Accelerating Simulation of Stiff Nonlinear Systems using Continuous-Time Echo State Networks

Ranjan Anantharaman
Massachusetts Institute of Technology
ranjanan@mit.edu

Yingbo Ma
Julia Computing Inc.

Shashi Gowda
Massachusetts Institute of Technology

Chris Laughman
Mitsubishi Electric Research Laboratories

Viral Shah
Julia Computing Inc.

Alan Edelman
Massachusetts Institute of Technology

Chris Rackauckas
Massachusetts Institute of Technology,
University of Maryland, Baltimore

Abstract

Modern design, control, and optimization often requires simulation of highly nonlinear models, leading to prohibitive computational costs. These costs can be amortized by evaluating a cheap surrogate of the full model. Here we present a general data-driven method, the continuous-time echo state network (CTESN), for generating surrogates of nonlinear ordinary differential equations with dynamics at widely separated timescales. We empirically demonstrate near-constant time performance using our CTESNs on a physically motivated scalable model of a heating system whose full execution time increases exponentially, while maintaining relative error of within 0.2 %. We also show that our model captures fast transients as well as slow dynamics effectively, while other techniques such as physics informed neural networks have difficulties trying to train and predict the highly nonlinear behavior of these models.

1 Introduction

Stiff nonlinear systems of ordinary differential equations, i.e., systems that are ill-conditioned in a computational sense, are widely prevalent throughout science and engineering [30, 26] and are characterized by dynamics with widely separated time scales. These systems require highly stable numerical methods to use large step-sizes reliably [8], and also tend to be computationally expensive to solve. Even with state-of-the-art simulation techniques, design, control, and optimisation of these systems remains intractable in many realistic engineering applications [1]. To address these challenges, we develop a data-driven technique, the continuous-time echo state network (CTESN), to obtain an approximation to the system (called a "surrogate") whose forward simulation time is relatively inexpensive while maintaining reasonable accuracy on these highly ill-conditioned models.

A popular class of methods to produce surrogates is projection based model order reduction, such as the proper orthogonal decomposition (POD) [1]. This method computes "snapshots" of the trajectory and uses the singular value decomposition of the linearization in order to construct a basis of a subspace of the snapshot space, and the model is remade with a change of basis. However, if the

Workshop on machine learning for engineering modeling, simulation and design @ NeurIPS 2020

system is very nonlinear, the computational complexity of this linearization-based reduced model can be almost as high as for the original model. One way to overcome this difficulty is through empirical interpolation methods [18]. Another recent method seeks to perform variable transformations to change the dynamics of the system before performing model order reduction on the transformed system [21]. However, such transformations could be system specific and it is not known whether such transformations exist for every system. Other methods to produce surrogates generally utilize the structural phonomena known about highly regular systems like partial differential equation discretizations [7].

Modern large-scale engineering models generated by symbolic systems such as Modelica [17, 31] and ModelingToolkit[1] can be quite heterogeneous and nonlinear without much repeated structure. Without the structure inherent in problems like partial differential equations, these problem-specific automated variable transformations and changes of bases could be hard to implement.

Numerical stiffness [28] and multiscale dynamics represent an additional challenge. While slow processes are relatively easy to capture, fast transients are not. Some model-order reduction techniques simply eliminate the fast modes and capture the slow modes accurately, such as the computational singular perturbation (CSP) method [10], which results in a non-stiff system. However, many systems (e.g., chemical reaction systems) require transient activations to properly arrive at the overarching dynamics. In this case, accurate representation of the fast dynamics can be required for appropriately reconstructing the constituent behavior [11]. We will demonstrate how a new surrogate technique, the CTESN, can overcome these difficulties and accurately capture highly nonlinear heterogeneous stiff systems in this paper.

