Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Efficient exponential methods for genetic regulatory systems[☆]

Vu Thai Luan [a,*], Nguyen Van Hoang [a], Julius O. Ehigie [b]

[a] *Department of Mathematics and Statistics, Mississippi State University, 410 Allen Hall, 175 President's Circle, Mississippi State, MS, 39762, USA*
[b] *Department of Mathematics, University of Lagos, 23401, Nigeria*

ARTICLE INFO

Article history:
Received 1 October 2022
Received in revised form 17 May 2023

Keywords:
Genetic regulatory systems
Exponential methods
Convergence analysis
Linear stability analysis

ABSTRACT

In this paper, we construct and analyze a new second-order exponential method for simulating genetic regulatory systems. For such systems involving highly nonlinearities, a novel splitting technique is proposed to improve the stability and thus further increase accuracy and efficiency of the method. This idea can be also used with high-order exponential Runge–Kutta methods. Another possibility is to use exponential Rosenbrock methods which are derived based on dynamic linearization of the system. In particular, we additionally suggest a second- and fourth-order exponential Rosenbrock methods for genetic regulatory systems depending on their structure and properties. The linear stability analysis of the proposed methods is also presented. Our numerical experiments on a 2-gene model, 3-gene repressilator and a 13-gene budding yeast cell cycle model show the effectiveness of the proposed methods and technique over several widely used methods for genetic regulatory systems in the literature.

1. Introduction

Gene expression is the process by which the information from a gene is used to produce proteins from RNA molecules and non-coding RNA. To explain the interactions betweens genes and visualize the chronological activities of gene expression, genetic regulatory networks are designed. In the past, experiments were performed to analyze and understand the mechanism. For example, the development of proteomic methods based on two-dimensional gel electrophoresis, mass spectrometry and double-hybrid systems allows the identification of proteins and their interaction on a genomic scale [1]. However, characteristics such as the adaptation to its environment, cell cycle differentiation and its evolution on a very long time scale cannot be easily analyzed due to the fact that molecular interactions are quite heterogeneous in nature. Given a large number of components of most networks of biological interactions connected by negative and positive feedback loops, an intuitive comprehension of the dynamics of the systems is often difficult, hence mathematical and computational tools are employed to simulate, understand and analyze the system under chosen experimental conditions, see, e.g., [2,3].

Ordinary differential equations (ODEs) have been widely used for modeling complex interactions in a genetic regulatory system. They can be used to describe the rates of change in the concentration of mRNAs and proteins and to understand the dynamic behavior exhibited by interacting genes in a biochemical network [4–7]. Specific applications can be found in, e.g., [8–13]. In most cases the closed form solution of these nonlinear models is not possible, hence numerical methods

[☆] V.T. Luan and N.V. Hoang have been supported in part by the National Science Foundation (NSF) under awards DMS–2012022 and DMS–2309821. J.O. Ehigie was supported by a grant under the IMU-Simons African Fellowship.

[*] Corresponding author.
E-mail addresses: luan@math.msstate.edu (V.T. Luan), vn208@msstate.edu (N.V. Hoang), jehigie@unilag.edu.ng (J.O. Ehigie).

https://doi.org/10.1016/j.cam.2023.115424
0377-0427/© 2023 Elsevier B.V. All rights reserved.

are used to generate approximate solutions. For the simulation of genetic regulatory systems, classical (explicit) Runge–Kutta (RK) methods are often used, such as the fourth-order RK scheme. However, many of these systems are stiff, highly nonlinear with oscillatory solutions (e.g., the repressilator models). In this case, classical RK methods usually fail to capture the right dynamic behavior of the system. This is due to their poor stability properties and the fact that these methods do not take into account the structure of the system.

Recently, a splitting method was proposed in [14] which entails the composition of the exact solution of the linear part comprising of the degradation rates of the mRNAs and protein together with an explicit treatment of the nonlinear part. This technique has been shown to be more efficient than the classical RK methods, but offers low level of accuracy. In [15] exponentially fitted two-derivative RK (TDRK) methods were proposed to better treat the oscillation components of the genetic regulatory systems. However, only classical order conditions were derived, which means that the accuracy still highly depends on frequencies of the system. Very recently, integrating factor Runge–Kutta methods (IFRK) [16] were applied as a steady-state preserving algorithm to genetic regulatory systems [17]. These methods can take advantage of the structure of the system to exactly capture (using exponential treatment) the linear components involving the degradation rates. For systems with strong/highly nonlinearities, these methods still suffer from order reduction, thereby reducing the level of accuracy.

Motivated by the observations above, in this work we propose a new second-order exponential method called `TDEXP2` for the simulation of genetic regulatory systems. This method not only takes advantage of the structure of the system but also can handle the case of strong nonlinearity through a novel partitioning technique. Furthermore, using this we explore the applications of recently advanced exponential Runge–Kutta/Rosenbrock (ExpRK/ExpRB) methods to these genetic regulatory systems. In particular, we suggest the use of the several efficient methods of orders 2 and 4 among these advanced time integration methods.

The outline of the paper is as follows. In Section 2, we introduce the genetic regulatory models. Section 3 devotes to the construction and analysis of the new method `TDEXP2` and to suggest a treatment of highly nonlinear term based on a partition technique for the efficient implementation of the new scheme and exponential RK methods. The linear stability analysis of the proposed methods is presented in Section 4. In Section 5 we carry out our numerical experiments on a 2-gene cross-regulation model, a 3-gene repressilator model, and a 13-gene budding yeast cell cycle model in order to show the efficiency and accuracy of the proposed methods. Finally, our concluding remarks and future works are given in Section 6.

2. Genetic regulatory models

Genetic regulatory systems have been modeled by using the first principles and common rules of Michaelis–Menten enzymatic and reaction kinetics (see, e.g., [18]), resulting in differential equations. These models involve a key component that is highly nonlinear, so-called the Hill function [19] (describing the transcription phase). In this work, we consider deterministic models for a typical transcription–translation process. These are ODEs that describe the rate of change of the concentrations of mRNAs, proteins, and other molecule elements of the system by time-dependent variables. The structure of the feedback effects on transcription, splicing, and translation processes for genetic regulatory networks has been extensively discussed in, e.g., [20–22].

Assuming that there are N inputs of proteins, say, $p_1(t), p_2(t), \dots, p_N(t)$, which come from its own or other nodes, respectively. The regulations and interactions of the inputs on the gene or mRNA i ($i = 1, \dots, N$) are represented as an increasing (decreasing) nonlinear function to depict activation (repression) effect of the individual proteins, respectively. But there is only one output, that is, the protein $p_i(t)$, from any single node i, which may activate or repress its own gene. The mRNA i and protein i degrade with constant rates $\gamma_i > 0$ and $\mu_i > 0$, respectively. The coupling and interaction of many nodes yield a genetic regulatory network (GRN). This GRN can also be modeled with the reaction rate equations like reactions of chemical compounds, expressing the rate of the production of a gene product (protein) or an mRNA (see, for instance, [2,6,15]). In the light of this, the system of ODEs modeling for a network with N genes is given by

$$\begin{aligned} &m_i'(t) = -\gamma_i m_i(t) + R_i(p_1(t), p_2(t), \dots, p_N(t)) \\ &p_i'(t) = -\mu_i p_i(t) + T_i(m_i(t)), \quad i = 1, \dots, N \end{aligned} \tag{2.1}$$

where $m_i(t),\ p_i(t) \in \mathbb{R}$ represent the concentration of mRNAs and proteins at time t (with degradation rates γ_i and μ_i), respectively. $R_i(p_1(t), \dots, p_N(t))$ and $T_i(m_i)$ are regulatory functions (generally of Michaelis–Menten type which shows switch-like phenomenon denoting the degree of cooperativity) and translation functions, respectively. Usually, mRNAs are translated to copies of proteins linearly at a constant rate κ, i.e., $T_i(m_i(t)) = \kappa m_i(t)$.

3. New proposed methods for simulating genetic regulatory models

In this section, we derive a new second-order exponential scheme for numerical solving the N-gene regulatory system (2.1). In addition to this, we also propose an efficient procedure to apply ExpRK and ExpRB methods to the regulatory system (2.1), particularly when the nonlinearity is strong.

3.1. Matrix form of the N-gene regulatory system

Note that (2.1) can be written in the following partitioned form

$$\boldsymbol{m}'(t) = \Gamma \boldsymbol{m}(t) + R(\boldsymbol{p}(t)) = f(\boldsymbol{m}(t), \boldsymbol{p}(t)), \tag{3.1a}$$

$$\boldsymbol{p}'(t) = M\boldsymbol{p}(t) + T(\boldsymbol{m}(t)) = g(\boldsymbol{m}(t), \boldsymbol{p}(t)), \tag{3.1b}$$

where we use the following vector and matrix notations

$$\boldsymbol{m}(t) = [m_1(t), \dots, m_N(t)]^{\mathrm{T}}, \quad \boldsymbol{p}(t) = [p_1(t), \dots, p_N(t)]^{\mathrm{T}}, \tag{3.2a}$$

$$\Gamma = \mathrm{diag}[-\gamma_1, \dots, -\gamma_N], \quad M = \mathrm{diag}[-\mu_1, \dots, -\mu_N] \tag{3.2b}$$

and

$$R(\boldsymbol{p}(t)) = [R_1(\boldsymbol{p}(t)), \dots, R_N(\boldsymbol{p}(t))]^{\mathrm{T}}, \ T(\boldsymbol{m}(t)) = [T_1(\boldsymbol{m}(t)), \dots, T_N(\boldsymbol{m}(t))]^{\mathrm{T}}.$$

