# Lipschitz-Constrained Neural Networks for Robust Chemical Kinetics Modelling

L. Castellanos[1,2] *, R. S. M. Freitas[3], R. Malpica Galassi[4], A. Parente[2], F. Contino[1]

Contact person: luisa.castellanos@uclouvain.be

[1]Universite Catholique de Louvain, Ecole polytechnique de Louvain, Institute of Mechanics, Materials and Civil Engineering, Ottignies-Louvain-la-Neuve, Belgium

[2]Universite Libre de Bruxelles, Ecole polytechnique de Bruxelles, Aero-Thermo-Mechanics Laboratory, Brussels, Belgium

[3]Queen Mary University London, School of Engineering and Materials Science, Centre for Sustainable Engineering, London, United Kingdom

[4]Sapienza Universitá di Roma, Facoltá di Ingegneria, Dipartimento di Ingegneria Meccanica e Aerospaziale, Rome, Italy

**Objective**

Test some strategies to improve neural networks robustness as time integrators.

## Introduction

- Combustion Machine Learning allows to predict thermochemical states efficiently [1].
- These predictions are implemented through neural networks, which are sensitive to small deviations of the input variables.
- **Small errors** can amplify when the model works as a time integrator (autoregression), leading to **potential blow ups** and wrong final states [5].
- **Robustness** is the characteristic that protects neural networks from such phenomenon, we explore training processes alternatives aiming for robustness improvement.

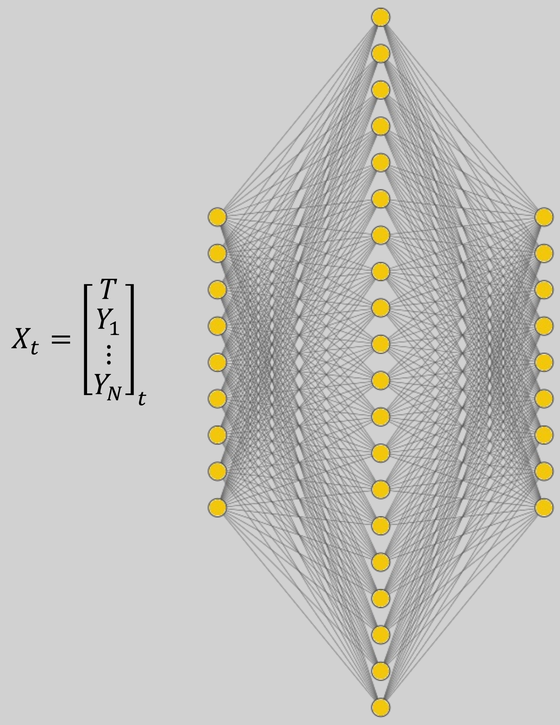
## Methodology

- **Lipschitz constrained** neural networks are of interest [4], as well as other standard **regularization techniques**, as the L2-regularization process [2].
- The Lipschitz constant is the maximum rate at which a continuous function can grow, with higher values producing blow ups. The Lipschitz constant is usually defined as:

$$L < \frac{|F(x_1) - F(x_2)|}{|x_1 - x_2|}$$

- The **L2-regularization** consists in adding the squared value of the networks' weights to the loss function:

$$L_2 = \lambda \sum_{i=1}^{P} \omega_i^2$$




$$\mathcal{L} = MSE + L_2 + L_{ct} + L_M$$

- The **Lipschitz** constant is applied in terms of an **implicit gradient** loss term.

$$L_{ct} = \alpha \left( \frac{|F(\tilde{x}_1) - F(x_2)|}{|x_1 - x_2|} - \frac{|F(x_1) - F(x_2)|}{|x_1 - x_2|} \right)$$

$$L_{ct} = \alpha \left( \frac{|F(\tilde{x}_1) - F(x_1)|}{|x_1 - x_2|} \right)$$

$$L_{ct} = \gamma(|F(\tilde{x}_1) - F(x_1)|)$$

- All the considered models include a **mass conservation** term in the loss function:

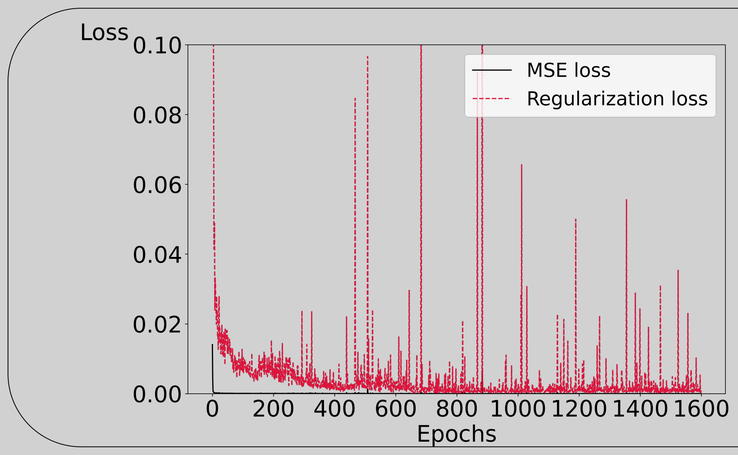
$$L_M = \beta \left( \sum_{i=1}^{N} \tilde{Y}_i - \sum_{i=1}^{N} Y_i \right)$$

- The different approaches are implemented for one-layer networks with 600 neurons, using the Pytorch Lightning framework .
- As proof of concept, the San Diego hydrogen [4] combustion mechanism is used with a temperature sampling [1150,1200K] with constant equivalence ratio $\phi = 1$.

