Nonlinear Analysis and Differential Equations, Vol. 12, 2024, no. 1, 17 - 31
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/nade.2024.91253

Exploring the Performance of the Optimized *Inductive Linearization* Solver for Simulation of the Van der Pol Oscillator (a stiff non-linear system)

Sepideh Sharif

Otago Pharmacometrics Group, School of Pharmacy
University of Otago, Dunedin, New Zealand
Corresponding author

Chihiro Hasegawa

Otago Pharmacometrics Group, School of Pharmacy
University of Otago, Dunedin, New Zealand

Matthew Fidler

Novartis Pharmaceuticals
East Hanover, NJ, USA

Stephen B. Duffull

Otago Pharmacometrics Group, School of Pharmacy
University of Otago, Dunedin, New Zealand
Certara, Princeton, NJ, USA

Abstract

Ordinary differential equations (ODEs) define the relationship of $x(t)$ values with time (t). While exact solutions for linear differential equations can be solved with (for example) matrix exponentials, non-linear systems generally do not have a closed form solution. Here, we explore a novel method (*Inductive Linearization*) that has proven to be fast and efficient for non-stiff systems and explore how it performs when solving non-linear stiff systems. The research team has developed and refined the *Inductive Linearization,* as an innovative numerical solver. In this instance, the IL solver has been applied to the Van der Pol system which represents a simple and tuneable stiff system. This method approximates solutions to non-linear systems utilizing iterative linearization to create a linear time-varying (LTV) system of ODEs that spans the entire time interval of interest. The resultant LTV is then solved using *eigenvalue decomposition* (EVD). This study used *ode45* and *ode23s* method as the non-stiff and stiff reference solvers to examine the performance of the IL solver. Our findings reveal that the optimized inductive solver greatly improves stiff system solutions more than non-stiff systems, and further benefits from repeating oscillations which advances the understanding of the solver.

Keywords: Non-linear ordinary differential equations, Numerical methods, Optimization, Inductive Linearization, Adaptive step-size algorithm

Introduction

The *Inductive Linearization* (IL) method is a numerical technique that has been used to iteratively solve non-linear ordinary differential equations (ODEs) [1]. IL is a numerical method made up of two components: 1) linearization of the non-linear ODEs to form linear time-varying (LTV) ODEs, 2) integration of the LTV ODEs using eigenvalue decomposition (EVD) to solve matrix exponentials [2-4]. In the first component the linearization is applied to the whole time span of interest rather than in a time-stepping approach. The IL solver has been utilized for several systems including a general method to simplify a non-linear quantitative systems pharmacology model for a stiff system in bone biology [5] and an HIV viral model [2]. Non-linear stiff systems are common and important in pharmacology models such as delay models [6], system models [7] and viral dynamic models. However, evaluation of the performance of solvers, including IL, in application settings is difficult as the stiffness of the underlying ODEs is variable and largely unknown.

The Van der Pol equation is a single parameter dynamic non-linear 2nd order differential equation established by Dutch physicist Van der Pol [6]. It is used in many fields of biology [8] and engineering [9]. The oscillator dynamic is considered a good example of a non-linear stiff system which has no algebraic solution [10].
In this setting the Van der Pol system provides a simple tuneable reference case for exploring solver performance for both non-stiff and stiff systems. Given the absence of a closed-form algebraic solution, solving the Van der Pol system requires numerical integration. Algebraic integration yields a precise and computationally efficient closed-form solution. However, it is important to note that the majority of non-linear systems lack algebraic solutions.
The aim of this study is to explore the performance of IL when applied to a reference ODE system (the Van Der Pol oscillator) which can be tuned to both stiff and non-stiff settings.

