

# Inference of Bifurcations with Differentiable Continuation

Gregory Szep<sup>1,2</sup>, Neil Dalchau<sup>2</sup> and Attila Csikasz-Nagy<sup>1,3</sup>

<sup>1</sup>*King's College London, UK*, <sup>2</sup>*Microsoft Research Cambridge, UK*, <sup>3</sup>*Pazmany  
Peter Catholic University, Budapest, Hungary*

October 7, 2020

## 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 likelihood term that is maximal when predicted bifurcations match the targets, and an unsupervised correction term that encourages bifurcations by maximising the curvature of 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, and an applied model from synthetic biology. This work forms the basis for the inference of more complicated bifurcation diagrams which may include Andronov-Hopf points or extended to partial differential equations.

## 1 Introduction and Motivation

### 1.1 Background

Backpropagation through differential equation solves has been a breakthrough over the past couple of years [1, 2] that enabled scalable parameter inference for differential equations. 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 ob-

served 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 dynamics – the bifurcation diagram – should precede any attempt at inferring kinetic parameters [3].

In this work we propose a gradient-based semi-supervised approach that focuses on fitting qualitative dynamics, defined by state space structures, rather than kinetics. We find that calculating the gradients, which inform the optimisation of parameters, shares the same computational complexity as the algorithm used to calculate the bifurcation diagram. We use a predictor-corrector method called deflated pseudo-arclength continuation [4, 5]. For 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 [6]. 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 [7].

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.

## 1.2 Similar and Related Works

- smooth and match estimator methods [8]
- bifurcation inference using Mixed Integer Nonlinear Programming optimization strategy [9]
- no other gradient-based bifurcation inference seen...?

## 2 Methodology

### 2.1 Problem Statement

Suppose we parameterise a set of differential equations for states  $u \in \mathbb{R}^N$  with a function  $F_\theta$  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  $p \in \mathbb{R}$ . The following sections outline how a gradient descent algorithm could encourage predicted bifurcations  $\mathcal{P}(\theta)$  coming from parameterised differential equations to match specified targets  $\mathcal{D}$ . This would allow us to design differential equations using high-level qualitative constraints and sample qualitatively equivalent models in regions of optimal  $\theta$ . Let the differential equations be defined as

$$\partial_t u = F_\theta(z) \quad \text{where } z := (u, p) \quad F_\theta : \mathbb{R}^{N+1} \rightarrow \mathbb{R}^N \quad (2.1)$$

First we view the steady states of this problem in terms of implicit space curves [10] in an augmented state space  $z \in \mathbb{R}^{N+1}$  which combines the state  $u$  and bifurcation parameter  $p$ . We then realise that the curvature of the Jacobian determinant  $\Delta_\theta(z) := |\partial F_\theta / \partial u|$  is maximised when bifurcations appear in the observed region of  $z$ . Luckily the curvature can be evaluated analytically [10]. The steady state regions of  $z$  are computed using a parameter continuation method [5, 4]. Finally a differentiable semi-supervised cost function  $L(\theta|\mathcal{D})$  is proposed that would allow gradient descent algorithms to efficiently encourage bifurcations and match their locations to targets  $\mathcal{D}$ .

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 1 and Figure 2 respectively. These figures show that the curvature of the determinant  $\Delta_\theta(z)$  increases in the vicinity of bifurcations and crosses zero at the bifurcations. This gives us our first hint of what should be optimised to encourage bifurcations. Success on these minimal models demonstrates the proof of concept, and success in practice is demonstrated on a four component genetic toggle switch model taken from synthetic biology.