2 Continuous-Time Echo State Networks

Echo State Networks (ESNs) are a framework for computation derived from recurrent neural network theory that maps signals into higher dimensional computational spaces through the dynamics of a fixed, non-linear system called a "reservoir". They are usually defined as high-dimensional discrete dynamical systems with an output readout layer which is then fit to an input signal [19]. While ESNs have traditionally been used as a stand-in for recurrent neural networks [14, 16], recent work has applied ESNs to learning chaotic systems [3, 4], nonlinear systems identification [13], bio-signal processing [15], and robot control [20],

Here we describe a new variant of ESNs which we call continuous-time echo state networks (CTESNs). Let N be the dimension of our model, and let P be a Cartesian space of parameters under which the model is expected to operate. The CTESN of dimension N_R is defined as

$$r' = \tanh(Ar + W_{in}x(p^*, t)) \tag{1}$$

$$x(t) = W_{out}r(t), \tag{2}$$

where A is a fixed sparse random graph of dimension $N_R \times N_R$ and W_{in} is a fixed random dense matrix of dimensions $N_R \times N$. Given these fixed values, the readout matrix W_{out} is the only trained portion of this network and is obtained through a least squares fit against the original timeseries.

To obtain a surrogate that predicts the dynamics at new physical parameters, the reservoir projection W_{out} is fit against many solutions at $p_1, \dots, p_n$, where n is the number of training data points sampled from the parameter space. Using these fits, an interpolating function $W_{out}(p)$ between the matrices can be trained. For the examples in this paper, we have sampled the high-dimensional spaces using a Sobol low-discrepancy sequence [27] and interpolated the W_{out} using a radial basis function provided by Surrogates.jl [2]. A prediction $\hat{x}(t)$ for a parameter vector $\hat{p}$ that is not in the training set would simply be:

$$\hat{x}(t) = W_{out}(\hat{p})r(t) \tag{3}$$

A strong advantage of our method is its ease of implementation and ease of training. Global L_2 fitting via stabilized methods like SVD are robust to ill-conditioning, alleviating many of the issues encountered when attempting to build neural surrogates of such equations. Also note that in this particular case, the readout matrix is fit against the same reservoir time series. Prediction does not need to simulate the reservoir, providing an extra acceleration.

