[1] University of Reading
[2] NCEO, United Kingdom
[3] University of of Utrecht, The Netherlands
[4] Colorado State University, Colorado.

University of Reading

Toward an adaptive numerical integration scheme dependent on ML-predicted accuracy

Daniel Ayers[1,2] | Javier Amezcua[1,2] | Varun Ojha[1] | Alberto Carrassi[1,2,3] | Peter Jan Van Leeuwen[1,2,4]

Motivation

The goal of this work is to explore the potential for an ML-enabled adaptive numerical integration scheme that **adapts as the simulation is running in response to what is occurring in the model**. The hope is that an adaptive numerical scheme will use fewer computational resources, thus freeing up resources that can be reinvested in, for example, more ensemble members. This goal of using resources more efficiently has motivated work into reduced precision computing, and inexact computing.[1]

Justification via a non-ML adaptive method

Fig. 1 offers a basic illustration of an adaptive numerical integration scheme. The plot shows trajectories of the Rössler system (an ODE system of 3 variables) calculated using Runge-Kutta fourth order (RK4), Runge-Kutta second order (RK2) and a simple adaptive scheme which relies on a manually-set threshold. The threshold method is defined as follows:

$$\mathbf{x}_{n+1} = \begin{cases} \mathrm{RK4}(\mathbf{x}_n) & g(\mathbf{x}_n) > \hat{g} \\ \mathrm{RK2}(\mathbf{x}_n) & g(\mathbf{x}_n) \leq \hat{g} \end{cases}$$

where $g\colon \mathbb{R}^N \to \mathbb{R}$, and $\hat{g} \in \mathbb{R}$ is a constant. In Fig. 1, $N = 3$, $\hat{g} = 4$.