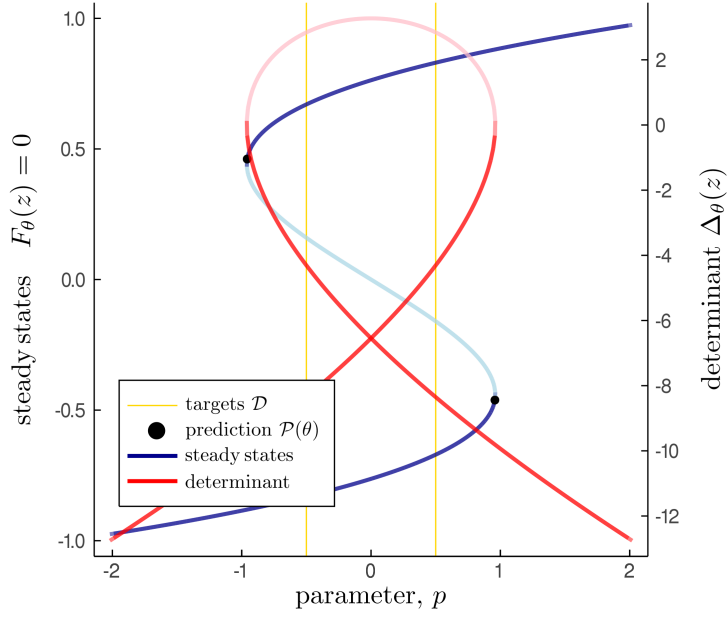


Figure 1: Saddle-node model  $F_\theta(z) = p + \theta_1 u + \theta_2 u^3$  with set  $\theta = (2, -1)$  and targets  $\mathcal{D} = \{-1/2, 1/2\}$ . Lighter shades indicate the determinant crossing zero for unstable solutions

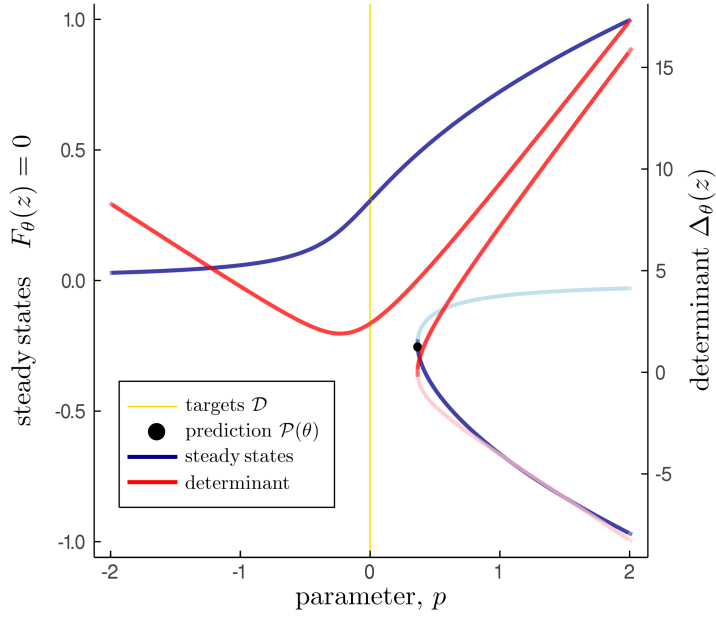


Figure 2: Pitchfork model  $F_\theta(z) = \theta_1 + pu + \theta_2 u^3$  with set  $\theta = (-1/10, -1)$  and target  $\mathcal{D} = \{0\}$ . Lighter shades indicate the determinant crossing zero for unstable solutions

## 2.2 Bifurcation Curves as Tangent Fields

Obtaining a bifurcation curve involves solving for the steady states of (2.1) along bifurcation parameter  $p$ . This is equivalent to finding the intersection between  $N$  surfaces where each surface is given by a component of the rate function  $F_\theta$ . Each surface is embedded in  $\mathbb{R}^{N+1}$  since the bifurcation parameter  $p$  is an additional unknown variable. Thus the co-dimension one bifurcation curve is an  $N + 1$  dimensional implicit space curve defined by the  $N$  equations