[1] https://github.com/SciML/ModelingToolkit.jl/

[2] https://github.com/SciML/Surrogates.jl

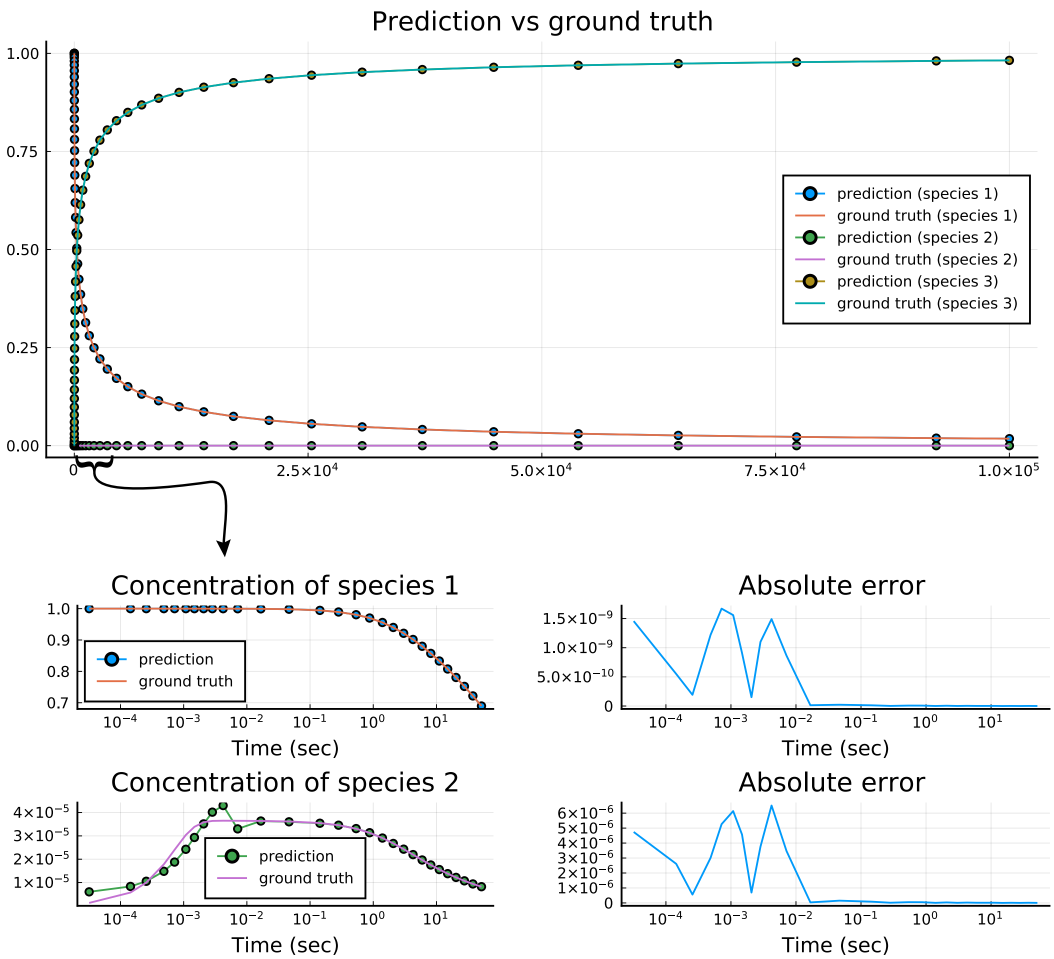

Figure 1: **Training performance on the the Robertson equations**. We see that our network is able to reproduce the dynamics, capturing the slow (second row) and fast transients (third row) with low error

3 Case Studies

3.1 Robertson Equations and High Stiffness

We first consider Robertson's chemical reaction equations:

$$\dot{y_1} = -0.04y_1 + 10^4 y_2 \cdot y_3 \tag{4}$$

$$\dot{y_2} = 0.04y_1 - 10^4 y_2 \cdot y_3 - 3 \cdot 10^7 y_2^2 \tag{5}$$

$$\dot{y_3} = 3 \cdot 10^7 y_2^2 \tag{6}$$

where y_1, y_2, y_3 are the concentration of three chemical reaction species. This system has widely separated time constants $(0.04, 10^4, 3 \cdot 10^7)$, is well known to be very stiff [9, 25, 24] and is used as an example to evaluate integrators of stiff ODEs [12]. Finding an accurate surrogate for this system is difficult because it needs to capture both the stable slow reacting system and the fast transients. Additionally, the surrogate needs to be consistent with this system's implicit physical constraints, such as the conservation of matter $(y_1 + y_2 + y_3 = 1)$ and positivity of the variables $(y_i > 0)$, in order to provide a stable solution. Figure 1 shows that training our CTESN yields good performance for both the larger and smaller quantities while accurately capturing the fast transients and respecting