Equivalently, one can also denote $\boldsymbol{y}(t) = [\boldsymbol{m}(t), \boldsymbol{p}(t)]^{\mathrm{T}}$ as the solution vector to represent (3.1) as

$$\boldsymbol{y}'(t) = L\boldsymbol{y}(t) + \mathcal{N}(\boldsymbol{y}(t)) =: F(\boldsymbol{y}(t)), \tag{3.3}$$

where

$$L = \begin{bmatrix} \Gamma & \mathbf{0} \\ \mathbf{0} & M \end{bmatrix}, \quad \mathcal{N}(\boldsymbol{y}) = \begin{bmatrix} R(\boldsymbol{p}) \\ T(\boldsymbol{m}) \end{bmatrix}, \quad F(\boldsymbol{y}) = \begin{bmatrix} f(\boldsymbol{m}, \boldsymbol{p}) \\ g(\boldsymbol{m}, \boldsymbol{p}) \end{bmatrix}. \tag{3.4}$$

3.2. Construction of a new second-order exponential method

Starting from initial conditions $\boldsymbol{m}(t_0)$ and $\boldsymbol{p}(t_0)$, we represent the solution of (3.1) at time t as

$$\boldsymbol{m}(t) = \mathrm{e}^{(t-t_0)\Gamma}\boldsymbol{m}(t_0) + \int_{t_0}^{t} \mathrm{e}^{(t-\tau)\Gamma} R(\boldsymbol{p}(\tau))\mathrm{d}\tau, \tag{3.5a}$$

$$\boldsymbol{p}(t) = \mathrm{e}^{(t-t_0)M}\boldsymbol{p}(t_0) + \int_{t_0}^{t} \mathrm{e}^{(t-\tau)M} T(\boldsymbol{m}(\tau))\mathrm{d}\tau \tag{3.5b}$$

by applying the variation-of-constants formula to (3.1a) and (3.1b), respectively. Next, we observe the following expansion of the solution of (3.1) based on (3.5).

Lemma 3.1. *Suppose that $\boldsymbol{m}(t)$ and $\boldsymbol{p}(t)$ are twice differentiable with bounded derivatives in $\mathbb{R}^N$ and that the regulatory and translation functions, $R(\boldsymbol{p})$, $T(\boldsymbol{m})$ are twice continuously differentiable in strip along the exact solution. Then, the solution of the N-gene regulatory system (3.1) at time $t_{n+1} = t_n + h$ (where $h > 0$ denotes the time step size) satisfies the following expansion*

$$\boldsymbol{m}(t_{n+1}) = \boldsymbol{m}(t_n) + h\varphi_1(h\Gamma)f(\boldsymbol{m}(t_n), \boldsymbol{p}(t_n)) + h^2\varphi_2(h\Gamma)R'(\boldsymbol{p}(t_n))g(\boldsymbol{m}(t_n), \boldsymbol{p}(t_n)) + \mathcal{O}(h^3), \tag{3.6a}$$

$$\boldsymbol{p}(t_{n+1}) = \boldsymbol{p}(t_n) + h\varphi_1(hM)g(\boldsymbol{m}(t_n), \boldsymbol{p}(t_n)) + h^2\varphi_2(hM)T'(\boldsymbol{m}(t_n))f(\boldsymbol{m}(t_n), \boldsymbol{p}(t_n)) + \mathcal{O}(h^3), \tag{3.6b}$$

where the functions $\varphi_1(z)$ and $\varphi_2(z)$ are given by

$$\varphi_1(z) = \int_0^1 \mathrm{e}^{(1-\theta)z}\mathrm{d}\theta = \frac{\mathrm{e}^z - 1}{z}, \quad \varphi_2(z) = \int_0^1 \mathrm{e}^{(1-\theta)z}\theta\,\mathrm{d}\theta = \frac{\mathrm{e}^z - 1 - z}{z^2}, \tag{3.7}$$

and the terms behind $\mathcal{O}(h^3)$ are bounded uniformly by Ch^3, where the constant C is independent of n and h.

Proof. Applying (3.5) for each time interval $[t_n, t_{n+1}]$ and changing the integration variable $\tau = t_n + \theta h$ leads to

$$\boldsymbol{m}(t_{n+1}) = \mathrm{e}^{h\Gamma}\boldsymbol{m}(t_n) + h\int_0^1 \mathrm{e}^{(1-\theta)h\Gamma} R(\boldsymbol{p}(t_n + \theta h))\mathrm{d}\theta, \tag{3.8a}$$

$$\boldsymbol{p}(t_{n+1}) = \mathrm{e}^{hM}\boldsymbol{p}(t_n) + h\int_0^1 \mathrm{e}^{(1-\theta)hM} T(\boldsymbol{m}(t_n + \theta h))\mathrm{d}\theta. \tag{3.8b}$$

Expanding $\boldsymbol{p}(t_n+\theta h)$ in a Taylor series at t_n gives $\boldsymbol{p}(t_n+\theta h)=\boldsymbol{p}(t_n)+\theta h\boldsymbol{v}$ with $\boldsymbol{v}=\boldsymbol{p}'(t_n)+\theta h\int_0^1 \boldsymbol{p}''(t_n+\xi\theta h)(1-\xi)\mathrm{d}\xi$. Using this, $R(\boldsymbol{p}(t_n+\theta h))$ can be expanded by its Taylor expansion at $\boldsymbol{p}(t_n)$ as

$$\begin{aligned} R(\boldsymbol{p}(t_n+\theta h)) &= R(\boldsymbol{p}(t_n))+\theta h R'(\boldsymbol{p}(t_n))\boldsymbol{p}'(t_n)\\ &\quad + (\theta h)^2 R'(\boldsymbol{p}(t_n))\int_0^1 \boldsymbol{p}''(t_n+\xi\theta h)(1-\xi)\mathrm{d}\xi\\ &\quad + (\theta h)^2\int_0^1 R''(\boldsymbol{p}(t_n)+\xi\theta h\boldsymbol{v})(\boldsymbol{v},\boldsymbol{v})(1-\xi)\mathrm{d}\xi. \end{aligned}$$

Inserting this into (3.8a) shows that

$$\boldsymbol{m}(t_{n+1})=\mathrm{e}^{h\Gamma}\boldsymbol{m}(t_n)+h\varphi_1(h\Gamma)R(\boldsymbol{p}(t_n))+h^2\varphi_2(h\Gamma)R'(\boldsymbol{p}(t_n))\boldsymbol{p}'(t_n)+\mathcal{O}(h^3), \tag{3.9}$$

in which the remainder term behind the Landau notation $\mathcal{O}(h^3)$ is given by

$$h^3\int_0^1 \mathrm{e}^{(1-\theta)h\Gamma}\int_0^1 \theta^2(1-\xi)\Big(R'(\boldsymbol{p}(t_n))\boldsymbol{p}''(t_n+\xi\theta h)+R''(\boldsymbol{p}(t_n)+\xi\theta h\boldsymbol{v})(\boldsymbol{v},\boldsymbol{v})\Big)\mathrm{d}\xi\,\mathrm{d}\theta$$

that is, under the regularity of the solution $\boldsymbol{p}(t)$ and $R(\boldsymbol{p})$ and the fact that $\|\mathrm{e}^{(1-\theta)h\Gamma}\|\le 1$), bounded (uniformly) by Ch^3 (with C is independent of n and h). The expansion (3.6a) now follows directly from (3.9) by using the equalities $\mathrm{e}^{h\Gamma}=I+h\varphi_1(h\Gamma)\Gamma$ (see $\varphi_1(z)$'s formula in (3.7)), $R(\boldsymbol{p}(t_n))=f(\boldsymbol{m}(t_n),\boldsymbol{p}(t_n))-\Gamma\boldsymbol{m}(t_n)$ (see (3.1a)), and (3.1b).

Similarly, one can expand $T(\boldsymbol{m}(t_n+\theta h))$ in a Taylor series and show that (3.8b) implies (3.6b). □

Now, using the result of Lemma 3.1 and assuming that $\boldsymbol{m}_n\approx\boldsymbol{m}(t_n)$, $\boldsymbol{p}_n\approx\boldsymbol{p}(t_n)$, we arrive at the following numerical scheme which will be called TDEXP2:

$$\boldsymbol{m}_{n+1}=\boldsymbol{m}_n+h\varphi_1(h\Gamma)f(\boldsymbol{m}_n,\boldsymbol{p}_n)+h^2\varphi_2(h\Gamma)R'(\boldsymbol{p}_n)g(\boldsymbol{m}_n,\boldsymbol{p}_n), \tag{3.10a}$$

$$\boldsymbol{p}_{n+1}=\boldsymbol{p}_n+h\varphi_1(hM)g(\boldsymbol{m}_n,\boldsymbol{p}_n)+h^2\varphi_2(hM)T'(\boldsymbol{m}_n)f(\boldsymbol{m}_n,\boldsymbol{p}_n). \tag{3.10b}$$

Note that, by using (3.3)–(3.4), this scheme can be also written as

$$\boldsymbol{y}_{n+1}=\boldsymbol{y}_n+h\varphi_1(hL)F(\boldsymbol{y}_n)+h^2\varphi_2(hL)\mathcal{N}'(\boldsymbol{y}_n)F(\boldsymbol{y}_n). \tag{3.11}$$

Remark 3.1. Since $\mathcal{N}'(\boldsymbol{y})F(\boldsymbol{y})=\big(F'(\boldsymbol{y})-L\big)F(\boldsymbol{y})=\boldsymbol{y}''(t)-LF(\boldsymbol{y})$ and $\varphi_1(z)=z\varphi_2(z)+1$ (see (3.7)), one can easily see that (3.11) is equivalent to

$$\boldsymbol{y}_{n+1}=\boldsymbol{y}_n+hF(\boldsymbol{y}_n)+h^2\varphi_2(hL)\boldsymbol{y}_n'', \tag{3.12}$$

where $\boldsymbol{y}_n''=F'(\boldsymbol{y}_n)F(\boldsymbol{y}_n)\approx\boldsymbol{y}''(t_n)$. This explains the abbreviation TDEXP2 (standing for "two-derivative exponential scheme of order 2") that we have used to name our new scheme (3.10).

In the following we show that TDEXP2 is convergent of order 2 when solving the N-gene regulatory system (3.1).