$$F_\theta(z) = 0 \quad (2.2)$$

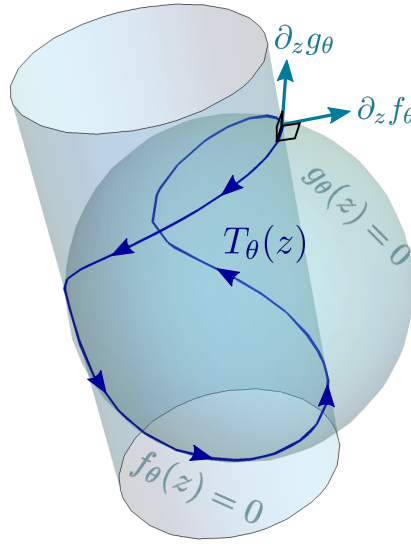


Figure 3: Two implicit surfaces  $f_\theta(z) = 0$  and  $g_\theta(z) = 0$  in  $\mathbb{R}^3$  intersecting to form a space curve which is tangent to field  $T_\theta(z)$  and perpendicular to gradients  $\partial_z f_\theta$  and  $\partial_z g_\theta$

An expression for the field  $T_\theta(z)$  tangent to the bifurcation curve would allow us to take derivatives and integrals along it. Fortunately the tangent field can be constructed by ensuring it is perpendicular to the gradient  $\partial_z$  of each component of  $F_\theta$  as illustrated by an example two component system in Figure 3. The tangent field  $T_\theta(z)$  can be constructed perpendicular to all gradient vectors using the properties of the determinant [10]

$$T_\theta(z) := \begin{vmatrix} \hat{z} \\ \partial_z F_\theta \end{vmatrix} \quad \text{where} \quad \hat{z} := (\hat{u}, \hat{p}) \quad T_\theta : \mathbb{R}^{N+1} \rightarrow \mathbb{R}^{N+1} \quad (2.3)$$

where  $\hat{z}$  is a collection of unit basis vectors in the  $\mathbb{R}^{N+1}$  space and  $\partial_z F_\theta$  is an  $N \times (N+1)$  augmented Jacobian matrix of partial derivatives including  $\partial_p F_\theta$  in the last column. This construction ensures that the dot product of this field any gradient of a component of  $F_\theta$

$$T_\theta(z) \cdot \partial_z f_\theta = \begin{vmatrix} \partial_z f_\theta \\ \partial_z F_\theta \end{vmatrix} = 0 \quad \forall f_\theta \in F_\theta \quad (2.4)$$

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 solutions where rates  $F_\theta(z) \neq 0$ .