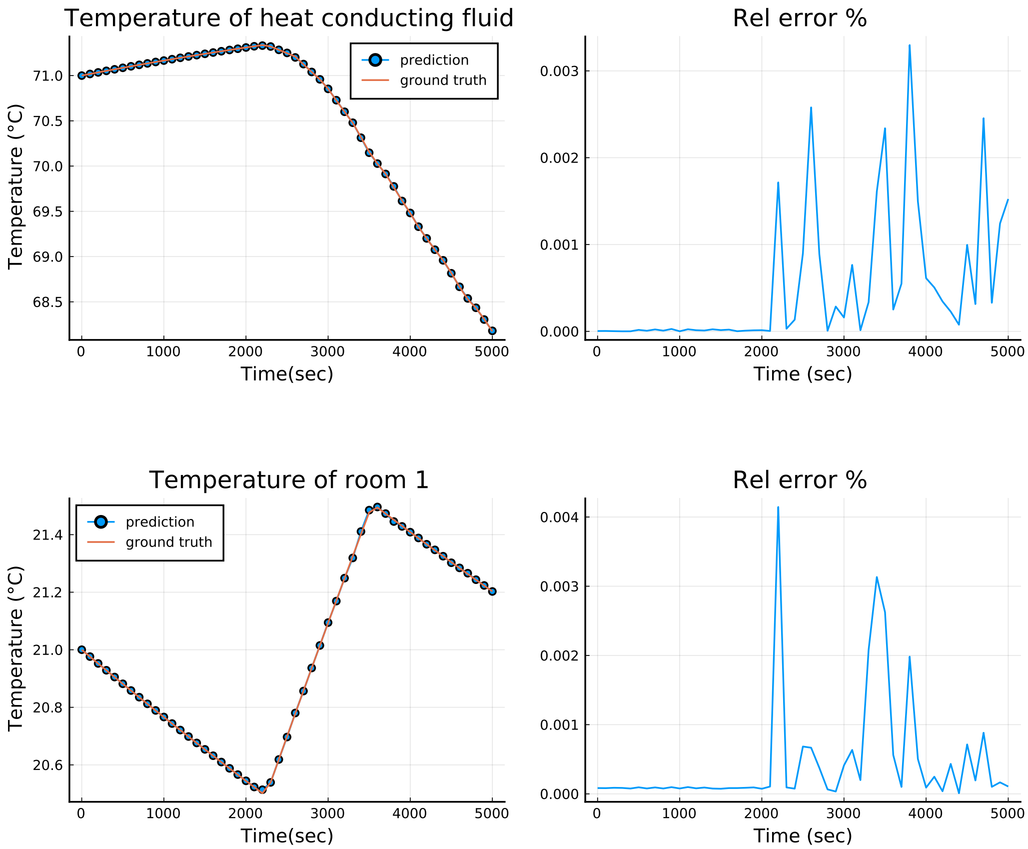

Figure 2: **TEST performance on scalable heating system with 10 rooms.** When tested with parameters it has not seen in training, our surrogate is able to reproduce the behaviour of the system to within 0.01% error.

physical constraints. Thus, for any set of parameters, if we know the approximate reading out matrix for the reservoir, we can achieve accurate predictions with $< 1\%$ error.

To further demonstrate the difficulty of this equation, we used the NeuralPDE.jl [3] physics-informed neural network (PINN) training framework to attempt to solve these equations using PINNs. PINNs are a recent data-driven method to obtain surrogates for partial differential equations by directly training the neural network to be the solution of the differential equation [23]. This software has been verified to automatically accurately solve many of the equations expounded upon in [23], including nonlinear partial equations like the Burger's equation, demonstrating its accuracy and the robustness of the implementation. However, when applied to Equation 4, the PINN training infrastructure was unable to adequately train neural networks to the solution the equation, never lowering below a relative error of 100% due to difficulties in capturing the short transient. The reason for the difficulty can be attributed to recently identified results in gradient pathologies in the training arising from stiffness [29]. With a highly ill-conditioned Hessian in the training process due to the stiffness of the equation, it would take an intractably long time to train a PINN surrogate on this problem with current methods. We note that in general, stiff systems of this form may be hard to capture by neural networks directly as they show a bias towards low frequency functions [22].