Theorem 3.1 (*Convergence of TDEXP2*). *Under the assumptions of Lemma 3.1, the numerical solution of the N-gene regulatory system (3.1) obtained by the scheme TDEXP2 (3.10) converges with order 2. In particular, the error bounds*

$$\|\boldsymbol{m}_n-\boldsymbol{m}(t_n)\|\le C_1h^2,\quad \|\boldsymbol{p}_n-\boldsymbol{p}(t_n)\|\le C_2h^2 \tag{3.13}$$

hold uniformly on $t_0\le t_n=t_0+nh\le t_{end}$, where the constants C_1 and C_2 depend on $t_{end}-t_0$, but are independent of n and h.

Proof. For the sake of convenience in our proof, we will use (3.11) as a compact form of TDEXP2. Let $e_{n+1}=\boldsymbol{y}_{n+1}-\boldsymbol{y}(t_{n+1})$ and LE_{n+1} denote its global and local error at time t_{n+1}, respectively. In view of (3.6), we see that

$$\boldsymbol{y}(t_{n+1})=\boldsymbol{y}(t_n)+h\varphi_1(hL)F(\boldsymbol{y}(t_n))+h^2\varphi_2(hL)\mathcal{N}'(\boldsymbol{y}(t_n))F(\boldsymbol{y}(t_n))+\mathcal{O}(h^3), \tag{3.14}$$

which implies $LE_{n+1}=\mathcal{O}(h^3)$, i.e., $\|LE_{n+1}\|\le Ch^3$ (with note that this constant C can be chosen uniformly and is independent of n and h, according to the proof of Lemma 3.1). Subtracting (3.14) from (3.11) and using (3.3) gives

$$e_{n+1}=e_n+h\varphi_1(hL)Le_n+hS_n+\mathcal{O}(h^3)=\mathrm{e}^{hL}e_n+hS_n+LE_{n+1}, \tag{3.15}$$

where

$$S_n=\varphi_1(hL)[\mathcal{N}(\boldsymbol{y}_n)-\mathcal{N}(\boldsymbol{y}(t_n))]+h\varphi_2(hL)[\mathcal{N}'(\boldsymbol{y}_n)F(\boldsymbol{y}_n)-\mathcal{N}'(\boldsymbol{y}(t_n))F(\boldsymbol{y}(t_n))]. \tag{3.16}$$

Solving the recurrence relation (3.15) and employing $e_0=\boldsymbol{y}_0-\boldsymbol{y}(t_0)=0$ leads to

$$e_n=h\sum_{j=0}^{n-1}\mathrm{e}^{(n-j)hL}S_j+\sum_{j=0}^{n-1}\mathrm{e}^{jhL}LE_{n-j}. \tag{3.17}$$

Due to the structure of $L = \mathrm{diag}[-\gamma_1, \dots, -\gamma_N, -\mu_1, \dots, -\mu_N]$ (see (3.4)), one can easily see that $\|\mathrm{e}^{(n-j)hL}\| \le 1$ and $\|\mathrm{e}^{jhL}\| \le 1$. Therefore, we get from (3.17) the bound

$$\|e_n\| \le h \sum_{j=0}^{n-1} \|S_j\| + \sum_{j=0}^{n-1} \|LE_{n-j}\|, \tag{3.18}$$

implying that the global error can be estimated by the sum of the propagated local errors. To verify the bounds in (3.13), it now remains to show the stability of the error propagation, which depends on the boundedness of $\|S_n\|$ given in (3.16).

Taking a closer look at (3.16) and using (3.3), one can verify that

$$S_n = \varphi_1(hL)\big(\mathcal{N}(\boldsymbol{y}_n) - \mathcal{N}(\boldsymbol{y}(t_n))\big) + h\varphi_2(hL)\big(\mathcal{N}'(\boldsymbol{y}_n) - \mathcal{N}'(\boldsymbol{y}(t_n))\big)\boldsymbol{y}'(t_n)$$
$$+ h\varphi_2(hL)\mathcal{N}'(\boldsymbol{y}_n)LE_n + h\varphi_2(hL)\mathcal{N}'(\boldsymbol{y}_n)\big(\mathcal{N}(\boldsymbol{y}_n) - \mathcal{N}(\boldsymbol{y}(t_n))\big).$$

Next we note that, under the assumptions of $R(\boldsymbol{p})$ and $T(\boldsymbol{m})$, it is easy to see that the functions $\mathcal{N}(\boldsymbol{y})$ and $\mathcal{N}'(\boldsymbol{y})$ are locally Lipschitz in a strip a long the exact solution. Together with the boundedness of $\varphi_1(hL)$ and $\varphi_2(hL)$, it is then straightforward that $\|S_n\| \le \widetilde{C}\|\boldsymbol{y}_n - \boldsymbol{y}(t_n)\| = \widetilde{C}\|e_n\|$, where $\widetilde{C}$ can be chosen independent of n and h.

Putting altogether, we derive from (3.18) that

$$\|e_n\| \le h \sum_{j=0}^{n-1} \widetilde{C}\|e_j\| + \sum_{j=0}^{n-1} Ch^3. \tag{3.19}$$

An application of a discrete Gronwall lemma then shows that $\|e_n\| \le \widehat{C}h^2$, which implies the bounds (3.13). □

Remark 3.2. Due to the construction of TDEXP2, one can see that it is an explicit one-stage scheme and can integrate the linear portion (involving the transcription and degradation rates) of the N-gene system (3.1) exactly. These are the advantages of TDEXP2 when simulating the genetic systems evolving a wide range of rates (concerning the stiffness). Also, while the matrix functions $\varphi_1(Z)$, $\varphi_2(Z)$ appear in this scheme, they can be computed directly (exactly) and efficiency because of the diagonal structure of Γ and M. Therefore, TDEXP2 could be implemented in an efficient way and we anticipate that it would be more competitive in terms of computational efficiency for larger systems when compared to other methods of the same order (see our numerical experiments in Section 5).

3.2.1. Improving stability and efficiency when integrating systems with strong nonlinearity

The proof of Theorem 3.1 gives us more insightful understanding about the stability of TDEXP2. In particular, when viewing the term S_n which controls how the errors are propagated through the integration process, we observe that if the N-gene regulatory system (3.1) has a strong (or stiff) nonlinearity (this is the case, for instance, with the 3-gene repressilator system), the error constant of TDEXP2 could be large. As a remedy for such cases, we propose several options to weaken the nonlinearity. A possibility is to partition it using the given initial data ($\boldsymbol{y}_0 = [\boldsymbol{m}_0, \boldsymbol{p}_0]^{\mathrm{T}}$) to have $\mathcal{N}(\boldsymbol{y}) = \mathcal{N}'(\boldsymbol{y}_0)\boldsymbol{y} + \widetilde{N}(\boldsymbol{y})$, where $\widetilde{N}(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}) - \mathcal{N}'(\boldsymbol{y}_0)y$, and use TDEXP2 to solve the equivalent system $\boldsymbol{y}'(t) = (L + \mathcal{N}'(\boldsymbol{y}_0))y + \widetilde{N}(\boldsymbol{y})$. Another possibility which may be more practical is to partition the original nonlinearity $\mathcal{N}(\boldsymbol{y})$ in such a way that this yields a new linear part L_{new} which has again a diagonal structure like L and a new nonlinearity N_{new} which should be weaker than $\mathcal{N}$ (in the sense that its Lipschitz constant is smaller than that of $\mathcal{N}$). Guided by this observation, we suggest to reformulate (3.3) as

$$\boldsymbol{y}'(t) = F(\boldsymbol{y}) = (L + cI)\boldsymbol{y} + \big(\mathcal{N}(\boldsymbol{y}) - c\boldsymbol{y}\big) = L_{\mathrm{new}}\boldsymbol{y} + N_{\mathrm{new}}(\boldsymbol{y}), \tag{3.20}$$

where I is the identity matrix and

$$\begin{aligned} L_{\mathrm{new}} &= L + cI = \mathrm{diag}[-\gamma_1 + c, \dots, -\gamma_N + c, -\mu_1 + c, \dots, -\mu_N + c], \\ N_{\mathrm{new}}(\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{y}) - c\boldsymbol{y}, \end{aligned} \tag{3.21}$$

and $c > 0$ is a constant. To determine c, one could first find an upper bound c^* for it based on performing a (linear) stability analysis and the information of the nonlinearity $\mathcal{N}(\boldsymbol{y})$. For instance, for the linear system $\boldsymbol{y}'(t) = L_{\mathrm{new}}\boldsymbol{y}(t)$ to be stable, one should require $c \le \min \gamma_i = c^*$. An optimal value for c can be then chosen in $(0, c^*]$ by measuring the accuracy of TDEXP2.

The construction of higher-order exponential methods like TDEXP2 for the regulatory system (3.1) is not straightforward and is out of the scope of this work. We will focus on this in our future work. Instead, for simulating (3.1) with higher-order methods, we suggest the recent developed ExpRK and ExpRB methods and propose an efficient approach to apply ExpRK methods to (3.1). To apply these high-order schemes, it is assumed, without further mention, that the solution $\boldsymbol{m}(t)$ and $\boldsymbol{p}(t)$ are sufficiently smooth and $R(\boldsymbol{p})$, $T(\boldsymbol{m})$ are sufficiently differentiable (with bounded derivatives) in strip along the exact solution.