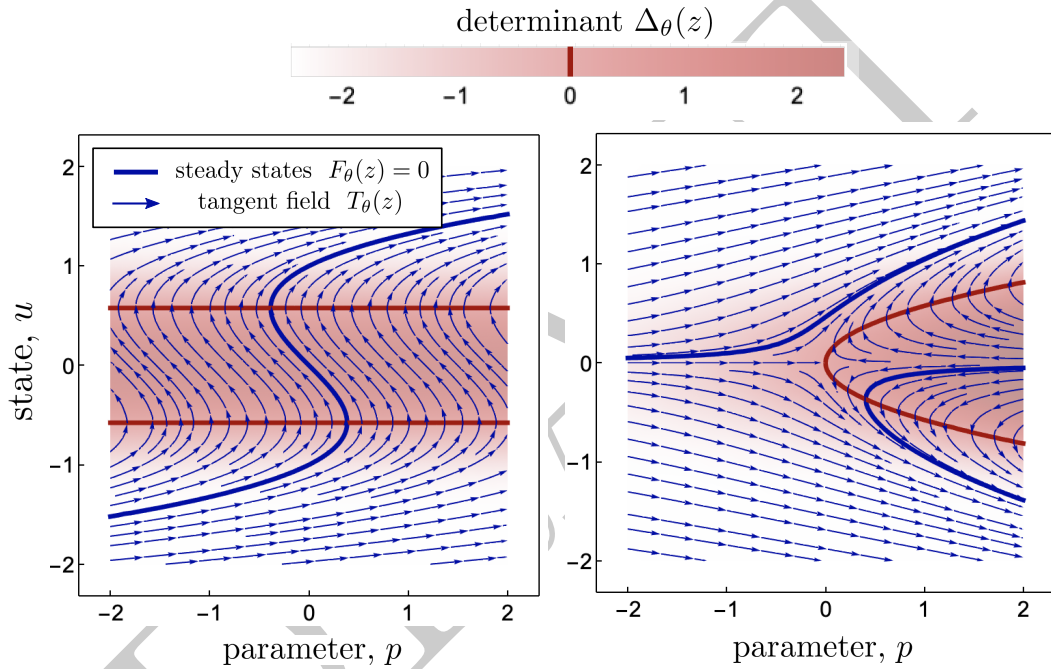


Figure 4: Left/Right : Determinant  $\Delta_\theta(z)$  and tangent field  $T_\theta(z)$  for the saddle-node/pitchfork models for some set values of  $\theta$  revealing that  $\Delta_\theta(z) = 0$  defines bifurcations

Figure 4 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 evaluated by taking the determinant in (2.3). 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

$$\begin{aligned} T_\theta(z) &= \hat{u} - (3\theta_2 u^2 + \theta_1) \hat{p} & T_\theta(z) &= u \hat{u} - (3\theta_2 u^2 + p) \hat{p} \\ &\text{saddle-node model} & &\text{pitchfork model} \end{aligned} \quad (2.5)$$

### 2.3 Evaluating Curvature of the Determinant

Bifurcation points are defined as values of  $z$  along the bifurcation curve where eigenvalues  $\lambda$  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  $\lambda$  crosses zero. Therefore the determinant of the Jacobian

$$\Delta_\theta(z) := |\partial_u F_\theta| \quad (2.6)$$

crossing zero can be used as a readout of whether a bifurcation has occurred. Figure 4 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  $\Delta_\theta(z) = 0$ . Plotting the value of the determinant along  $F_\theta(z) = 0$  from Figure 4 would give rise to Figures 1 and 2.

Consequently the curvature of the determinant  $\partial_{T_\theta}^2 \Delta_\theta$  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  $T_\theta(z)$

$$\partial_{T_\theta}^2 \Delta_\theta := \left( \partial_z \left( \partial_z \Delta_\theta \cdot \hat{T}_\theta \right) \right) \cdot \hat{T}_\theta \quad \text{where} \quad \hat{T}_\theta := \frac{T_\theta}{|T_\theta|} \quad (2.7)$$

$$= \hat{T}_\theta \cdot \partial_z^2 \Delta_\theta \cdot \hat{T}_\theta + \partial_z \Delta_\theta \cdot \partial_z \hat{T}_\theta \cdot \hat{T}_\theta \quad (2.8)$$

$$= \frac{|T_\theta|^2 T_\theta \cdot \partial_z^2 \Delta_\theta \cdot T_\theta + \partial_z \Delta_\theta \cdot (|T_\theta|^2 - T_\theta T_\theta^\top) \cdot \partial_z T_\theta \cdot T_\theta}{|T_\theta|^4} \quad (2.9)$$

The first term in the expression above is the usual result involving the Hessian matrix  $\partial_z^2 \Delta_\theta$  when taking second order directional derivatives. The second term involving the gradient  $\partial_z \Delta_\theta$  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 (2.5) and determinants for the saddle-node and pitchfork models the determinant curvatures respectively become

$$\partial_{T_\theta}^2 \Delta_\theta = - \frac{6\theta_2 (\theta_1^2 - 9\theta_2^2 u^4 + 1)}{((\theta_1 + 3\theta_2 u^2)^2 + 1)^2} \quad \text{saddle-node model} \quad \partial_{T_\theta}^2 \Delta_\theta = - \frac{2u^2 (p + 3\theta_2 u^2) (9\theta_2 (p + \theta_2 u^2) + 1)}{((p + 3\theta_2 u^2)^2 + u^2)^2} \quad \text{pitchfork model} \quad (2.10)$$