[3]https://github.com/SciML/NeuralPDE.jl

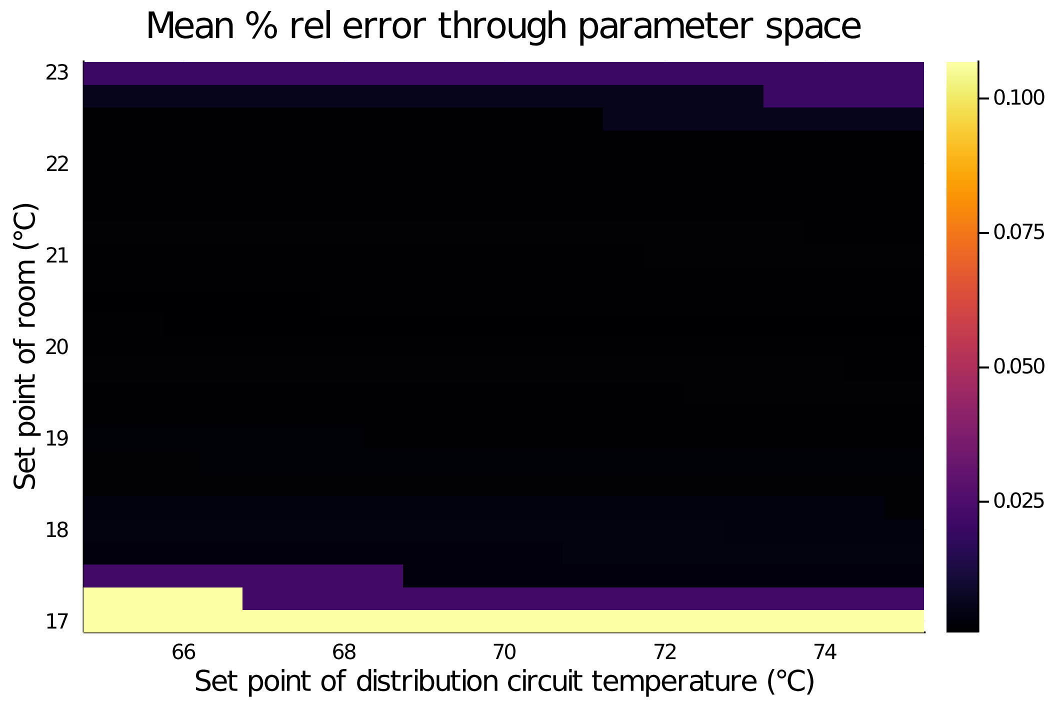

Figure 3: **Reliability of surrogate through parameter space.** We sampled our grid at over 500 grid points and plotted a heatmap of test error through our parameter space. We find our surrogate performs reliably even at the border of our space with error within 0.1%

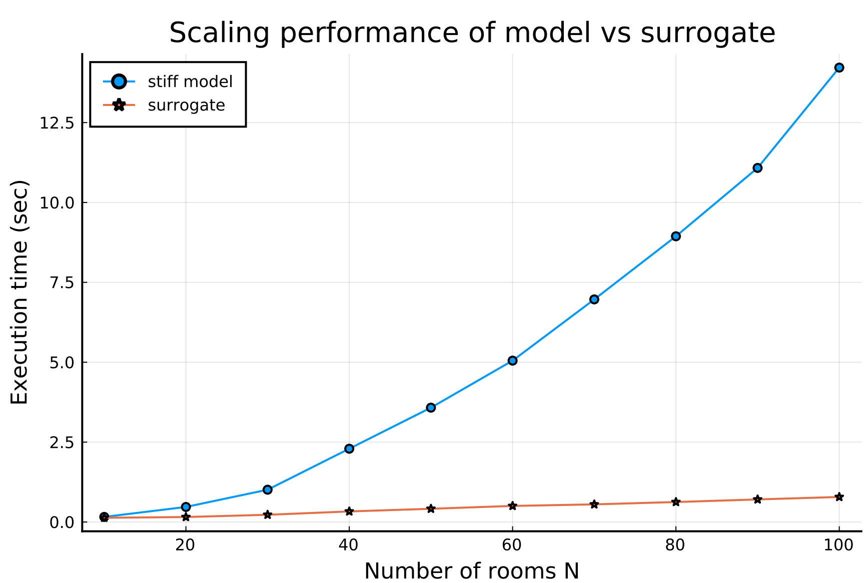

Figure 4: **Scaling performance of surrogate on heating system.** Our surrogate's runtime grows in near-constant time while the benchmark problem shows exponential growth in runtime.

3.2 Stiffly-Aware Surrogates of HVAC Systems

Our second test problem is a simplified, lumped-parameter model of a heating system with a central heater supplying heat to several rooms through a distribution network. The resulting system of equations is very stiff and has localized activity. This provides a useful test for our method, exhibiting stiff dynamics, and is a useful scalable benchmark used in the engineering community [2].