Methods

All simulations were performed using MATLAB (R2024a) and the reference ODE solvers were the inbuilt numerical ODE solvers, *ode45* (for non-stiff systems) and *ode23s* (for stiff systems).
Inductive linearization is a numerical method introduced to solve non-linear ODEs [1]. It has similarities to Picard's method [11]. IL is a type of linear multistep method [12] and is made up of two components which when combined are termed the IL solver. The components are:

1) *Inductive linearization*– which linearizes the non-linear ODEs to form LTV ODEs. It has its own primary property, i.e., the number of iterations (n) which is defined to determine convergence of the method. The previous work has identified the benefits of optimizing the number of iterations based on convergence of the algorithm for a non-stiff system (denoted in this work as *N*-optimized) [13].
2) The LTV integrator– in this study, we used EVD. Note, EVD is exact for linear time-invariant systems. Here we divide the LTV into a series of linear piecewise elements and apply EVD to each element. The greater the number of elements (smaller the step-size [*ss*]) the greater the accuracy but at the cost of increased run-time. The previous work [13] has identified the benefits of optimizing the step-size of the EVD solver for a non-stiff system (denoted in this work as *ss*-optimized).

Inductive linearization

This method is described briefly here; further details are available in [1, 13]. Consider a simple general form of a non-linear ODE:

$$\frac{dy}{dt} = f - A(t, y) \cdot y; \tag{1}$$

where the homogenous term $\boldsymbol{A}(\cdot)$ is dependent on $\boldsymbol{y}$ and therefore the system is non-linear. The IL operates by removing the dependency of the homogeneous term on the unknown value of $\boldsymbol{y}$. (Note, application the IL is not limited to the homogeneous term). Here we redefine the function $\boldsymbol{A}$ so that it is dependent on a previous iteration of $\boldsymbol{y}$ such that the system is no longer self-referential as:

$$\frac{dy^{[n]}}{dt} = f(t) - A\left(t, y^{[n-1]}\right) \cdot y^{[n]}; \tag{2}$$

with an initial value of $\boldsymbol{y}^{[n=1]} = 0$ for all t. As $n \to \infty$ the LTV solution approaches the non-linear solution and the difference of successive iterations will approach 0. The maximum number of iterations N (i.e., a stopping rule) can therefore be optimized based on the difference of successive iterations, such that

$$\max_{t \in T} \left| \left(y(t)^{[n-1]} - y(t)^{[n]} \right) \right| < \varepsilon \tag{3}$$

where T is the span of time of interest and ε is some level of tolerance, in this case 1e-6. In the previous work the number of iterations is often less than 10 [13]. In this work we consider both fixed N and optimized N based on the stopping rule. Note the value of N transitions from a fixed, non-optimized state to an optimized condition where N is controlled by the IL solver.

Eigenvalue Decomposition

Eigenvalue decomposition is used to solve the LTV system of ODEs (produced by *Inductive linearization*). This method is based on factorization and decomposition of a rate constant matrix $\boldsymbol{K}$. Each matrix decomposition is built on similarity transformations of the form:

$K = \bar{V} \Lambda \bar{V}^{-1}$. The idea being to find $\bar{\boldsymbol{V}}$, the eigenvectors, and Λ, the eigenvalues, of $\boldsymbol{K}$, which then allow computation of $e^{t\Lambda}$ (see [14] for details of applying this method in this setting).

As per [13], the efficiency of EVD for solving an LTV is defined by the number and the size of each piecewise linear element. This is resolved by considering a step-size (ss) to cut the LTV into segments of various lengths. The units of ss are the same as the units of the independent variable, in this case time. The units of time here are arbitrary with no effect on the system. In this work we consider both fixed step-size (for example $ss = 0.01$) (as per [2]) and adaptive step-size employing various values of α (as per [13]). The latter determined as:

$$ss = \alpha \left| \left(\frac{dy^*}{dt} \right)^{-1} \right| \tag{4}$$

<u>The Van der Pol equation</u>

The Van der Pol equation is given by:

$$\frac{d^2y}{dt^2} - \mu(1 - y^2)\frac{dy}{dt} + y = 0 \tag{5}$$

$\frac{dx}{dt}$ is the first derivative and $\frac{d^2x}{dt^2}$ is the second derivative, and μ is the damping parameter.