## 2.4 Semi-supervised Cost Function

This curvature of the determinant  $\partial_{T_\theta}^2 \Delta_\theta$  can be extremised to encourage bifurcations. We can treat this curvature as an unsupervised correction of magnitude  $\lambda$  to a supervised likelihood  $L_\theta(z|\mathcal{D})$ . For a particular bifurcation curve defined by  $F_\theta(z) = 0$  we can express the cost function as the negative log-likelihood

$$L(\theta|\mathcal{D}) := -\log \left( \int_{F_\theta(z)=0} L_\theta(z|\mathcal{D}) + \lambda K_\theta(z)^2 dz \right) \quad (2.11)$$

$$L_\theta(z|\mathcal{D}) := e^{-|\mathcal{D}-p|^2/\varepsilon - \Delta_\theta(z)^2} \quad K_\theta(z) := \frac{\partial^2 \Delta_\theta}{\partial T_\theta^2} \quad (2.12)$$

where we have chosen a Gaussian likelihood to bias optimisation of points in  $z$  where the determinant  $\Delta_\theta(z)$  is close to zero towards targets  $\mathcal{D}$ . Figure 5 shows how the likelihood  $L_\theta(z|\mathcal{D})$  is sharply peaked around the bifurcation points  $\mathcal{P}(\theta)$  and is maximal when they match the targets. The parameter  $\varepsilon$  controls the desired precision of the likelihood function and  $\lambda$  controls the strength of the unsupervised bias.

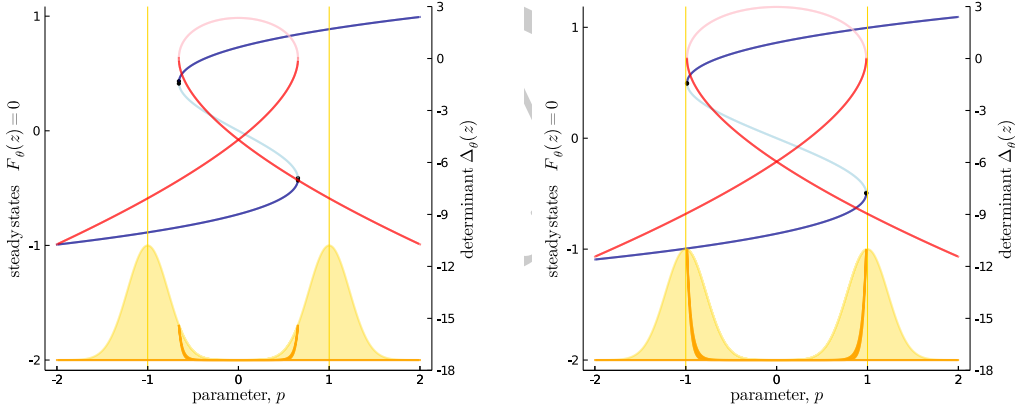


Figure 5: Likelihood  $L_\theta(z|\mathcal{D})$  shown in dark gold is sharply peaked around the bifurcation points  $\mathcal{P}(\theta)$  and is maximal when they match the targets (Right). The light gold envelope shows the maxima of the likelihood as the points  $\mathcal{P}(\theta)$  sweep across different values of  $p$

Applying the gradient operator with respect to  $\theta$  to the cost function, we find that not only the integrand but also the integration region needs to be differentiated. This requires the general