3.3. Application of exponential Runge–Kutta methods

We first note that, using the reformulated form (3.3), one can apply explicit ExpRK methods [23–25] to (3.1), which are multistage (s-stage) schemes of the form

$$\boldsymbol{Y}_{ni} = \boldsymbol{y}_n + c_i h\varphi_1(c_i hL)F(\boldsymbol{y}_n) + h\sum_{j=2}^{i-1} a_{ij}(hL)D_{nj},\ \ 2 \le i \le s, \tag{3.22a}$$

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h\varphi_1(hL)F(\boldsymbol{y}_n) + h\sum_{i=2}^{s} b_i(hL)D_{ni}, \tag{3.22b}$$

where

$$D_{ni} = \mathcal{N}(\boldsymbol{Y}_{ni}) - \mathcal{N}(\boldsymbol{y}_n), \qquad 2 \le i \le s. \tag{3.23}$$

Here, $\boldsymbol{Y}_{ni} \approx \boldsymbol{y}(t_n + c_i h)$ using nodes c_i and the coefficients $a_{ij}(z)$ and $b_i(z)$ are usually linear combinations of the well-known φ functions

$$\varphi_k(z) = \int_0^1 \mathrm{e}^{(1-\theta)z}\frac{\theta^{k-1}}{(k-1)!}\,\mathrm{d}\theta, \quad k \ge 1. \tag{3.24}$$

ExpRK methods of the form (3.22) have been constructed up to order 5. Since the fourth-order RK is widely used for time-dependent problems as well as for the system (3.1), we propose to use the following fourth-order ExpRK method (constructed in [23] and named as `ExpRK4s5` in [25]), for later comparison.

$$\begin{aligned}
\boldsymbol{Y}_{n2} &= \boldsymbol{y}_n + h\tfrac{1}{2}\varphi_1(\tfrac{1}{2}hL)F(\boldsymbol{y}_n),\\
\boldsymbol{Y}_{n3} &= \boldsymbol{y}_n + h\tfrac{1}{2}\varphi_1(\tfrac{1}{2}hL)F(\boldsymbol{y}_n) + h\varphi_2(\tfrac{1}{2}hL)D_{n2},\\
\boldsymbol{Y}_{n4} &= \boldsymbol{y}_n + h\varphi_1(hL)F(\boldsymbol{y}_n) + h\varphi_2(hL)(D_{n2} + D_{n3}),\\
\boldsymbol{Y}_{n5} &= \boldsymbol{y}_n + h\tfrac{1}{2}[\varphi_1(\tfrac{1}{2}hL)F(\boldsymbol{y}_n) + \varphi_2(\tfrac{1}{2}hL)(D_{n2} + D_{n3} - \tfrac{1}{2}D_{n4})\\
&\quad + \ \varphi_3(\tfrac{1}{2}hL)(-D_{n2} - D_{n3} + D_{n4})] + h[\varphi_2(hL)\tfrac{1}{4}(D_{n2} + D_{n3} - D_{n4})\\
&\quad + \ \varphi_3(hL)(-D_{n2} - D_{n3} + D_{n4})],\\
\boldsymbol{y}_{n+1} &= \boldsymbol{y}_n + h\varphi_1(hL)F(\boldsymbol{y}_n) + h\varphi_2(hL)(-D_{n4} + 4D_{n5}) + h\varphi_3(hL)(4D_{n4} - 8D_{n5}).
\end{aligned} \tag{3.25}$$

For the systems with strong nonlinearities, we again suggest the technique mentioned in Section 3.2.1 above. In this case, `ExpRK4s5` will be used with L_{new} and $\mathcal{N}_{\text{new}}$ (given in (3.21)) in place of L and $\mathcal{N}$, respectively.

3.4. Application of exponential Rosenbrock methods

Another approach which turns out to be efficient and accurate when integrating highly nonlinear time-dependent problems is based on dynamic linearization. This technique is applicable to any problem wherein its linearization captures the majority of the dynamics and often tells us the behavior of the system in the neighborhood of equilibrium points. In the context of using exponential integrators, we suggest ExpRB methods which have been recently developed [26–29] and applied to several applications, see, e.g., [30–35]. The main idea is first to perform a continuous linearization of the nonlinear system in each integration step along the numerical solution $\boldsymbol{y}_n$, leading to a sequence of semilinear systems

$$\boldsymbol{y}'(t) = F(\boldsymbol{y}) = L_n\boldsymbol{y} + \mathcal{N}_n(\boldsymbol{y}), \tag{3.26}$$

where L_n is the Jacobian and $\mathcal{N}_n$ is the new nonlinearity, respectively given by

$$L_n = F'(\boldsymbol{y}_n),\ \ \mathcal{N}_n(\boldsymbol{y}) = F(\boldsymbol{y}) - \mathcal{N}_n\boldsymbol{y}, \tag{3.27}$$

and then apply ExpRK methods to (3.26). This eventually results in ExpRB methods whose formulation have the same form as (3.22) but with L_n and $\mathcal{N}_n$ in place of L and $\mathcal{N}$, respectively. Applying (3.27) to the regulatory system (3.1), we have

$$L_n = L + \mathcal{N}'(\boldsymbol{y}_n),\ \ \mathcal{N}_n(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}) - \mathcal{N}'(\boldsymbol{y}_n)\boldsymbol{y}. \tag{3.28}$$

With the special property that the linear and nonlinear portions are both updated in each time step and that $\mathcal{N}_n'(\boldsymbol{y}_n) = 0$, ExpRB methods have shown to offer better stability than ExpRK methods (since the new nonlinearity $\mathcal{N}_n(\boldsymbol{y})$ becomes smaller than the original one $F(u)$ from step to step), as well as they can attain high-order with a few stages only. For simulating (3.1) in this work, in addition to `TDEXP2` and `ExpRK4s5`, we also propose to use the one-stage second-order exponential Robsenbrock-Euler scheme [26], which will be named here as `ExpRB2`, and the 2-stage fourth-order ExPRB scheme `ExpRB42` proposed in [29].

ExpRB2:

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h\varphi_1(hL_n)F(\boldsymbol{y}_n). \tag{3.29}$$

ExpRB42:

$$\begin{aligned} \boldsymbol{Y}_{n2} &= \boldsymbol{y}_n + \tfrac{3}{4}h\varphi_1(\tfrac{3}{4}hL_n)F(\boldsymbol{y}_n), \\ \boldsymbol{y}_{n+1} &= \boldsymbol{y}_n + h\varphi_1(hL_n)F(\boldsymbol{y}_n) + h\tfrac{32}{9}\varphi_3(hL_n)D_{n2}. \end{aligned} \tag{3.30}$$

Remark 3.3. While these methods are explicit and superconvegent, they require computing L_n, $\varphi_1(hL_n)$, and $\varphi_3(hL_n)$, where L_n is now the full Jacobian matrix of the system. Since L_n is no longer having the diagonal structure like L (due to $L_n = L + \mathcal{N}'(\boldsymbol{y}_n)$), one cannot compute these φ matrix functions directly as done for the implementation of TDEXP2 and ExpRK4s5. Therefore, for an efficient implementation of ExpRB2 and ExpRB42 we have adapted the code expm.m by Higham et al. [36] to use Pade approximation for these φ functions. This is also motivated by the fact that the size of the ODE genetic regulatory systems is usually small (for large-scale systems, the Krylov subspace methods are preferable for computing φ functions).

4. Linear stability analysis

The Dahlquist test equation $y' = \lambda y$ is usually used for studying linear stability analysis of time integration methods when applied to problems of interest. As seen, genetic regulator systems can be cast in the form (3.3) which consists of the linear part (involving the degradation rates of the system) and the nonlinear part (involving transcription/translation nonlinear functions). The important note here is that these two parts do not commute in general. For such case, the Dahlquist test equation might not be sufficient for the analysis. Therefore, we instead consider a scalar version of (3.3) (as suggested in [37])

$$u' = \ell u + \eta(u) = f(u). \tag{4.1}$$

Suppose that there exists an equilibrium (fixed) point u_0 that satisfies $f(u_0) = 0$. Nearby this point, one can perform a linearization of (4.1) using the Taylor expansion

$$f(u) = f(u_0) + f'(u_0)(u-u_0) + \mathcal{O}\big((u-u_0)^2\big) = \big(\ell + \eta'(u_0)\big)(u-u_0) + \mathcal{O}\big((u-u_0)^2\big).$$

Since this could capture sufficient information of the solution in the neighborhood of u_0, one can instead consider a perturbed equation of (4.1)

$$v' = \ell v + \lambda v \tag{4.2}$$

as our new test equation, where $v = u - u_0$ is the perturbation to u_0 and here $\lambda = \eta'(u_0)$ (clearly, the fixed point is stable if $Re(\ell + \lambda) < 0$).

For our considered genetic regulatory systems (2.1), we note that one can assume ℓ and λ are real numbers. The application of the proposed methods to the test Eq. (4.2) yields a difference equation of the form

$$v_{n+1} = R(x,y)v_n, \qquad y = \ell h, \qquad x = \lambda h \tag{4.3}$$

where $R(x,y)$ is the stability function of the method, which measures the growth factor $\dfrac{v_{n+1}}{v_n}$. The stability region is then determined by

$$R_S = \{(x,y) \in \mathbb{R}^2 : |R(x,y)| \le 1\}.$$

Note that, one can easily find the stability functions and their corresponding stability regions for our proposed methods. For instance, applying TDEXP2 to (4.2) gives

$$R(x,y) = 1 + (x+y)\varphi_1(y) + x(x+y)\varphi_2(y) = 1 + (x+y)(\varphi_1(y) + x\varphi_2(y)).$$

Solving $R(x,y) = 1$ shows that the line $x+y = 0$ and the curve $x = -\dfrac{\varphi_1(y)}{\varphi_2(y)}$ are the boundary of TDEXP2's stability region. For ExpRB2 and ExpRB42, it is easy to see that they share the same stability function $R(x,y) = e^{x+y}$ and thus have the same stability region $\{(x,y) \in \mathbb{R}^2 : x+y \le 0\}$. Similarly, it is straightforward (but tedious) to find the $R(x,y)$ and R_S for ExpRK45. We omit the details.

With these calculations in hand, one can easily plot the stability regions of the proposed methods, as shown in Fig. 2 (the entire gray shaded region). For our comparison in the next section, we also include in Figs. 1 and 2 the stability plots of several methods which have been used for genetic regulatory systems, such as the classical RK and TDRK methods of orders 2 and 4, respectively (RK2, RK4, TDRK2, and TDRK4) [15], and the IFRK methods of orders 2 and 4 (IFRK2 and IFRK4) [16,17].