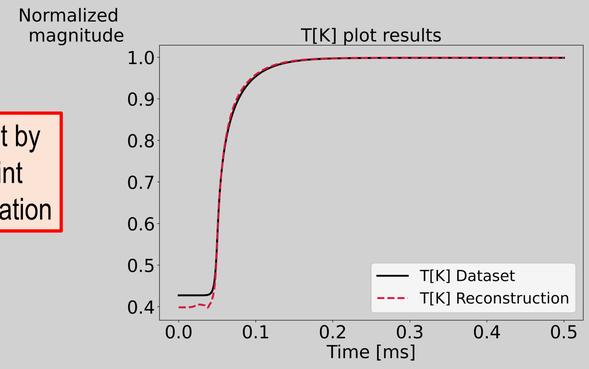
$$\Delta t = 1x10^{-8} s$$

## Results

$\mathcal{L} = MSE + L_M$



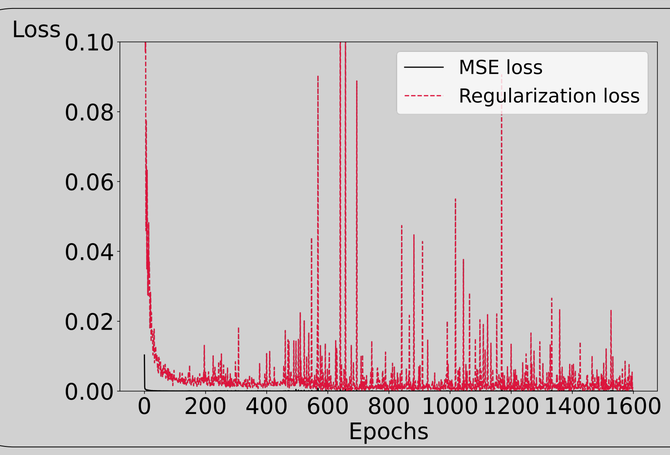

Point by point evaluation



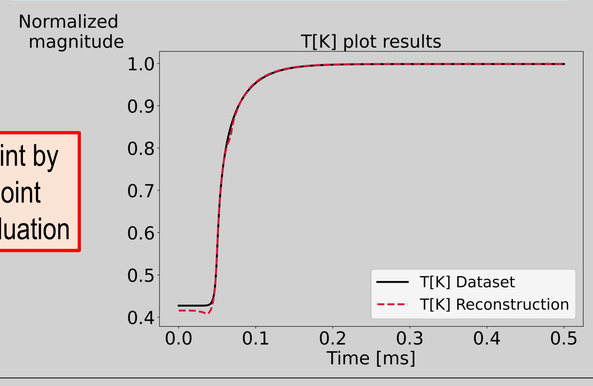

After a couple of steps in autoregression, the model blows-up
$\beta = 0.1$

$R^2 = 0.989$

$\mathcal{L} = MSE + L_{ct} + L_M$



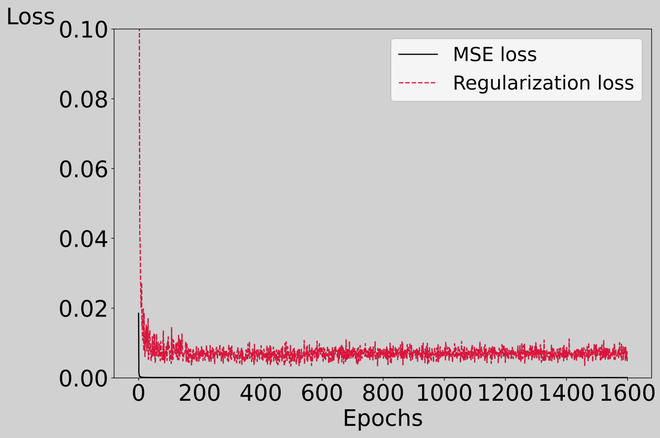

Point by point evaluation



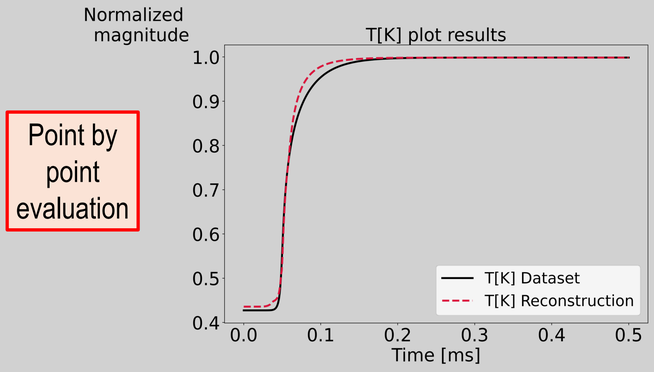

After a couple of steps in autoregression, the model blows-up
$\gamma = 0.2$

$R^2 = 0.996$

$\mathcal{L} = MSE + L_2 + L_{ct} + L_M$




Point by point evaluation




Autoregression arrives to final steady state, however, the trajectory presents some errors.
$\lambda = 1x10^{-4}$

$R^2 = 0.966$

## Conclusions

- L2 regularization presents the better improvement in terms of blow-up avoidance, however, the dynamics are not learnt accurately. When evaluated point by point, there is a decrease in accuracy.
- Implicit gradient alone helps reducing the error in the loss function, but it does not translate in any robustness increase.
- The implicit gradient helps the L2 regularization reactiveness, and the final steady-state is more accurate, however, the trajectory is not accurate enough.
- New options must be tested, as well as network architectures, as well as different regularization constants.

**References**

[1] M. Ihme, W. T. Chung, and A. A. Mishra, Combustion machine learning: Principles, progress and prospects, Progress in Energy and Combustion Science 91, 101010 (2022), ISSN 0360-1285.

[2] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, Regularisation of neural networks by enforcing lipschitz continuity, Machine Learning 110, 393 (2021), ISSN 1573-0565.

[3] P. Saxena and F. A. Williams, Testing a small detailed chemical-kinetic mechanism for the combustion of hydrogen and carbon monoxide, Combustion and Flame 145, 316 (2006), ISSN 0010-2180.

[4] P. Pauli, A. Koch, J. Berberich, P. Kohler, and F. Allg¨ower, Training robust neural networks using lipschitz bounds, IEEE Control Systems Letters 6, 121 (2022).

[5]. D. M. Hawkins, Introduction (Springer Netherlands, Dorcrecht, 1980), 1-12, ISSABN 978-94-015-3994-4,

**Acknowledgements**

This research is funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-F.N.R.S) project METRIC, under grant No.T.0175.21. Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region. Also, this work has received funding from the European Union's Horizon 2020 research and innovation programme under Marie Skłodowska-Curie grant agreement No 801505.