The size of the heating system is scaled by a parameter N which refers to the number of users/rooms. This produces a system with $2N + 1$ equations. This "scalability" lets us test how our method scales. To train the surrogate, we define a parameter space P under which we expect it to operate. First, we assume set point temperature of each room to be between $17°C$ to $23°C$, which is what we expect users to set their room temperature to. Each room is warmed by a heat conducting fluid, whose set point is likely between $65°C$ and $75°C$. Thus the parameter space over which we expect our surrogate to work is the rectangular space denoted by $[17°C, 23°C] \times [65°C, 75°C]$. Figure 2 demonstrates that the training technique is accurately able to find matrices W_{out} which capture the stiff system within 0.01% error. Figure 3 demonstrates that the interpolated $W_{out}(p)$ is able to adequately capture the dynamics throughout the trained parameter space. Lastly, Figure 4 demonstrates the almost $O(1)$ cost of the surrogate evaluation, where at the high end the surrogate accelerates a 201 dimensional stiff ODE system by over 15x.

4 Discussion, Conclusion & Future Work

We present CTESNs, a data-driven method for generating surrogates of nonlinear ordinary differential equations with dynamics at widely separated timescales. Our method maintains accuracy for different parameters in a chosen parameter space, and shows favourable scaling with system size.

Known limitations: Our method utilizes the continuous nature of differential equation solutions. Hybrid dynamical systems, such as those with event handling [5], can introduce discontinuities into the system which will not be adequately captured by the current method. Further extensions to the method will be required to handle both derivative discontinuities and events present in Filippov dynamical systems [6].

Reliability: In this work we have not proven any error bounds for our method, and while the empirical results are promising, some formal intuition on its reliability is a possible area of further study. In addition, more testing needs to be done on CTESNs under controls and as components within larger systems.

Performance: While we show promising scaling results, we have not shown tight bounds on execution time. For the surrogate on the heating system, we do not need solve the reservoir ODE every time we predict a time series because the surrogate has been trained for various parameters against the same reservoir. In other words, the reservoir ODE itself has no dependence on the parameter space (only the readout matrix does). Hence, we can simply return the solution to the reservoir ODE after training and just use it directly for prediction. However, this approach is memory-intensive: the reservoir time series of dimensions and the dense readout matrices of dimensions $N \times N_R$ are both large. In addition, for CTESN formulations that take in input forcing functions or have some dependence on parameters, the reservoir would need to be simulated every single time a prediction is made. Future work will comment on other such formulations. We do note in passing that simulating the reservoir is quite fast in practice as it is non-stiff, and thus techniques which regenerate reservoirs on demand will likely not incur a major run time performance cost.

Opportunities: Since CTESNs are largely data driven (the physics dependence is largely driven by using a candidate solution as "input" to the reservoir), an opportunity exists to train surrogates of black-box systems. Further opportunities could explore utilizing more structure within equations for building a more robust CTESN or decrease the necessary size of the reservoir.

Broader Impact

This section does not apply to our work.

Acknowledgement

The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0001222. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] Peter Benner, Serkan Gugercin, and Karen Willcox. "A survey of projection-based model reduction methods for parametric dynamical systems". In: *SIAM review* 57.4 (2015), pp. 483–531.

[2] Francesco Casella. "Simulation of large-scale models in Modelica: State of the art and future perspectives". In: *Proceedings of the 11th International Modelica Conference*. 2015, pp. 459–468.

[3] Ashesh Chattopadhyay et al. "Data-driven prediction of a multi-scale Lorenz 96 chaotic system using a hierarchy of deep learning methods: Reservoir computing, ANN, and RNN-LSTM". In: (2019).

[4] Nguyen Anh Khoa Doan, Wolfgang Polifke, and Luca Magri. "Physics-informed echo state networks for chaotic systems forecasting". In: *International Conference on Computational Science*. Springer. 2019, pp. 192–198.

[5] D Ellison. "Efficient automatic integration of ordinary differential equations with discontinuities". In: *Mathematics and Computers in Simulation* 23.1 (1981), pp. 12–20.

[6] Aleksei Fedorovich Filippov. *Differential equations with discontinuous righthand sides: control systems*. Vol. 18. Springer Science & Business Media, 2013.

[7] Michalis Frangos et al. "Surrogate and reduced-order modeling: a comparison of approaches for large-scale statistical inverse problems [Chapter 7]". In: (2010).