By reforming Equation 5, we can obtain the following form for typical ODEs:

$$\frac{d^2y}{dt^2} = \mu(1 - y^2)\frac{dy}{dt} - y; \tag{6}$$

The function can be rewritten as a system of two first-order ODEs (Equation 7), with its initial conditions:

$$\frac{dy}{dt} = \begin{pmatrix} \frac{dy_1}{dt} \\ \frac{dy_2}{dt} \end{pmatrix} = \begin{bmatrix} y_2 \\ -y_1 + \mu(1 - y_1^2)y_2 \end{bmatrix}; \quad y_1(0) = 1;\ y_2(0) = 0 \tag{7}$$

Linearizing the ODEs yields the ODEs as per equations 8 & 9 which than then be formulated as a $\boldsymbol{K}$ matrix (Equation 10):

$$\frac{dy_1}{dt} = y_2 \tag{8}$$

$$\frac{dy_2^{[n]}}{dt} = -y_1 + \mu\left(1 - \left(y_1^{[n-1]}\right)^2\right) y_2{}^{[n]} \tag{9}$$

$$K = \begin{bmatrix} 0 & 1 \\ -1 & \mu\left(1 - \left(y_1^{[n-1]}\right)^2\right) \end{bmatrix} \tag{10}$$

The reference simulation was based on Equation 6 and the IL expression on Equation 7. The Van der Pol system was examined for two values of the damping parameter $\mu = 1, 10$ denoting the non-stiff and stiff versions, respectively. The MATLAB built-in functions *ode23s* (stiff) and *ode45* (non-stiff) were the reference time-stepping ODE solvers. We arbitrarily set the time units as hours.

Exploring the Van der Pol equation using the *Inductive Linearization* solver

The following scenarios were considered:

- The reference IL settings
  - Non-stiff ($\mu = 1$): fixed number of iterations of the linearization ($N = 20$) and fixed step-size ($ss = 0.01$) for EVD (as per [2])
  - Stiff ($\mu = 10$): fixed number of iterations of the linearization $N = 60$ and fixed step-size ($ss = 0.001$) for EVD
- The optimized IL settings
  - optimizing stopping rule with $\varepsilon = 1e-6$ (*N*-optimized)
  - optimizing adaptive step-size within a limited range of possible values $\alpha = \{0.001, 0.01, 0.03\}$ (*ss*-optimized)

The timespan of interest for the oscillator was $T = [0{:}30]$ for optimization of settings. The two phases of optimization here were (1) *N*-optimized then (2) adding on adaptive step-size (*N*-*ss*-optimized).

Finally, we took advantage of the known periodicity of the oscillator by solving the first cycle of the Van der Pol system and carrying forward the values of $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$ from the first cycle to be the plug-in values of $\boldsymbol{y_0}$ for all future cycles and setting $N = 0$ for subsequent cycles. This yields a single-step linearization process for the subsequent cycles. For the non-stiff system, the timespan of the first cycle is 10 hours and that for the 2 cycles is therefore 20 hours. In contrast, for the stiff system the timespan of the first cycle is 20 hours and that for the 2 cycles is therefore 40 hours. For both systems, the first cycle was solved using a *N*-*ss*-optimized IL and the second cycle using a 1-step linearized IL (i.e., the second cycle was linearized in a single-step approach rather than iteratively).

Results of the application of the IL are described in terms of accuracy and run-time. Accuracy was evaluated both numerically as the relative difference of the reference of IL from the ODE solutions (*ode45* and *ode23s*). In addition, since the IL approach linearizes the whole time span of interest during each iteration and the accuracy of IL decreases with increasing time it is therefore easy to visualize divergence of IL from the reference time-stepping ODE solvers (*ode45* or *ode23s*) which provides a simple and pragmatic tool for evaluating the IL solver.

Results

The results for run-time are presented in Table 1., Fig. 1a and Fig. 1b which show the graphical outcomes of the best $N\text{-}ss$ combinations for non-stiff and stiff systems, respectively. For the non-stiff setting the *N*-optimized IL converged in 1.86 seconds, which was twice the speed of the reference (non-optimized) IL. For the stiff setting the N-optimized was 4 times faster. Both solutions were however much slower than MATLAB's inbuilt time-stepping ODE solvers (*ode45* and *od23s*). Applying the *N-ss*-optimized IL to the non-stiff problem yielded a further 3 fold improvement in speed ($\alpha = 0.001$) which tracked the reference solution closely (Fig. 1a) but was slower than the reference *ode45*. For the stiff setting the *N-ss*-optimized solution was 6 times faster than the non-optimized approach ($\alpha = 0.03$) but was 2-fold slower than *ode23s*. Again this optimized setting tracked *ode23s* closely.