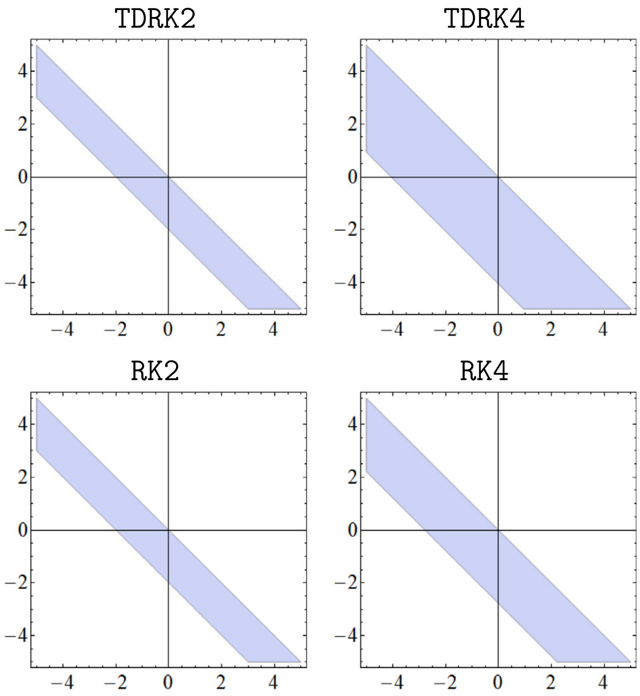

Fig. 1. Stability regions for the classical RK and TDRK methods of orders 2 and 4.

As seen from Fig. 2, all the proposed exponential methods, as well as the IFRK2 and IFRK4 methods, are A-stable since their stability regions contain the entire negative y-axis. However, the ExpRB2 and ExpRB42 have the largest stability region. We also note that only these two methods have entire lower left quadrant shaded on the stability plots.

On the other hand, it can be confirmed through these plots that the classical methods RK2, RK4, TDRK2, and TDRK4 are not A-stable. Fig. 1 tells us that TDRK4 has the largest stability region among these methods, followed by the RK4, TDRK2, and RK2.

5. Numerical experiments

In this section, we evaluate the effectiveness of the proposed and suggested methods in Section 3 (TDEXP2, ExpRB2, ExpRB42, and ExpRK4s5) when applied to the genetic regulatory system (2.1). Specifically, we will perform our experiments on three benchmark test problems: the 2-gene cross-regulation, the 3-gene repressilator, and the 13-dimensional budding yeast cell cycle model, respectively. As mentioned previously, several classical methods have been used for simulating these models so far; most are second- and fourth-order schemes including RK2, RK4, the TDRK methods TDRK2, TDRK4 [38], and IFRK methods IFRK2, and IFRK4 [16,17]. Therefore, we will compare our proposed methods against these methods, particularly in terms of accuracy and efficiency.

For accuracy comparison, same step sizes will be used for all the considered methods. For efficiency comparison, the step sizes will be chosen so that these methods achieve about the same error thresholds. The errors at the final time are computed using the maximum norm.

All the numerical simulations were performed in MATLAB using a MacBook Pro Apple M1, 8 GB 8-core CPU on a single workstation. As the true solution for each test model is unknown, a reference solution is computed by using MATLAB solver ode15s with $\texttt{AbsTol} = \texttt{RelTol} = 2e - 14$.

5.1. The 2-gene cross-regulation model

As the first example, we consider the dynamic equations of a 2-gene regulatory networks. In this case, the translation and degradation of the mRNAs as well as proteins are processes involving the molecular species mRNAs, m_1 and m_2, and proteins, p_1 and p_2 derived from the rate reactions of the two genes. The model is given by:

$$\begin{aligned}\dot{m}_1 &= -\gamma_1 m_1 + \lambda_1 H^{+}(p_2;\theta_2,n_2),\\ \dot{m}_2 &= -\gamma_2 m_2 + \lambda_2 H^{-}(p_1;\theta_1,n_1),\\ \dot{p}_1 &= -\mu_1 p_1 + \kappa_1 m_1,\\ \dot{p}_2 &= -\mu_2 p_2 + \kappa_2 m_2.\end{aligned} \tag{5.1}$$

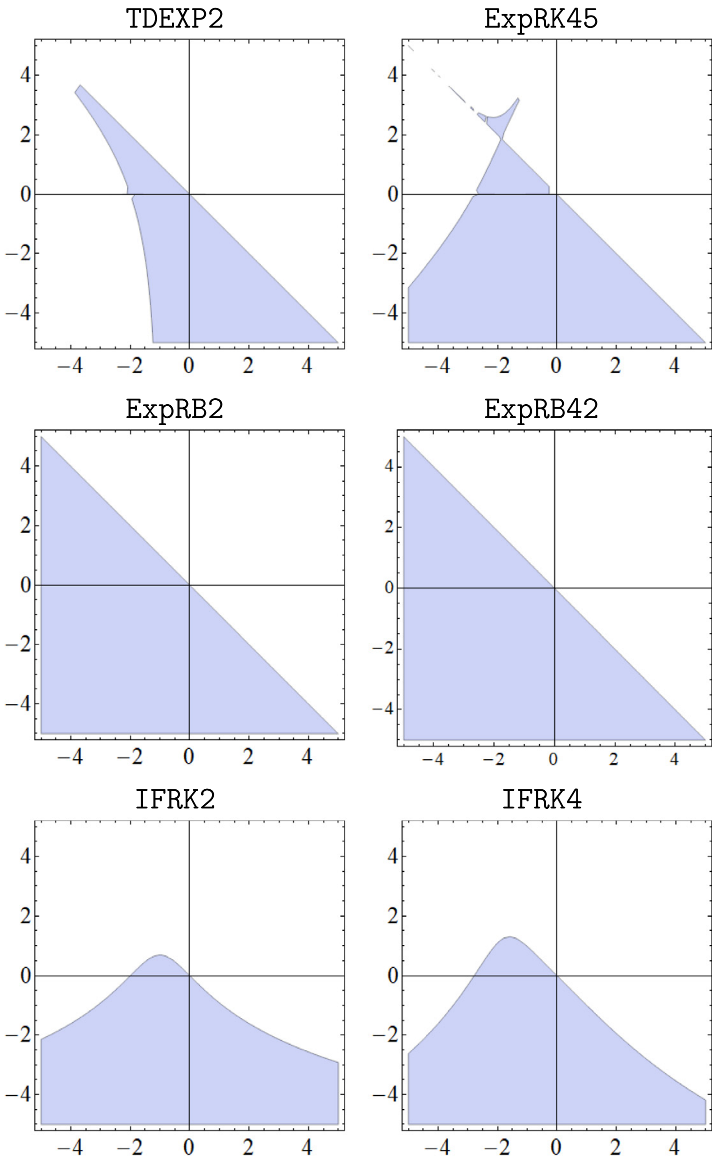

Fig. 2. Stability regions for the IFRK methods and the proposed exponential methods.

where γ_i and μ_i $(i = 1, 2)$ are the degradation rates of the mRNAs and proteins, respectively. Here

$$H^+(p_2;\theta_2,n_2) = \frac{p_2^{n_2}}{\theta_2^{n_2}+p_2^{n_2}}, \quad H^-(p_1;\theta_1,n_1) = \frac{\theta_1^{n_1}}{\theta_1^{n_1}+p_1^{n_1}}$$

are Hill functions for the activation and inhibition, respectively, n_1 and n_2 are the Hill coefficients which depicts the number of protein monomers required for saturation of binding to DNA, and θ_1 and θ_2 are the thresholds.

We first simulate the system (5.1) on the time interval [0, 100] using the initial condition $(m_1(0), m_2(0), p_1(0), p_2(0)) = (0.7, 0.6, 0, 0)$ and parameters (see [6])

$$\gamma_i = 1,\ \mu_i = 1,\ n_i = 3,\ \theta_i = 0.6542,\ \lambda_i = 1.8,\ \kappa_i = 1\ (i = 1, 2)$$

With these values of parameters, this is a nonstiff version of (5.1) (see the time evolution of proteins plotted on the left diagram of Fig. 4).

As seen from Fig. 3 (the two diagrams on the left), while all the considered methods show their expected orders (2 and 4) of convergence (using the same set of step sizes $\{h = 2^{-k}, k = 3, 4, 5, 6\}$), the ExpRB schemes ExpRB2 and ExpRB42 are exceptional with about two orders of magnitude more accurate than the others schemes (showing almost identical