[8] C William Gear. "Numerical initial value problems in ordinary differential equations". In: *nivp* (1971).

[9] Matthias K Gobbert. "Robertson's example for stiff differential equations". In: *Arizona State University, Technical report* (1996).

[10] M Hadjinicolaou and Dimitris A Goussis. "Asymptotic solution of stiff PDEs with the CSP method: the reaction diffusion equation". In: *SIAM Journal on Scientific Computing* 20.3 (1998), pp. 781–810.

[11] Adrien Henry and Olivier C Martin. "Short relaxation times but long transient times in both simple and complex reaction networks". In: *Journal of the Royal Society Interface* 13.120 (2016), p. 20160388.

[12] ME Hosea and LF Shampine. "Analysis and implementation of TR-BDF2". In: *Applied Numerical Mathematics* 20.1-2 (1996), pp. 21–37.

[13] Herbert Jaeger. "Adaptive nonlinear system identification with echo state networks". In: *Advances in neural information processing systems*. 2003, pp. 609–616.

[14] Herbert Jaeger et al. "Optimization and applications of echo state networks with leaky-integrator neurons". In: *Neural networks* 20.3 (2007), pp. 335–352.

[15] Dhireesha Kudithipudi et al. "Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing". In: *Frontiers in neuroscience* 9 (2016), p. 502.

[16] Mantas Lukoševičius and Herbert Jaeger. "Reservoir computing approaches to recurrent neural network training". In: *Computer Science Review* 3.3 (2009), pp. 127–149.

[17] Sven Erik Mattsson, Hilding Elmqvist, and Martin Otter. "Physical system modeling with Modelica". In: *Control Engineering Practice* 6.4 (1998), pp. 501–510.

[18] VB Nguyen et al. "Model reduction for reacting flow applications". In: *International Journal of Computational Fluid Dynamics* 28.3-4 (2014), pp. 91–105.

[19] Mustafa C Ozturk, Dongming Xu, and José C Prıncipe. "Analysis and design of echo state networks". In: *Neural computation* 19.1 (2007), pp. 111–138.

[20] Athanasios Polydoros, Lazaros Nalpantidis, and Volker Krüger. "Advantages and limitations of reservoir computing on model learning for robot control". In: *IROS Workshop on Machine Learning in Planning and Control of Robot Motion, Hamburg, Germany*. 2015.

[21] Elizabeth Qian et al. “Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems”. In: *Physica D: Nonlinear Phenomena* 406 (2020), p. 132401.

[22] Nasim Rahaman et al. “On the spectral bias of neural networks”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5301–5310.

[23] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378 (2019), pp. 686–707.

[24] HH Robertson. “Numerical integration of systems of stiff ordinary differential equations with special structure”. In: *IMA Journal of Applied Mathematics* 18.2 (1976), pp. 249–263.

[25] HH Robertson and J Williams. “Some properties of algorithms for stiff differential equations”. In: *IMA Journal of Applied Mathematics* 16.1 (1975), pp. 23–34.

[26] Lawrence F Shampine and Charles William Gear. “A user’s view of solving stiff ordinary differential equations”. In: *SIAM review* 21.1 (1979), pp. 1–17.

[27] Ilya M Sobol’ et al. “Construction and comparison of high-dimensional Sobol’generators”. In: *Wilmott* 2011.56 (2011), pp. 64–79.

[28] Gustaf Söderlind, Laurent Jay, and Manuel Calvo. “Stiffness 1952–2012: Sixty years in search of a definition”. In: *BIT Numerical Mathematics* 55.2 (2015), pp. 531–558.

[29] Sifan Wang, Yujun Teng, and Paris Perdikaris. “Understanding and mitigating gradient pathologies in physics-informed neural networks”. In: *arXiv preprint arXiv:2001.04536* (2020).

[30] Gerhard Wanner and Ernst Hairer. *Solving ordinary differential equations II*. Springer Berlin Heidelberg, 1996.

[31] Michael Wetter et al. “Modelica buildings library”. In: *Journal of Building Performance Simulation* 7.4 (2014), pp. 253–270.