A benefit of the *Inductive linearization* solver is its ability to take advantage of repetitive patterns in the system. Results for solving the system for the first period and using this as a 1-step linearization for the subsequent period are provided in Table 2, Fig. 2a and Fig. 2b. Using the model predictions of y_1 and y_2 from period 1 as the 1-step linearization for period 2 resulted in a 4.7- and 2.3- fold increase in speed for cycle 2 for the non-stiff and stiff Van der Pol settings. This speed advantage improved the overall speed of the IL-solver over both non-stiff and stiff settings which would continue to improve the more cycles were solved. For the stiff system this improvement was comparable to *ode23s*.

Finally, accuracy defined as relative error versus time is shown in Fig.3a &3b for y_1. specified that the relative error between the reference solver (*ode45*) and the optimized IL compared for non-stiff Van der Pol to ($\mu = 1, \alpha = 0.001, y_1$) increased over the time after the two cycle more than IL for the stiff system. Therefore, the optimized IL more accurately solved the stiff Van der Pol ($\mu = 10, \alpha = 0.03, y_1$).

Discussion

We have investigated the performance of the *Inductive linearization* solver for both non-stiff and stiff non-linear differential equations. Our findings indicate that the optimized IL is capable of solving the stiff system with comparable performance to the non-stiff system. It is slower for the stiff system which is also a feature of reference time-stepping solvers. The optimization settings within IL were different between the stiff and non-stiff system which indicates a need to efficiently recognize stiff numerical ODE systems.

As anticipated optimizing, the number of iterations of the linearization component, via a stopping rule for *N*, resulted in significant improvements in run-time without loss of accuracy. Essentially, *N* is a linear operator on the computation load as it defines the number of times with which the system is integrated. In the case we explored, the improvement in run-time was modest over the reference non-optimized IL because our initial guess of the fixed value of *N* was not very different from the optimized value (see the non-stiff example where our fixed value was *N*=20 and the optimized value for this problem was *N*=19).

In addition, our findings also supported the importance of optimizing the step-size of the EVD integrator. In the case of the stiff system, our findings identified that $\alpha = 0.03$ was preferable to $\alpha = 0.001$. Applying the adaptive step-size optimization step with the best value of α for each scenario improved the run-time for both non-stiff and stiff systems about 3- and 6-fold, respectively. In the current work, values of α were chosen from a small set of possible values (based on previous experience (see [13])). Further research is required to optimize the value of α which will depend on the properties of the underlying system.

We showed that based on the information provided in Fig. 1a and Fig. 3a, it becomes evident that the IL solver solves the system throughout the entire time span. As the time span shortens, the solver operates more rapidly. However, due to the propagation of errors over time, the deviation between the solver's results and a reference solution increases over time.

We observed that the optimized IL can take advantage of periodicity, where solutions of $\boldsymbol{y}(t)$ from the first cycle can be plugged in as $\boldsymbol{y_0}(t)$ for subsequent oscillatory cycles. In this setting $\boldsymbol{y_0}(t)$ was not updated further. This is a compelling feature of IL as reusing the solution from period 1 for multi-period problems reduces the time-domain of the iterative linearization with subsequent periods being evaluated as a single function call.

A further benefit of IL is its application in iterative optimization problems, such as a gradient search method for parameter estimation. In this scenario it is straightforward to evaluate the next step in the parameter optimization process with the final estimates of the last step. As the optimization algorithm proceeds then each subsequent iteration will provide a solution that is closer to the final iteration. Hence plugging in the values of $\boldsymbol{y}(t)$ in the previous set iteration as the starting point of next iterations, $\boldsymbol{y_0}(t)$, takes advantage of the convergent nature of the algorithm and has been shown to result in significant benefits in run-time [13].

