta(z)$ cross either the real or imaginary axis in the complex plane. Restricting our case to $\lambda \in \mathbb{R}$ for now, it would be sufficient to look for values of z where one of the eigenvalues λ crosses zero. Therefore the determinant of the Jacobian

$$\left|\frac{\partial F_\theta}{\partial u}\right| \tag{2.11}$$

crossing zero can be used as a readout of whether a bifurcation has occurred. Figure **??** reveals this is indeed also true for any value z — not just along $F_\theta(z) = 0$ — for the saddle-node and pitchfork. The tangent field $T_\theta(z)$ only folds when $\left|\frac{\partial F_\theta}{\partial u}\right| = 0$. Plotting the value of the determinant along $F_\theta(z) = 0$ from Figure **??** would give rise to Figures **??** and **??**.

Consequently the curvature of the determinant along the tangent field $T_\theta(z)$ in the vicinity of bifurcations tends to increase. Fortunately this curvature can always be obtained in closed analytical form. The second directional derivative with respect to tangent field $T_\theta(z)$

$$:= \left(\partial_z \left(\partial_z \left|\frac{\partial F_\theta}{\partial u}\right| \cdot \hat{T}_\theta\right)\right) \cdot \hat{T}_\theta \quad \text{where} \quad \hat{T}_\theta := \frac{T_\theta}{|T_\theta|} \tag{2.12}$$

$$= \hat{T}_\theta \cdot \partial_z^2 \left|\frac{\partial F_\theta}{\partial u}\right| \cdot \hat{T}_\theta \quad + \quad \partial_z \left|\frac{\partial F_\theta}{\partial u}\right| \cdot \partial_z \hat{T}_\theta \cdot \hat{T}_\theta \tag{2.13}$$

The first term in the expression above is the usual result involving the Hessian matrix $\partial_z^2 \left|\frac{\partial F_\theta}{\partial u}\right|$ when taking second order directional derivatives. The second term involving the gradient $\partial_z \left|\frac{\partial F_\theta}{\partial u}\right|$ appears because $\hat{T}_\theta(z)$ is a function of z giving rise to a non-zero Jacobian matrix $\partial_z \hat{T}_\theta$. Using expressions

for tangent fields (**??**) the determinant curvatures for our models become

$$= -\underbrace{\frac{6\theta_2\left(\theta_1^2 - 9\theta_2^2u^4 + 1\right)}{\left(\left(\theta_1 + 3\theta_2u^2\right)^2 + 1\right)^2}}_{\text{saddle-node model}} \quad = -\underbrace{\frac{2u^2\left(p + 3\theta_2u^2\right)\left(9\theta_2\left(p + \theta_2u^2\right) + 1\right)}{\left(\left(p + 3\theta_2u^2\right)^2 + u^2\right)^2}}_{\text{pitchfork model}} \tag{2.14}$$

2.4 Implementation

Here provide a high-level view of Algorithm **??**. We use `BifurcationKit.jl` [**?**] to calculate bifurcation diagrams, which requires the rate function F_θ, its Jacobian J_θ which can be calculated using `ForwardDiff.jl` [**?**], and an initial guess u_0 at an initial parameter p_0. Let this algorithm be called by function `getSteadyStates` and return the steady states that form the integration region ∂S_θ and a new guess u_0 that is the true steady state at p_0.

The algorithm also takes hyperparameters β which will contain arclength step sizes, eigenvalue and Newton solver options and unsupervised correction factor λ. The adaptive hyperparameter update is done by `updateHyperparameters` and should depend on the current solutions ∂S_θ and the targets $\mathcal{D}$. The step sizes and solver options should adapt to the expected space of the solutions ∂S_θ so that the waiting time between iterations is minimised. The unsupervised correction factor λ should be large when there are no predicted bifurcations and decrease as the number of predicted bifurcations approaches the number of targets.

The evaluation of the cost gradient is done by `costGradient` using `ForwardDiff.jl` [**?**] which has a large repository of chain rules that it uses to correctly evaluate derivatives. At the time of writing this paper the repository does not include the Leibniz rule [**?**] for implicitly defined line integrals, and we therefore define additional rules that implement the results shown in Appendix **??** and **??**. The cost gradient is used by `updateParameters`, which can be any gradient-based optimiser such as ADAM or Momentum gradient decent, to update the parameters θ.