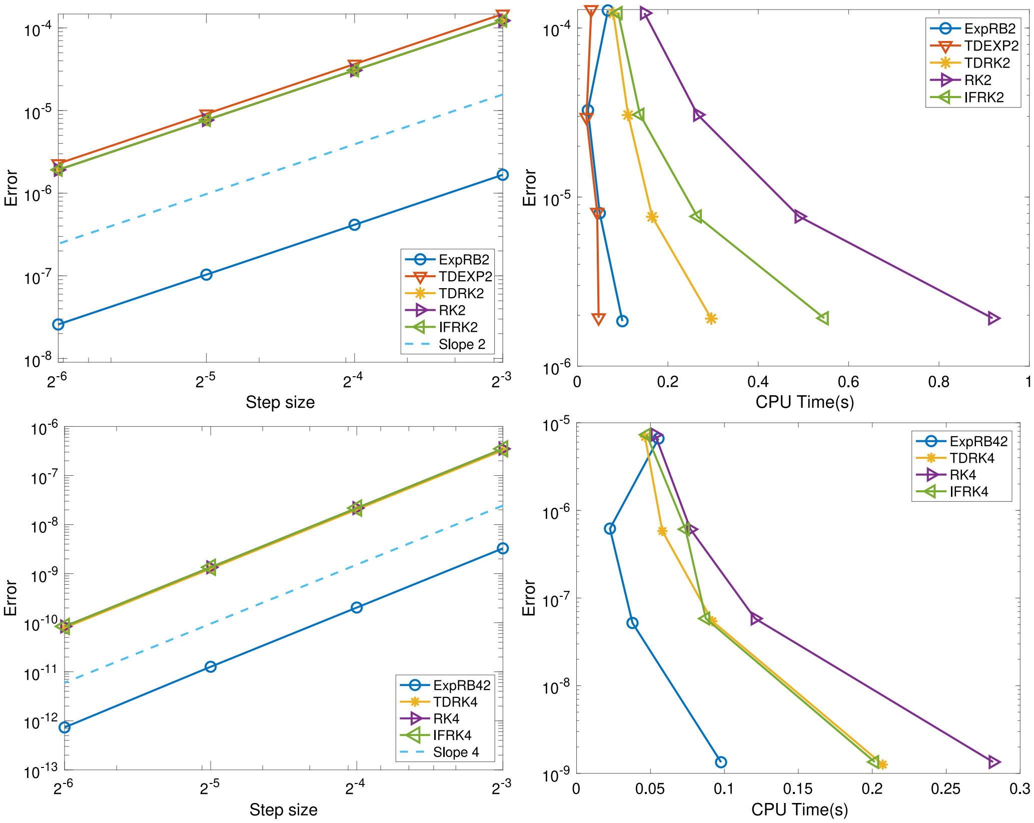

Fig. 3. Accuracy (left) and efficiency (right) plots of the considered methods when applied to a nonstiff model (5.1). To verify convergence rates, straight lines with slopes 2 and 4 were also added.

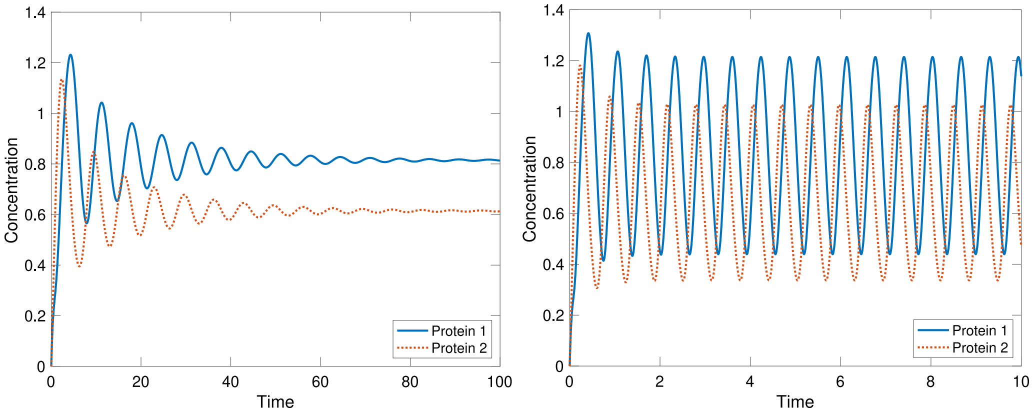

Fig. 4. Time evolution of proteins in the non-stiff two-gene system (left) and stiff two-gene system (right) simulated by TDEXP2.

accuracy). Given about the same level of accuracy for all the second-order methods, however, the right digram shows that new method TDEXP2 is the most efficient one, following closely by ExpRB2. These two methods are much faster than TDRK2, IFRK2, and RK2 (being the slowest one). Similar conclusion is observed for the fourth-order methods, that is now ExpRB42 being the most efficient (note again that a fourth-order exponential method of the format similarly to TDEXP2 is not available yet).

Next, to further demonstrate the accuracy and efficiency of the proposed methods, we increase the stiffness of (5.1) by considering the parameters

$$\gamma_i = 10,\ \ \mu_i = 10,\ \ n_i = 4,\ \ \lambda_i = 18,\ \ \kappa_i = 10\ (i = 1, 2).$$

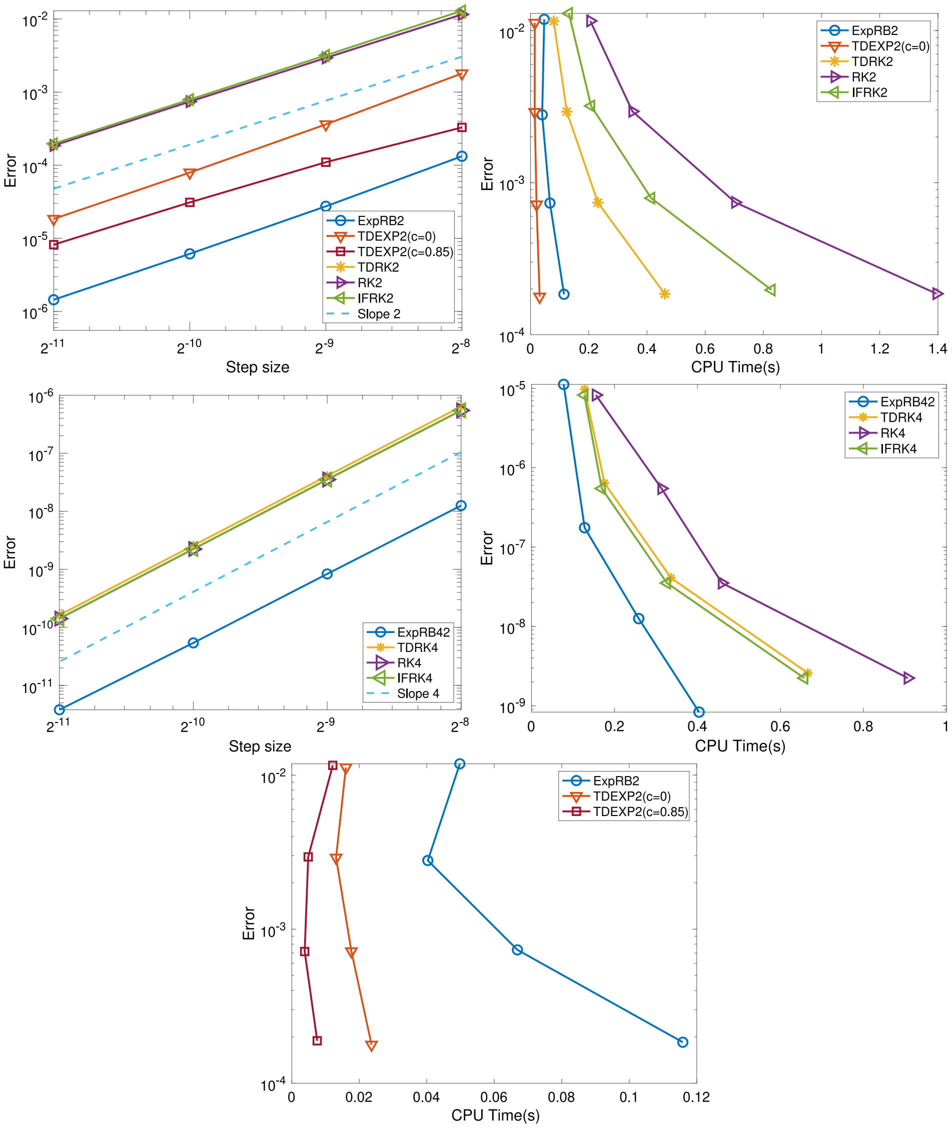

Fig. 5. Accuracy (left) and efficiency (right) plots of the considered methods when applied to a stiffer model of (5.1). To verify convergence rates, straight lines with slopes 2 and 4 were also added. The bottom plot shows the efficiency of TDEXP2 when using the partition idea proposed in Section 3.2.1.

This also makes the nonlinearity becomes stronger ($n_i = 4$). This stiffer model is now integrated on [0, 10] using the same initial condition (see the time evolution of proteins plotted on the right diagram of Fig. 4) as in the nonstiff case.

Again, we see that all the methods show the correct orders of convergence (see the left diagrams of Fig. 5). As predicted by the theory, for this stiffer model, the new second-order exponential method TDEXP2 is now much more accurate than RK2, IFRK2, and TDRK2. On the other hand, ExpRB2 is still the most accurate scheme which can be understood due to its better stability. In terms of efficiency, however, we also see again that the new method TDEXP2 is the fastest one among these second-order methods. In view of the middle diagrams of Fig. 5, we observe a consistent behavior of the fourth-order methods as in the nonstiff model (with ExpRB42 being the most accurate and efficient method).

Finally, as this stiffer model has a stronger nonlinearity, we now test the efficiency of TDEXP2 when using with the partition idea (3.20) proposed in Section 3.2.1. For linear stability, one needs $0 < c \leq c^* = \min\{\gamma_1, \gamma_2, \mu_1, \mu_2\} = 10$. By checking the accuracy of TDEXP2, we found that the smallest error occurs when $c = 0.85$. Using this value of c, we compare the accuracy and efficiency of TDEXP2 itself, both with and without the employment of the partition idea ($c = 0$), in the top left and bottom diagrams of Fig. 5. The results demonstrate that TDEXP2($c = 0.85$) is more accurate and efficient than TDEXP2($c = 0$). Consequently, it outperforms the second most efficient method, ExpRB2, in terms of computing time as well.

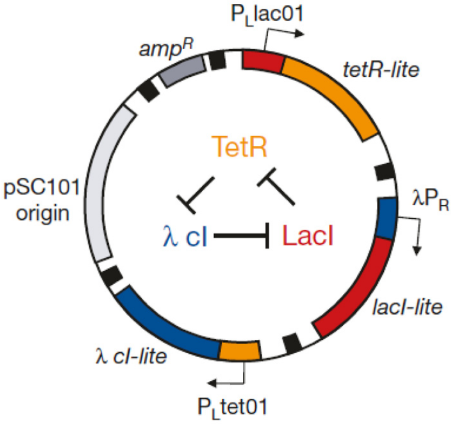

Fig. 6. The repressilator, see [10].