Conclusion

In conclusion, our brief investigation supports the application of the *Inductive Linearization* solver for stiff systems of non-linear ordinary differential equations. Of interest, the IL method can also take advantage of known periodicities in oscillatory systems that considerably improve its efficiency .

Acknowledgements. Sepideh Sharif is supported by University of Otago postgraduate scholarship.

Author Contributions. All authors conceived of the research and conducted the literature review.

Competing Interests. The authors declare that they have no conflict of interest. This work was supported by a doctoral scholarship from the University of Otago.

Code availability. The codes generated during the current study are available in the GITHUB repository: https://github.com/sepisharif/Van-der-Pol-system/tree/main

Method	N	ss	**Run-time (s)**
$\mu = 1$			
ode45	-	-	0.38
Non-optimized IL (reference)	20	0.01	3.86
N-optimized IL	19	0.01	1.86
N-ss-optimized IL	19	$\alpha = 0.001$	0.572
$\mu = 10$		ss	
ode23s	-	-	2.47
Non-optimized IL (reference)	60	0.001	125
N-optimized IL	49	0.001	32.07
N-ss-optimized IL	49	$\alpha = 0.03$	5.23

Table 1. Run-time of the optimized *Inductive Linearization* solver

Method	ss	**Run-time (s)**
$\mu = 1$		
ode45	-	0.20
The N-ss-optimized IL for cycle 1	$\alpha = 0.001$	0.240
The 1-step linearized IL for cycle 2	$\alpha = 0.001$	0.051
Total over both cycles		0.291
$\mu = 10$		
ode23s	-	2.80
The N-ss-optimized IL for cycle 1	$\alpha = 0.03$	1.44
The 1-step linearized IL for cycle 2	$\alpha = 0.03$	0.631
Total over both cycles		2.071

Table 2. Run-time of the one cycle solver predicting into subsequent cycles

References

[1] S.B. Duffull, G. Hegarty, An inductive approximation to the solution of systems of nonlinear ordinary differential equations in pharmacokinetics-pharmacodynamics, *Theor Comput Sci.*, **1** (2014), 1000119. https://doi.org/10.4172/jtco.1000119

[2] C. Hasegawa, S.B. Duffull, Exploring inductive linearization for pharmacokinetic–pharmacodynamic systems of nonlinear ordinary differential equations, *J. Pharmacokinet Pharmacodyn*, **45** (2018), 35-47. https://doi.org/10.1007/s10928-017-9527-z

[3] P. Jordan, R. Gieschke, Explicit solutions for a class of indirect pharmacodynamic response models, *Comput. Methods Programs Biomed.*, **77** (2005), 91-97. https://doi.org/10.1016/j.cmpb.2004.02.002

[4] G. Hegarty, S.B. Duffull, Solving systems of ordinary differential equations that arise in PKPD: first order (time-varying) linear systems [Internet]. School of Pharmacy, University of Otago; (2012). https://ndhadeliver.natlib.govt.nz/delivery/DeliveryManagerServlet?dps_pid=IE12244507

[5] C. Hasegawa , S.B. Duffull, Automated scale reduction of nonlinear QSP models with an illustrative application to a bone biology system, *CPT Pharmacometrics Syst Pharmacol*, **7** (2018), 562-572. https://doi.org/10.1002/psp4.12324

[6] F. M. Atay, Van der Pol's oscillator under delayed feedback, *J Sound Vib.* **218** (1998), 333-339. https://doi.org/10.1006/jsvi.1998.1843

[7] M. Peterson, M. Riggs, Predicting nonlinear changes in bone mineral density over time using a multiscale systems pharmacology model, *CPT Pharmacometrics Syst Pharmacol*, **1** (2012), no. 11, 1-8. https://doi.org/10.1038/psp.2012.15

[8] N.H. Abdul Aziz, Z. Abdul Majid, F. Ismail, Solving delay differential equations of small and vanishing lag using multistep block method, *J. Appl. Math.*, **2014** (2014), 1-10. https://doi.org/10.1155/2014/348912