Improved accuracy for a small cost

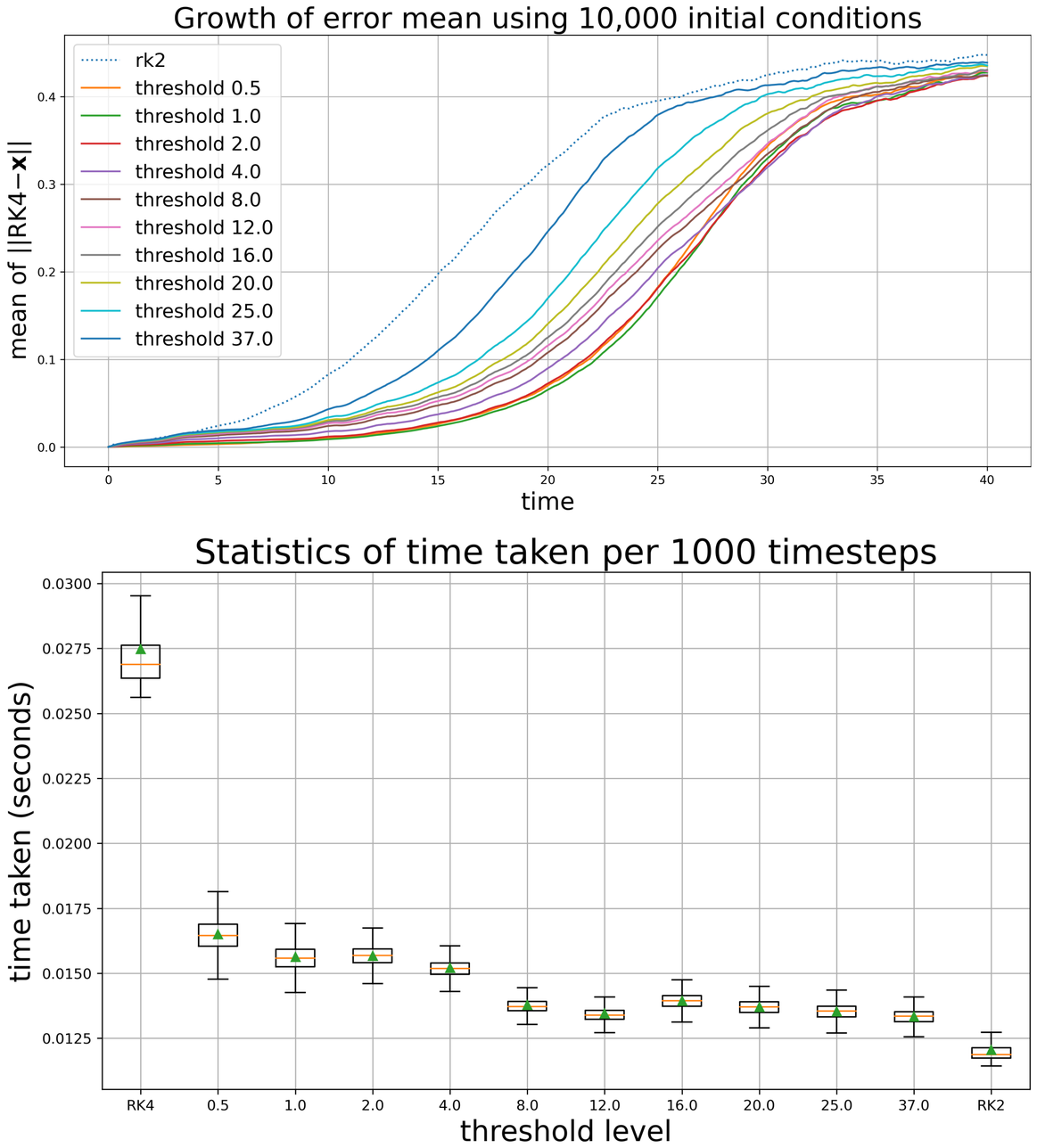

Figure 2: Error growth and timing statistics of threshold methods for 10,000 trajectories.

The threshold method capitalises on the fact that **local instability is not uniform.** The Rössler system is sufficiently simple that one can spot by visual inspection that there is a strong correlation between the value of z and the instability. The threshold system uses RK2 by default, and then *adapts* by switching to RK4 precisely when z is greater than the set threshold value. Fig. 1 shows a case in which the threshold method is clearly successful: treating the RK4 trajectory to be the truth, the threshold method remains "accurate" until around $t = 28$, compared to $t = 13$ for RK2. The threshold method uses RK2 for 86% of the total trajectory.