Leibniz rule [11] which yields

$$\frac{\partial L(\theta|\mathcal{D})}{\partial \theta} = - \frac{\int_{F=0} \text{div}\left(\frac{\partial z}{\partial \theta} (L + \lambda K^2)\right) + \frac{\partial L}{\partial \theta} + 2\lambda K \frac{\partial K}{\partial \theta} dz}{\int_{F=0} (L + \lambda K^2) dz} \quad (2.13)$$

where  $\frac{\partial z}{\partial \theta}$  is the velocity field of the integration region due to changes in  $\theta$ . Fortunately the velocity field of implicitly defined regions can be solved for analytically [12]. This is done by taking the total derivative of the implicit equation

$$\frac{dF_\theta(z)}{d\theta} = \frac{\partial F}{\partial \theta} + \frac{\partial F}{\partial z} \frac{\partial z}{\partial \theta} \quad (2.14)$$

We can rearrange for  $\frac{\partial z}{\partial \theta}$  using the Moore-Penrose inverse of the augmented Jacobian matrix  $\frac{\partial F}{\partial z}$  which appeared in equation (2.3). Since the surface is defined by  $F_\theta(z) = 0$  we arrive at

$$\frac{\partial z}{\partial \theta} = - \frac{\partial F}{\partial z}^\top \left( \frac{\partial F}{\partial z} \frac{\partial F}{\partial z}^\top \right)^{-1} \frac{\partial F}{\partial \theta} \quad (2.15)$$

which for our two minimal models yield

$$\frac{\partial z}{\partial \theta} = \underbrace{\frac{\hat{p} + \hat{u} (\theta_1 + 3\theta_2 u^2)}{1 + (\theta_1 + 3\theta_2 u^2)^2}}_{\text{saddle-node model}} \left( u \hat{\theta}_1 + u^3 \hat{\theta}_2 \right) \quad \frac{\partial z}{\partial \theta} = \underbrace{\frac{p \hat{u} + u (\hat{p} + 3\theta_2 u \hat{u})}{(p + 3\theta_2 u^2)^2 + u^2}}_{\text{pitchfork model}} \left( \hat{\theta}_1 + u^3 \hat{\theta}_2 \right) \quad (2.16)$$

Thus the integrands in the gradient (2.13) are completely analytic, leaving only region  $F_\theta(z) = 0$  to be evaluated numerically. Since these analytic forms can be pre-calculated, calculating the gradients shares the same computational complexity as calculating the steady states.

## 2.5 Optimisation Loop

Here provide a high-level view of Algorithm 1. We use `BifurcationKit.jl` [5] to calculate bifurcation diagrams, which requires the rate function  $F_\theta$ , its Jacobian  $J_\theta$ , 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  $\Omega$  and a new guess  $u_0$  that is the true steady state at  $p_0$ .

The algorithm also takes hyperparameters  $\beta$  which will contain arclength step sizes, eigenvalue and Newton solver options and unsupervised correction factor  $\lambda$ . The adaptive hyperparameter update is done by `updateHyperparameters` and should depend on the current solutions  $\Omega$  and the

targets  $\mathcal{D}$ . The step sizes and solver options should adapt to the expected space of the solutions  $\Omega$  so that the waiting time between iterations is minimised. The unsupervised correction factor  $\lambda$  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 (2.13) is done by `costGradient` using `Zygote.jl` [13] and requires the following analytical forms: the determinant of the Jacobian  $\Delta_\theta$ , its curvature  $K_\theta$  and the integration region velocity  $\frac{\partial z}{\partial \theta}$ . It also requires the integration region  $\Omega$  and the locations of targets  $\mathcal{D}$ . This gradient is used by `updateParameters`, which can be any gradient-based optimiser such as ADAM or Momentum gradient decent, to update the parameters  $\theta$ .