[9] V.P. Zhuravlev, Van der Pol oscillator. Technical Applications, *Mechanics of Solids*, **55** (2020), 132-137. https://doi.org/10.3103/s0025654420010203

[10] W.H. Enright, T. Hull, B. Lindberg, Comparing numerical methods for stiff systems of ODEs, *BIT Numerical Mathematics*, **15** (1975), 10-48. https://doi.org/10.1007/bf01932994

[11] L. Shampine, Monotone iterations and two-sided convergence, *SIAM J Numer Anal.*, **3** (1966), 607-615. https://doi.org/10.1137/0703053

[12] D.F. Griffiths, D.J. Higham, Numerical methods for ordinary differential equations: initial value problems. London (20101)

[13] S. Sharif, C. Hasegawa, S.B. Duffull, Exploring Inductive Linearization for simulation and estimation with an application to the Michaelis–Menten model, *J Pharmacokinet Pharmacodyn*, **49** (2022), 1-9.
https://doi.org/10.1007/s10928-022-09813-z

[14] C. Moler, C. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, *SIAM J Math Data Sci.*, **45** (2003), 3-49.
https://doi.org/10.1137/s00361445024180

Figure legends

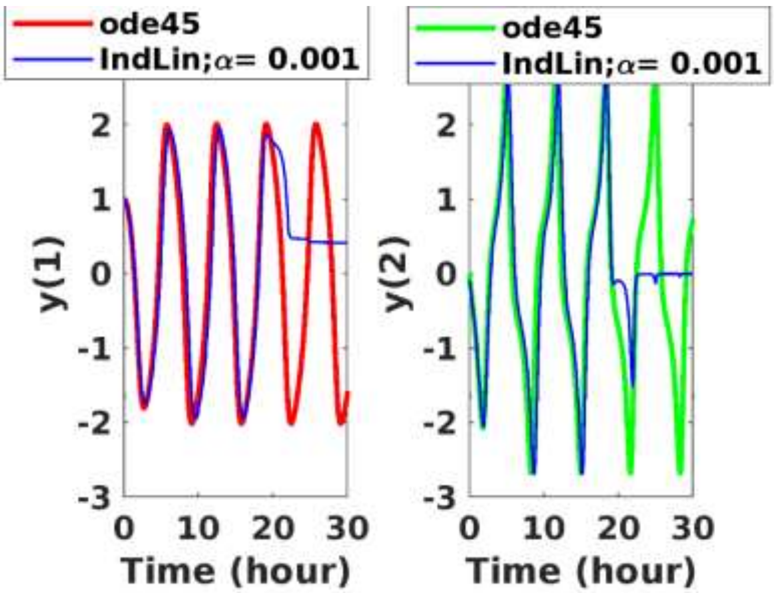

Fig. 1a The solutions for both dependent variables $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$ for the Van der Pol equation solved by the optimized *Inductive Linearization* solver ($\mu = 1$, $\alpha = 0.001$). Red (or green) and blue lines represent the *ode45* and *Inductive Linearization* solver solution, respectively.