In practice the bifurcation curve is only evaluated in a finite region $p \in \Omega$ which includes the targets $\mathcal{D}$. We are now ready to write down the semi-supervised cost function for a bifurcation curve defined by $F_\theta(z) = 0$ in an observed region $p \in \Omega$.

Algorithm 1: Bifurcation Optimisation Loop

Inputs Function $F_\theta : \mathbb{R}^{N+1} \to \mathbb{R}^N$ for $u \in \mathbb{R}^N$ and $p \in \mathbb{R}$
target bifurcations $\mathcal{D} = \{p_1 \dots p_K : p \in \Omega\}$ and convergence tolerance ε
Output Optimised $\theta \in \mathbb{R}^M$ that satisfy targets $\mathcal{D}$
$u_0 \leftarrow \text{rand} \in \mathbb{R}^N \quad \theta \leftarrow \text{rand} \in \mathbb{R}^M$
$\beta \leftarrow$ `getHyperparameters`$(\mathcal{D})$
while $\text{tolerance}(\beta) > \varepsilon$ **do**
  $\partial S_\theta, u_0 \leftarrow$ `getSteadyStates`(F_θ, u_0, β)
  $\beta \leftarrow$ `updateHyperparameters`$(\beta, \partial S_\theta, \mathcal{D})$
  $\partial L \leftarrow$ `costGradient`$(F_\theta, \partial S_\theta, \mathcal{D}, \beta)$
  $\theta \leftarrow$ `updateParameters`$(\theta, \partial L, \beta)$
end

3 Experiments & Results

3.1 Minimal Models

Figures **??** and **??** show example optimisations of (θ_1, θ_2) for the minimal saddle-node and pitchfork models respectively. The magnitude of the correction factor $\lambda = 0$ when bifurcations are present and $\lambda \neq 0$ otherwise. Optimisation trajectories approach lines of global minima in the cost function, which represent a set of geometrically equivalent models, depicted in the right panels. Two bifurcation curves are geometrically equivalent if the number, type and locations of bifurcations match.

We can see that the geometrically equivalent lines are contained within larger basins defined by $\lambda = 0$ where the correct number and type of bifurcations are present $p \in \Omega$, but do not match the locations of targets $\mathcal{D}$. All models within this basin are in some sense topologically equivalent. This hierarchical classification allows us to identify the set of models that satisfy observed qualitative

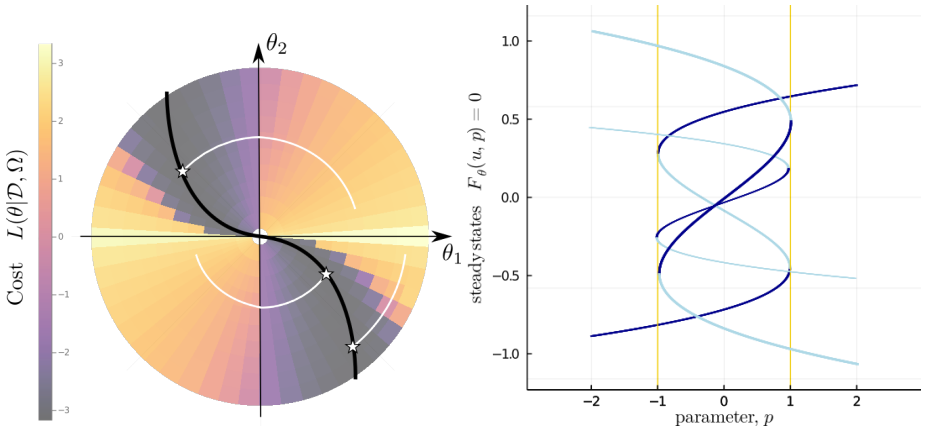

Figure 3.1: Saddle-node model $F_\theta(u,p) = p + \theta_1 u + \theta_2 u^3$ optimised with respect to targets $\mathcal{D} = \{-1, 1\}$ in observation region $\Omega \in [-2, 2]$. The right panel shows bifurcations diagrams for the three optima marked by stars on the left panel. The optimisation trajectories in white follow the gradient of the cost, approaching the black line of global minima in the left panel