5.2. Repressilator

As the second example, we consider a repressilator model which is a genetic regulatory network consisting of at least one feedback loop with at least 3 genes (LacI from E. coli, tetracycline from Tn10, and cl from λ phage) where each one represses its successor in the loop (see e.g., Fig. 6 which is extracted from a seminal contribution [10]). The dynamics of the repressilator can be described by the following system of ODEs (see also [39]):

$$\begin{aligned}
\dot{m}_{lacl} &= -\gamma_1 m_{lacl} + \frac{\alpha}{1+\left(\frac{p_{cl}}{K_d}\right)^r} + \alpha_0,\\
\dot{m}_{tetR} &= -\gamma_2 m_{tetR} + \frac{\alpha}{1+\left(\frac{p_{lacl}}{K_d}\right)^r} + \alpha_0,\\
\dot{m}_{cl} &= -\gamma_3 m_{cl} + \frac{\alpha}{1+\left(\frac{p_{tetR}}{K_d}\right)^r} + \alpha_0,\\
\dot{p}_{lacl} &= -\mu_1 p_{lacl} + \kappa m_{lacl},\\
\dot{p}_{tetR} &= -\mu_2 p_{tetR} + \kappa m_{tetR},\\
\dot{p}_{cl} &= -\mu_3 p_{cl} + \kappa m_{cl}.
\end{aligned} \tag{5.2}$$

where p_j and m_j ($j \in \{lacl, tetR, cl\}$) are protein concentrations and the mRNA concentrations, respectively. γ_i and μ_i ($i = 1, 2, 3$) are the degradation rates of mRNA and repressors, respectively. κ, α, and α_0 are the translation, transcription, and leakage rates, respectively. Finally, r is the Hill coefficient and K_d is the dissociation constant. For our experiments, we consider (5.2) on the period of 12 h, using the initial value

$$(m_{lacl}(0), m_{tetR}(0), m_{cl}(0), p_{lacl}(0), p_{tetR}(0), p_{cl}(0)) = (0, 0, 0, 150, 0, 0)$$

and the parameters

$$\gamma_i = 17.2, \mu_i = 7.8, \alpha = 248.05, \alpha_0 = 7.15, \kappa = 109.15, r = 4.4, K_d = 123.12.$$

This corresponds to a highly oscillatory repressilator system (see Fig. 7) which has a strong nonlinearity.

In view of the results showing in Fig. 8, we now separately comment on the performance of the considered second-order (the two precision diagrams on the top) and fourth-order methods (the two precision diagrams on the bottom) on this repressilator model. All the second-order methods present their desired orders and we see again the similar trend observed in the 2-gene system that ExpRB2 is the most accurate one (using the same set of time step sizes $h = 2^{-i}$, $i = 7, 8, 9, 10$) and the new method TDEXP2 is the fastest one. IFRK2, TDRK2 and RK2 have almost the same accuracy. Interestingly, while TDEXP2 is just slightly faster than ExpRB2 for the 2-gene model (5.1), it is now much faster than that scheme for this 3-gene model (though they both are 1-stage explicit schemes). This can be explained as follows (based on our Remarks 3.2–3.3). First, in each step, ExpRB2 requires evaluating the Jacobian matrix of the full system and the matrix function φ_1 of it (as noted in Remark 3.3, since this full Jacobian matrix is not a diagonal matrix, Pade approximation is used for approximating φ_1 function). Second, when comparing with the 2-gene model (5.1), we note that the 3-gene model (5.2) has a larger size and involves a stronger nonlinearity. Third, concerning TDEXP2, it requires computing the Jacobian of the nonlinearity instead of the full Jacobian. With the diagonal form of the linear part, it does not require any methods for approximating matrix functions φ.

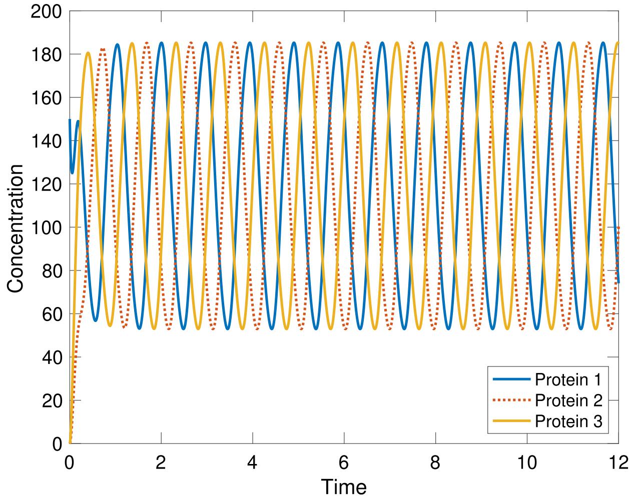

Fig. 7. Time evolution of proteins in the Repressilator.

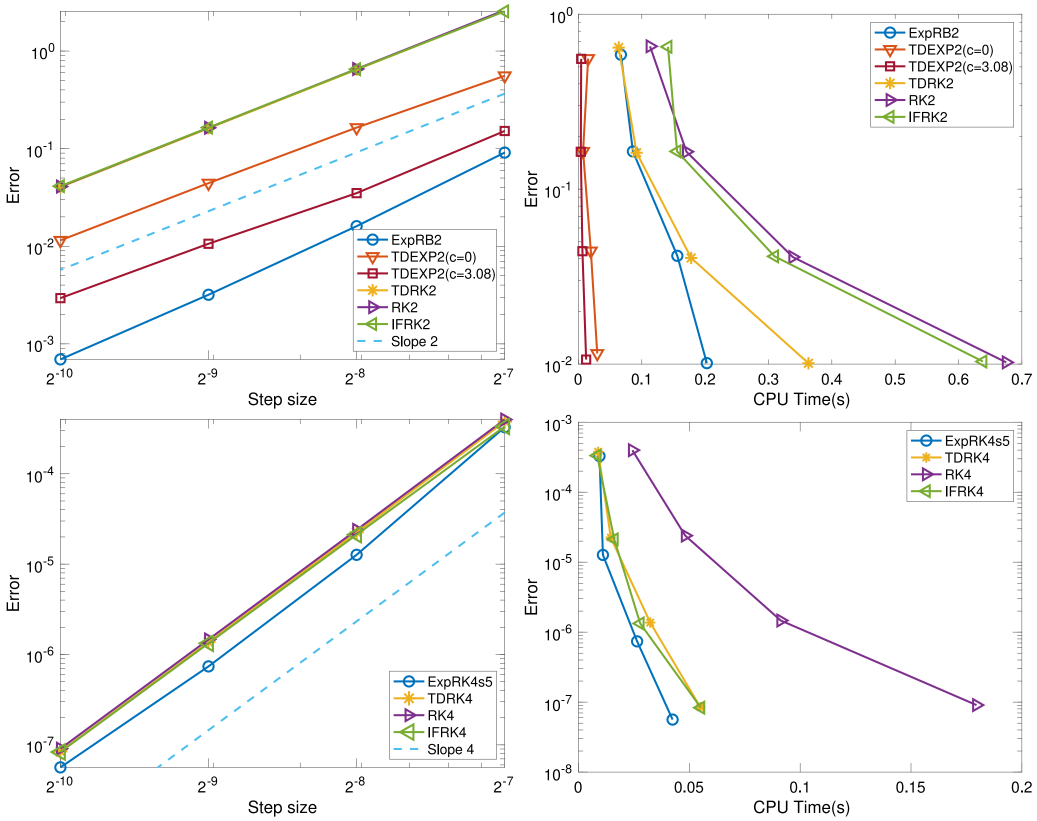

Fig. 8. Accuracy (left) and efficiency (right) plots of the considered methods when applied to the repressilator model (5.2). To verify convergence rates, straight lines with slopes 2 and 4 were also added.

Moreover, due to the strong nonlinearity of the 3-gene model (5.2), we have also implemented TDEXP2 in an efficient way based on the partitioning idea presented in Section 3.2.1 (see (3.20)). For this, we chose c value based on the linear

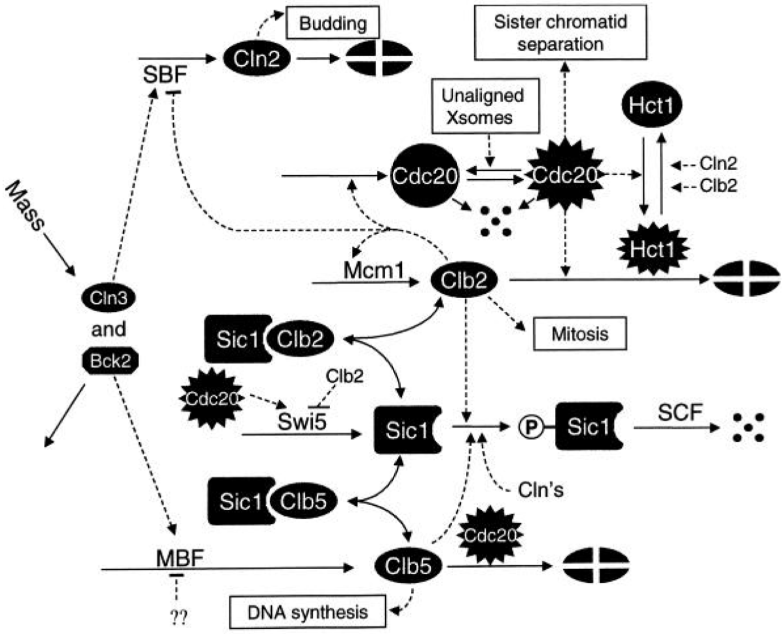

Fig. 9. The consensus model of the cell cycle control mechanism in budding yeast [40].

stability and accuracy of the method as done for the 2-gene above. Specifically, we found $c = 3.08$ as an optimal value. Using this we again compare the accuracy and efficiency of TDEXP2 itself, both with and without the employment of the partition idea ($c = 0$). The results on the top diagrams of Fig. 8 clearly show that TDEXP2($c = 3.08$) is more efficient and much more accurate than TDEXP2($c = 0$).