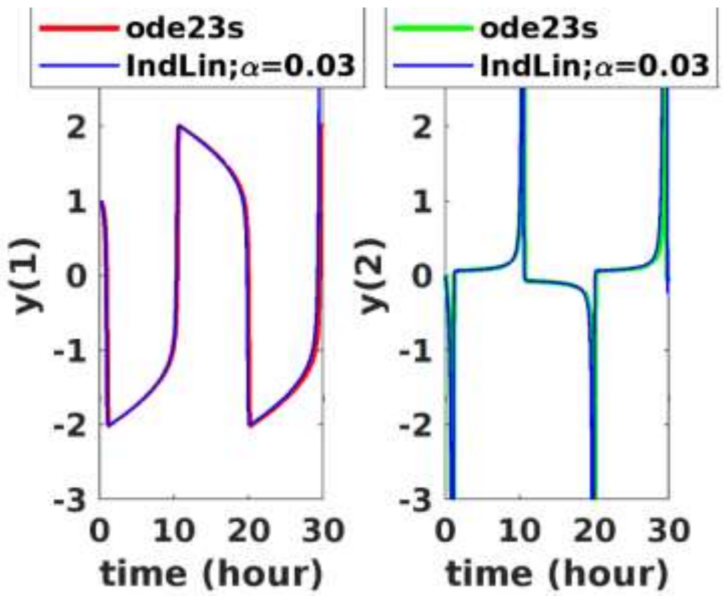

Fig. 1b The solutions for both dependent variables y_1 and y_2 for the Van der Pol equation solved by the optimized *Inductive Linearization* solver ($\mu = 10$, $\alpha = 0.03$). Red (or green) and blue lines represent the *ode23s* and *Inductive Linearization* solver solution, respectively.

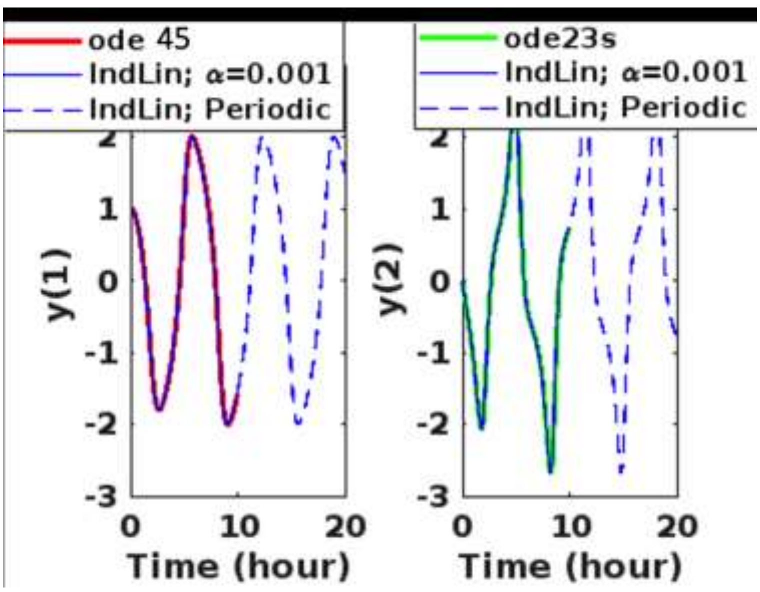

Fig. 2a The solutions for periodicity optimization for the non-stiff system while it can solve the second cycle based on the first solution. Red (or green) and blue lines represent the *ode45* and *Inductive Linearization* solver solution, respectively.

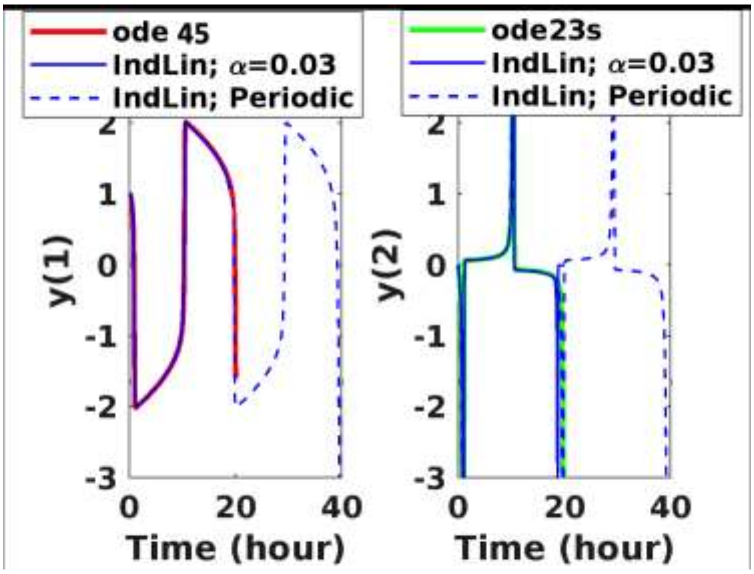

Fig. 2b The solutions for periodicity optimization for the stiff system while it can solve the second cycle based on the first solution. Red (or green) and blue lines represent the *ode23s* and *Inductive Linearization* solver solution, respectively.