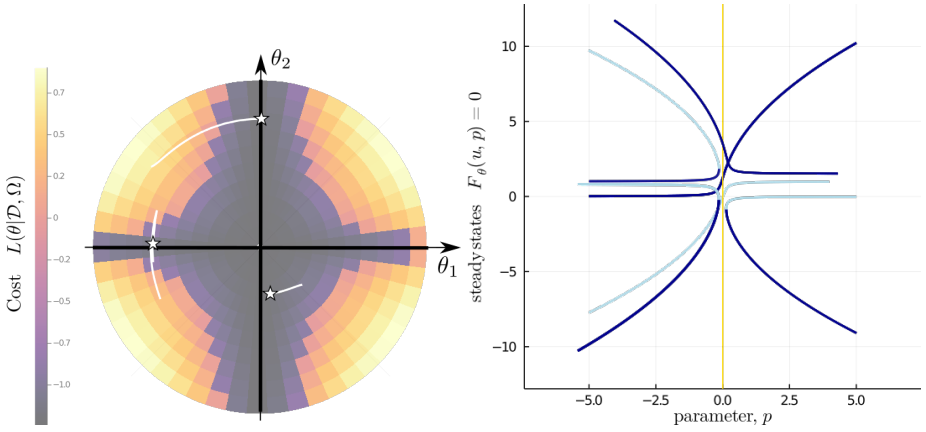

Figure 3.2: Pitchfork model $F_\theta(u,p) = \theta_1 + up + \theta_2 u^3$ optimised with respect to targets $\mathcal{D} = \{0\}$ in observation region $\Omega \in [-5, 5]$. The right panel shows bifurcations diagrams for the three optima marked by stars on the left panel. The optimisation trajectories in white follow the gradient of the cost, approaching the black lines of global minima in the left panel

behaviour [**?**] before any attempt at inferring kinetic parameters, which is done by choosing a model along the line of geometrically equivalent models.

Optimisation trajectories for the two minimal models appear mostly circumferential. This is because the models were set up such that the radial direction from the origin in θ space mostly scale kinetics whereas the circumferential direction changes the bifurcation topology. This suggests that the gradients of our cost function seek to change model geometry over kinetics.

3.2 Genetic Toggle Switch

In this section we optimise a more complicated model, popular in the synthetic biology community

$$\frac{du_1}{dt} = \frac{a_1 + (pu_2)^2}{1 + (pu_2)^2} - \mu_1 u_1, \quad \frac{du_2}{dt} = \frac{a_2 + (ku_1)^2}{1 + (ku_1)^2} - \mu_2 u_2 \tag{3.1}$$

3.3 Hyperparameters

3.4 Computational Complexity

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4 Conclusion & Broader Impact

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Appendix

A Supplementary Figures

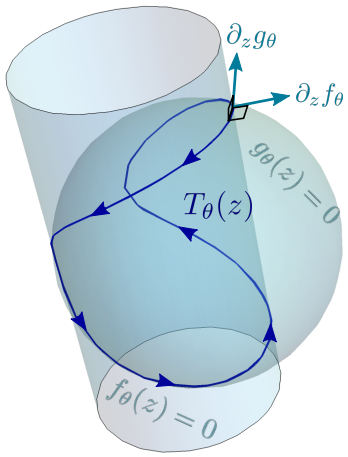

Figure A.1: Two implicit surfaces $f_\theta(z) = 0$ and $g_\theta(z) = 0$ in $\mathbb{R}^3$ intersecting to form a space curve which is tangent to field $T_\theta(z)$ and perpendicular to gradients $\partial_z f_\theta$ and $\partial_z g_\theta$