---

**Algorithm 1:** Bifurcation Optimisation Loop

---

**Inputs** Function  $F_\theta : \mathbb{R}^{N+1} \rightarrow \mathbb{R}^N$  for  $u \in \mathbb{R}^N$  and  $p \in \mathbb{R}$

target bifurcations  $\mathcal{D} = \{p : p \in [p_{\min}, p_{\max}]\}$  and convergence tolerance  $\varepsilon$

**Output** Optimised  $\theta \in \mathbb{R}^M$  that satisfy targets  $\mathcal{D}$

$J_\theta \leftarrow \partial F / \partial u$        $\Delta_\theta \leftarrow |\det J_\theta|$        $K_\theta \leftarrow \partial^2 \Delta_\theta / \partial T_\theta^2$

$\frac{\partial z}{\partial \theta} \leftarrow -\frac{\partial F}{\partial z}^\top \left( \frac{\partial F}{\partial z} \frac{\partial F}{\partial z}^\top \right)^{-1} \frac{\partial F}{\partial \theta}$

$u_0 \leftarrow \text{rand} \in \mathbb{R}^N$      $\theta \leftarrow \text{rand} \in \mathbb{R}^M$

$\beta \leftarrow \text{getHyperparameters}(\mathcal{D})$

**while**  $\text{tolerance}(\beta) > \varepsilon$  **do**

$\Omega, u_0 \leftarrow \text{getSteadyStates}(F_\theta, J_\theta, u_0, \beta)$

$\beta \leftarrow \text{updateHyperparameters}(\beta, \Omega, \mathcal{D})$

$\partial L \leftarrow \text{costGradient}(\Delta_\theta, K_\theta, \frac{\partial z}{\partial \theta}, \Omega, \mathcal{D})$

$\theta \leftarrow \text{updateParameters}(\theta, \partial L, \beta)$

**end**

---

## 3 Results

### 3.1 Model Synthesis

### 3.2 Model Selection

## 4 Conclusions and Extensions

- cell cycle / hopf
- application to structure  $\rightarrow$  function (Luca)
- pattern formation in pdes

## References

- [1] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural Ordinary Differential Equations,” *NIPS*, vol. 109, pp. 31–60, 6 2018.
- [2] C. Rackauckas, M. Innes, Y. Ma, J. Bettencourt, L. White, and V. Dixit, “DiffEqFlux.jl-A Julia Library for Neural Differential Equations,” tech. rep., 2019.
- [3] M. P. H. Stumpf and E. Roesch, “Parameter inference in dynamical systems with co-dimension 1 bifurcations,” 2019.
- [4] P. E. Farrell, C. H. L. Beentjes, and Å. Birkisson, “The computation of disconnected bifurcation diagrams,” *arXiv*, 3 2016.
- [5] R. Veltz, “PseudoArcLengthContinuation.jl,” 2019.
- [6] D. Avitabile, D. J. Lloyd, J. Burke, E. Knobloch, and B. Sandstede, “To snake or not to snake in the planar Swift-Hohenberg Equation,” *SIAM Journal on Applied Dynamical Systems*, vol. 9, pp. 704–733, 7 2010.
- [7] D. A. Aruliah, L. Van Veen, and A. Dubitski, “Algorithm 956: PAMPAC, a parallel adaptive method for pseudo-arclength continuation,” *ACM Transactions on Mathematical Software*, vol. 42, pp. 1–18, 2 2016.

- [8] S. Ranciati, C. Viroli, and E. Wit, “Bayesian Smooth-and-Match strategy for ordinary differential equations models that are linear in the parameters,” tech. rep., 2017.
- [9] I. Otero-Muras and J. R. Banga, “Optimization-based prediction of fold bifurcations in nonlinear ODE models,” *IFAC-PapersOnLine*, vol. 51, pp. 485–490, 1 2018.
- [10] R. Goldman, “Curvature formulas for implicit curves and surfaces,” in *Computer Aided Geometric Design*, vol. 22, pp. 632–658, Elsevier, 10 2005.
- [11] H. Flanders, “Differentiation Under the Integral Sign,” *The American Mathematical Monthly*, vol. 80, p. 615, 6 1973.
- [12] S. Jos and R. Schmidt, “On the velocity of an implicit surface,” *ACM Transactions on Graphics*, vol. 30, pp. 1–7, 5 2011.
- [13] M. Innes, “Don’t Unroll Adjoint: Differentiating SSA-Form Programs,” 10 2018.