Illustration of threshold method

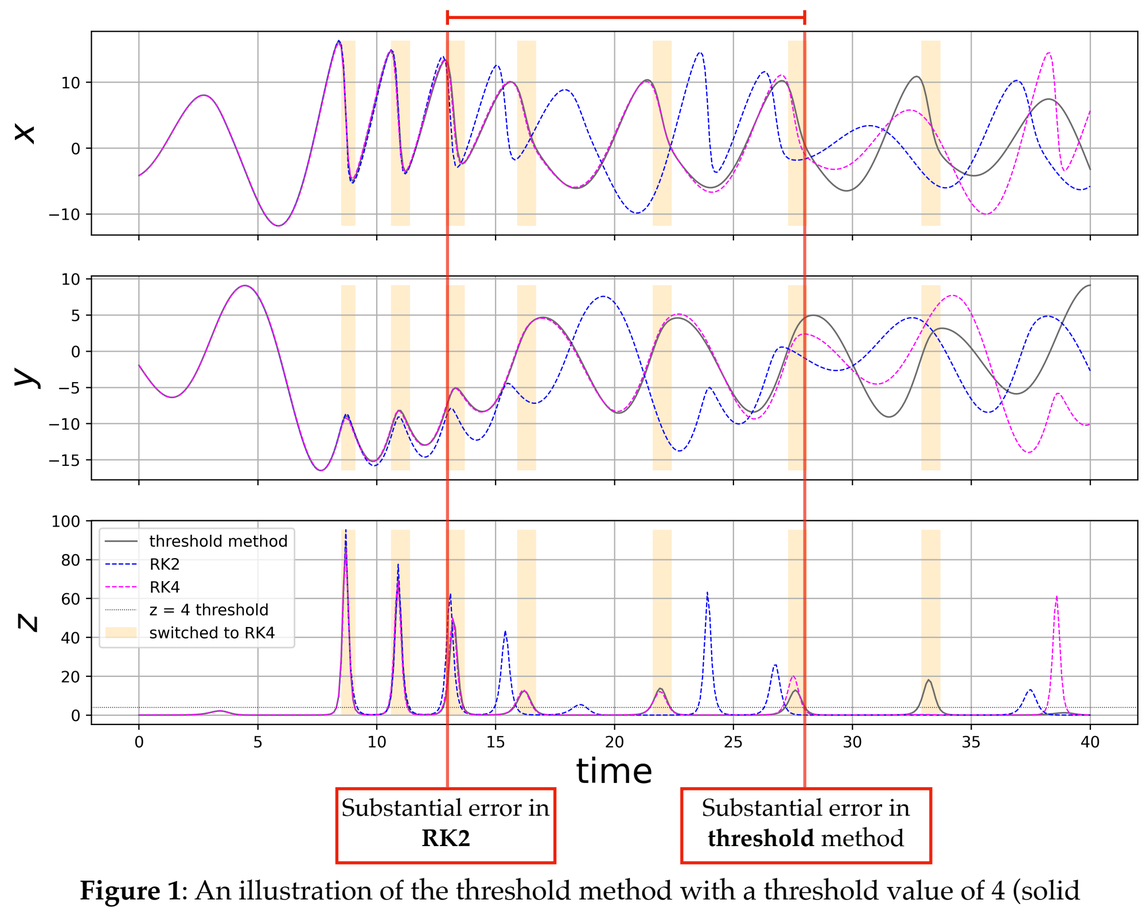

Figure 1: An illustration of the threshold method with a threshold value of 4 (solid grey line). This is shown alongside RK2 (blue dashed) and RK4 (pink dashed) solutions. The threshold method uses RK2 unless the current timestep has a z-value greater than 4. The red lines highlight the improved duration of accuracy of the threshold method over RK2.

The merit of such threshold schemes is demonstrated more robustly by the data in Fig. 2, which shows the growth of the mean error (where RK4 is taken to be the truth) and computation time per 1000-timestep trajectory for 10,000 initial conditions. Fig. 2 shows that the threshold with $\hat{z} = 4$ was on average 45% faster than RK4, and 25% slower than RK2, whilst having a mean error growth around 100 timesteps slower than that of RK2, which is significant given that the error of RK2 saturates at around 250 timesteps.

Machine Learning

ML is the solution to the problem of developing the threshold method (more generally, an adaptive approach) for more realistic, higher dimensional models. Our first step is to derive a ML-enabled rule to decide whether to use RK2 or RK4 - rather than picking the threshold rule ourselves. We plan to train a neural network to predict the deviation of RK2 from RK4 at the next timestep, and this prediction will facilitate a decision. Currently we are testing a *Long short-term memory* (LSTM) recurrent neural network (RNN), as LSTMs have been shown to be effective at learning from sequential data.[2] More work is required to determine how to generate a ML-enabled rule for other, potentially more complicated classes of adaptation, such as changing the spatial and temporal resolution.

References

1. Düben, Peter D., et al. "On the Use of Inexact, Pruned Hardware in Atmospheric Modelling." Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 372 (2018). https://doi.org/10.1098/rsta.2013.0276.
2. Graves, Alex. 2012. Supervised Sequence Labelling with Recurrent Neural Networks. Vol. 385. Studies in Computational Intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-24797-2.

Acknowledgements

- We would like to thank the Engineering and Physical Sciences Research Council for funding this research.
- We thank the members of Data Assimilation Research Centre (DARC) at the University of Reading for useful discussions.

Contact information

- Email: d.ayers@pgr.reading.ac.uk
- www.reading.ac.uk/met