B Gradient of Space Curve

Suppose there exists a one dimensional space curve $\mathcal{C}(\theta)$ embedded in $z \in \mathbb{R}^{N+1}$ whos geometry changes depending on input parameters $\theta \in \mathbb{R}^M$. This curve could be open or closed and changes in θ could change the curve topology as well. Let the function $\gamma_\theta : \mathbb{R} \to \mathbb{R}^{N+1}$ be a parameterisation of the position vector along the curve within a fixed domain $s \in \mathcal{S}$. Note that the choice of

parameterisation is arbitrary and our results should not depend on this choice. Furthermore, if we parametrise the curve $\mathcal{C}(\theta)$ with respect to a fixed domain $\mathcal{S}$ the dependence on θ is picked up by the parameterisation $\gamma_\theta(s)$. We can write a line integral of any scalar function $L_\theta : \mathbb{R}^{N+1} \to \mathbb{R}$ on the curve as

$$L(\theta) := \int_{\mathcal{C}(\theta)} L_\theta(z)\,\mathrm{d}z = \int_{\mathcal{S}} L_\theta(z) \left|\frac{d\gamma_\theta}{ds}\right| \mathrm{d}s \;_{z=\gamma_\theta(s)} \tag{B.1}$$

where $\left|\frac{d\gamma_\theta}{ds}\right|$ is the magnitude of tangent vectors to the space curve and we remind ourselves that the integrand is evaluated at $z = \gamma_\theta(s)$. We would like to track how this integral changes with respect to θ. The total derivative with respect to θ can be propagated into the integrand [**?**] as long as we keep track of implicit dependencies

$$\frac{dL}{d\theta} = \int_{\mathcal{S}} \left|\frac{d\gamma_\theta}{ds}\right| \left(\frac{\partial L}{\partial \theta} + \frac{\partial L}{\partial z}\cdot\frac{dz}{d\theta}\right) + L_\theta(z)\frac{d}{d\theta}\left|\frac{d\gamma_\theta}{ds}\right| \mathrm{d}s \;_{z=\gamma_\theta(s)} \tag{B.2}$$

Here we applied the total derivative rule in the first term due to the implicit dependence of z on θ through $z = \gamma_\theta(s)$. Applying the chain rule to the second term

$$\frac{d}{d\theta}\left|\frac{d\gamma_\theta}{ds}\right| = \left|\frac{d\gamma_\theta}{ds}\right|^{-1} \frac{d\gamma_\theta}{ds}\cdot\frac{d}{d\theta}\left(\frac{d\gamma_\theta}{ds}\right) \tag{B.3}$$

By choosing an s that has no implicit θ dependence we can commute derivatives

$$\frac{d}{d\theta}\left(\frac{d\gamma_\theta}{ds}\right) = \frac{d}{ds}\left(\frac{d\gamma_\theta}{d\theta}\right) \quad \Rightarrow \quad \frac{d}{d\theta}\left|\frac{d\gamma_\theta}{ds}\right| = \left|\frac{d\gamma_\theta}{ds}\right|^{-1} \frac{d\gamma_\theta}{ds}\cdot\frac{d}{ds}\left(\frac{d\gamma_\theta}{d\theta}\right) \tag{B.4}$$

To proceed we note that the unit tangent vector can be written as an evaluation of a tangent field $\hat{T}_\theta(z)$ defined in the whole domain $z \in \mathbb{R}^{N+1}$ along the parametric curve $z = \gamma_\theta(s)$. The unit tangent field may disagree with the tangent given by $\frac{d\gamma_\theta}{ds}$ up to a sign

$$\hat{T}_\theta(z)\Big|_{z=\gamma_\theta(s)} = \pm\left|\frac{d\gamma_\theta}{ds}\right|^{-1}\frac{d\gamma_\theta}{ds} \tag{B.5}$$

and somehow this leads to numerically verified result

$$\frac{d}{d\theta}\left|\frac{d\gamma_\theta}{ds}\right| = \left|\frac{d\gamma_\theta}{ds}\right|\left(\hat{T}_\theta(z)\cdot\frac{\partial}{\partial z}\left(\frac{d\Gamma_\theta}{d\theta}\right)\cdot\hat{T}_\theta(z)\right)_{z=\gamma_\theta(s)} \tag{B.6}$$