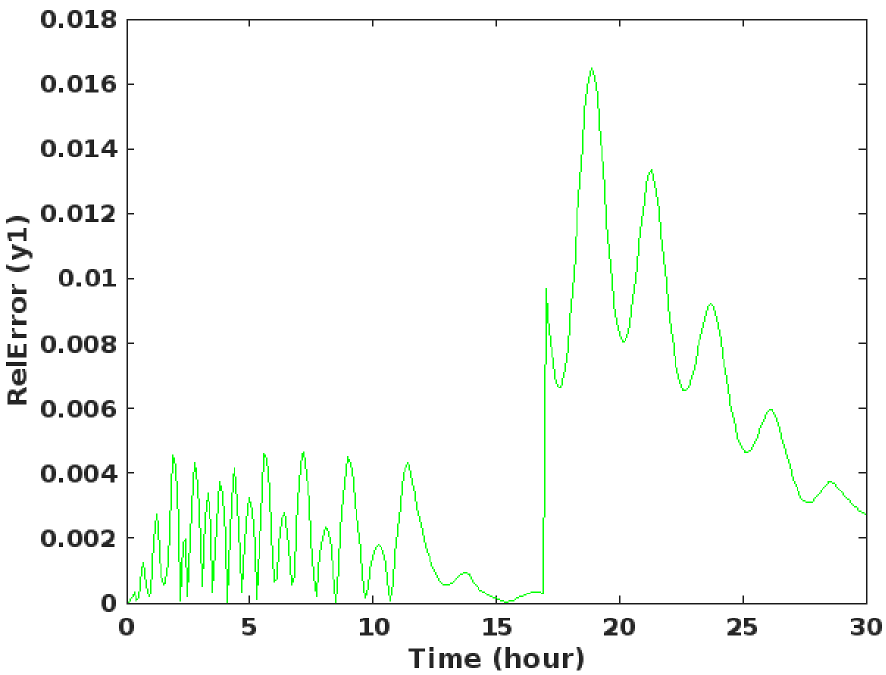

Fig. 3a The relative error (RelError) vs. Time (hour) for the optimized IL solver (μ = 1, α = 0.001, y_1) and *ode45* solver.

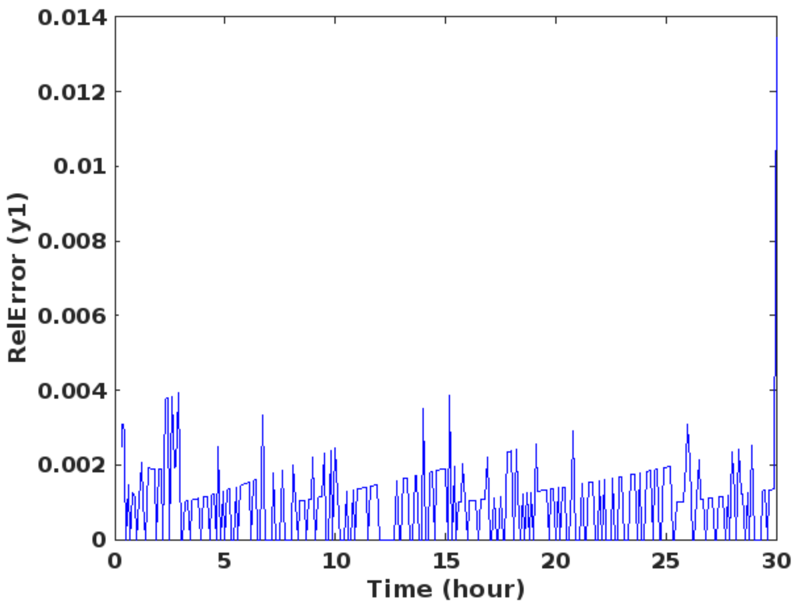

Fig. 3b The relative error (RelError) vs. Time (hour) for the optimized IL solver (μ = 10, α = 0.03, y_1) and *ode23s* solver.

Received: May 27, 2024; Published: June 7, 2024