C Deformation of Implicit Surfaces

It is possible to find the normal deformation of the implicit space curves due to changes in θ. This can be done by taking the total derivative of the implicit equation defining the level set

$$\frac{dF_\theta(z)}{d\theta} = \frac{\partial F}{\partial \theta} + \frac{\partial F}{\partial z}\cdot\frac{dz}{d\theta} \tag{C.1}$$

We can rearrange for $\frac{dz}{d\theta}$ using the Moore-Penrose inverse of the rectangular Jacobian matrix $\frac{\partial F}{\partial z}$ which appeared in equation (**??**). Since the level set is defined by $F_\theta(z) = 0$ the total derivative along the level set $dF_\theta(z) = 0$ and we arrive at an expression for the deformation field [**?**]

$$\frac{dz}{d\theta} = -\frac{\partial F}{\partial z}^\top \left(\frac{\partial F}{\partial z}\frac{\partial F}{\partial z}^\top\right)^{-1}\frac{\partial F}{\partial \theta} \tag{C.2}$$

The tangential component of the deformation field is not uniquely determined because there is no unique way of parameterising a surface. This is the subject of many computer graphics papers [**? ? ?**]. We are however not interested in the continuous propagation of a mesh - as is the subject of those papers. In fact we are looking for a deformation field that is orthogonal to the tangent vector $\hat{T}_\theta(z)\cdot\frac{dz}{d\theta} = 0$ for the space curve, and therefore letting the tangential component of the deformation equal zero is a valid choice and we can it instead of the parameterised deformation

$$\frac{d\gamma_\theta}{d\theta} \to \frac{dz}{d\theta} \tag{C.3}$$

To summarise we now have the gradient of our line integral only in terms of the implicit function defining the integration region.

$$\frac{dL}{d\theta} = \int_{F_\theta(z)=0} \frac{\partial L}{\partial \theta} + \frac{\partial L}{\partial z} \cdot \varphi_\theta(z) + L_\theta(z)\ \hat{T}_\theta(z) \cdot \frac{\partial \varphi}{\partial z} \cdot \hat{T}_\theta(z)\, \mathrm{d}z \tag{C.4}$$

$$\text{where} \quad \hat{T}_\theta(z) := \frac{T_\theta(z)}{|T_\theta(z)|} \qquad T_\theta(z) := \left| \frac{\hat{z}}{\partial_z F_\theta} \right| \qquad \varphi_\theta(z) := -\frac{\partial F}{\partial z}^\top \left(\frac{\partial F}{\partial z} \frac{\partial F}{\partial z}^\top \right)^{-1} \frac{\partial F}{\partial \theta} \tag{C.5}$$

We have settled on choosing normal deformations which we will call $\varphi_\theta(z)$. The above result can be seen a the generalised Leibniz rule [**?**] for the case of line integration regions. The last integrand term can be seen as the divergence the vector field $\varphi_\theta(z)$ projected onto the one dimensional space curve.

D Simplification of two-state model

Consider the general two-state model:

$$\frac{dv_i}{dt} = \frac{a_i + b_i(K_i v_j)^2}{1 + (K_i v_j)^2} - \mu_i v_i \tag{D.1}$$

Without loss of generality, we can rescale $v_i = U_i u_i$. Then,

$$\frac{U_i du_i}{dt} = \frac{a_i + b_i(K_i U_j u_j)^2}{1 + (K_i U_j u_j)^2} - \mu_i U_i u_i \tag{D.2}$$

Dividing through by U_i, we obtain

$$\frac{du_i}{dt} = \frac{\frac{a_i}{U_i} + \frac{b_i}{U_i}(K_i U_j u_j)^2}{1 + (K_i U_j u_j)^2} - \mu_i u_i \tag{D.3}$$

By choosing $U_i = b_i$, the model reduces to:

$$\frac{du_1}{dt} = \frac{\hat{a}_1 + (pu_2)^2}{1 + (pu_2)^2} - \mu_1 u_1, \quad \frac{du_2}{dt} = \frac{\hat{a}_2 + (ku_1)^2}{1 + (ku_1)^2} - \mu_2 u_2 \tag{D.4}$$

where $p = \frac{K_1}{\sqrt{b_1}}$ and $k = \frac{K_2}{\sqrt{b_2}}$.