For fourth-order methods, our experiments showed that ExpRB42 is still the most accurate method, but no longer faster than the TDRK or IFRK methods as we have seen in the 2-gene model (5.1). This can be explained by the similar reasons as for ExpRB2 above (note that both TDRK4 and ExpRB2 use 2-stages, but the latter requires approximating matrix functions φ of the full Jacobian matrix in each step). In this case, we thus suggest using ExpRK4s5 method as it does not require the Jacobian matrix. Guided by Section 3.3, one can implement it efficiently using again the partitioning idea (3.20). As seen from Fig. 8 (bottom), ExpRK4s5 becomes the most accurate and efficient methods among the considered fourth-order methods.

5.3. The budding yeast cell cycle

As the last example, we consider a budding yeast cell cycle model which describes the molecular machinery of the cell cycle control for the budding yeast. The mechanism describes the active phase of cyclin-dependent kinase activities. The cell duplicates all of its parts and distributes them roughly equally between two daughter cells, each of which is equipped with enough information and machinery to carry out the process again. Fig. 9 depicts a potential molecular process for controlling DNA synthesis, bud appearance, mitosis, and cell division in budding yeast. The dynamics of the molecular mechanism can be described by the system of 13 ODEs given in Table 1 (see also [40]). The model includes 50 parameters and their values (for wild-type cells) are specified in Table 2. In Fig. 10, we plot the time evolution of budding yeast cell cycle over the interval [0, 100].

$$\begin{array}{lll}
[\mathrm{Cln2}] = 0.0716 & [\mathrm{Clb2}]_T = 0.0508 & [\mathrm{Clb5}]_T = 0.0875 \\
[\mathrm{Sic1}]_T = 0.1424 & [\mathrm{Clb2/Sic1}] = 0.0377 & [\mathrm{Clb5/Sic1}] = 0.0585 \\
[\mathrm{Cdc20}]_T = 0.9739 & [\mathrm{Cdc20}] = 0.0908 & [\mathrm{Hct1}] = 0.9878 \\
[\mathrm{mass}] = 0.8544 & [\mathrm{ORI}] = 0 & [\mathrm{BUD}] = 0 \\
[\mathrm{SPN}] = 0 & &
\end{array}$$

In our experiments below, we run the model over the period of 65 mins with the initial values and only perform evaluations among the second-order methods (ExpRB2, TDRK2, RK2) in order to show the efficiency of our proposed method TDEXP2. We have also implemented TDEXP2 both with (see (3.20)) and without using the partition idea ($c = 0$). As done for two previous test models, we determined $c = 0.36$ as an optimal value for the splitting (3.20) of this model. In view of Fig. 11, we observe consistent results as obtained in the 2-gene cross-regulation and the 3-gene repressilator

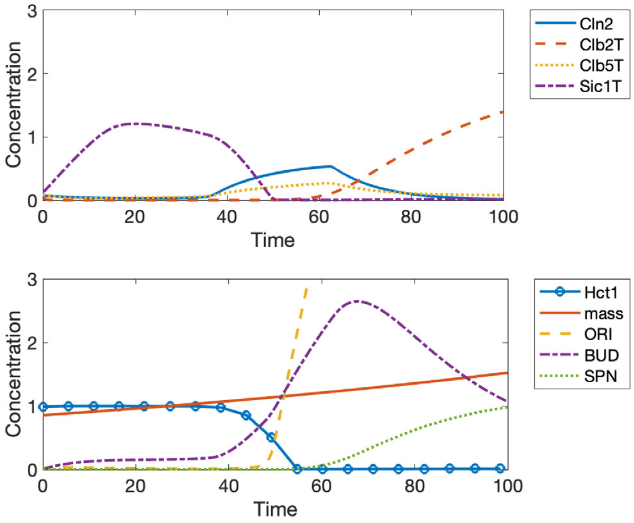

Fig. 10. Time evolution of budding yeast cell cycle.

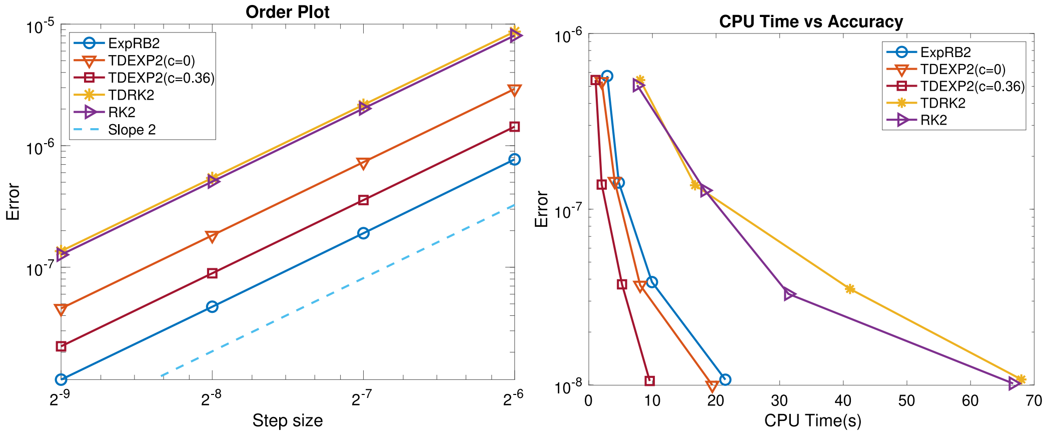

Fig. 11. Accuracy (left) and efficiency (right) plots of the considered methods when applied to the budding yeast cell cycle model (5.2). To verify convergence rates, a straight line with slope 2 is also added.

models. While all the considered second-order methods show the desired order 2, ExpRB2 is the most accurate scheme, followed closely by TDEXP2($c = 0.36$) and then TDEXP2($c = 0$). In terms of efficiency, however, TDEXP2($c = 0.36$) is fastest method.

6. Conclusion

Genetic regulatory systems are networks of molecular interactions that control gene expression, which is crucial for many biological processes, including protein synthesis and the functioning of all living organisms (such as metabolism, cell signaling, and immune response). In this work, we have constructed and analyzed a new second-order exponential method TDEXP2 for simulating genetic regulatory systems modeled by a set of nonlinear ODEs. Moreover, a novel splitting technique has been proposed to improve the stability and increase accuracy and efficiency of the new method (as well as for high-order exponential Runge–Kutta methods) when applied to systems with high nonlinearities. In addition, we suggest the use of a second- and fourth-order exponential Rosenbrock methods based on dynamic linearization of the system. The linear stability analysis of the proposed methods is also presented. Our numerical experiments on various models, including a 2-gene and 3-gene repressilator models and a budding yeast cell cycle model demonstrate the effectiveness of the proposed methods. Our future work will focus on constructing high order exponential methods of type TDEXP2 and consider PDE models for genetic regulatory systems.

Table 1
Mathematical model of the budding yeast cell cycle.

$$\frac{\mathrm{d}}{\mathrm{dt}}[\mathrm{Cln2}] = (k'_{s,n2} + k''_{s,n2}[\mathrm{SBF}]).\ \mathrm{mass} - k_{d,n2}[\mathrm{Cln2}]$$

$$\frac{\mathrm{d}}{\mathrm{dt}}[\mathrm{Clb2}]_T = (k'_{s,b2} + k''_{s,b2}[\mathrm{Mcm1}]).\mathrm{mass} - V_{d,b2}[\mathrm{Clb2}]_T$$

$$V_{d,b2} = k'_{d,b2}([\mathrm{Hct1}]_T - [\mathrm{Hct1}]) + k''_{d,b2}[\mathrm{Cdc20}]$$

$$\frac{\mathrm{d}}{\mathrm{dt}}[\mathrm{Clb5}]_T = (k'_{d,b5} + k''_{d,b5}[\mathrm{MBF}]).\mathrm{mass} - V_{d,b5}[\mathrm{Clb5}]_T,\ V_{d,b5} = k'_{d,b5} + k''_{d,b5}[\mathrm{Cdc20}]$$

$$[\mathrm{Bck2}] = [\mathrm{Bck2}]^0.\mathrm{mass}\ ,\ [\mathrm{Cln3}]^* = [\mathrm{Cln3}]_{\mathrm{max}}\frac{D_{n3}\mathrm{mass}}{J_{n3} + D_{n3}\mathrm{mass}},\ [\mathrm{Clb2}]_T = [\mathrm{Clb2}] + [\mathrm{Clb2/Sic1}]$$

$$[\mathrm{Clb5}]_T = [\mathrm{Clb5}] + [\mathrm{Clb2/Sic1}]\ ,\ [\mathrm{Sic1}]_T = [\mathrm{Sic1}] + [\mathrm{Clb2/Sic1}] + [\mathrm{Clb5/Sic1}]$$

$$\frac{\mathrm{d}}{\mathrm{dt}}[\mathrm{Sic1}]_T = k'_{s,c1} + k''_{s,c1}[\mathrm{Swi5}] - \left(k_{d1,c1} + \frac{V_{d2,c1}}{J_{d2,c1} + [\mathrm{Sic1}]_T}\right)[\mathrm{Sic1}]_T$$

$$\frac{\mathrm{d}}{\mathrm{dt}}[\mathrm{Clb2/Sic1}] = k_{as,b2}[\mathrm{Clb2}][\mathrm{Sic1}] - \left(k_{di,b2} + V_{d,b2} + k_{d1,c1} + \frac{V_{d2,c1}}{J_{d2,c1} + [\mathrm{Sic1}]_T}\right)[\mathrm{Clb2/Sic1}]$$

$$\frac{\mathrm{d}}{\mathrm{dt}}[\mathrm{Clb5/Sic1}] = k_{as,b5}[\mathrm{Clb5}][\mathrm{Sic1}] - \left(k_{di,b5} + V_{d,b5} + k_{d1,c1} + \frac{V_{d2,c1}}{J_{d2,c1} + [\mathrm{Sic1}]_T}\right)[\mathrm{Clb5/Sic1}]$$

$$V_{d2,c1} = k_{d2,c1}(\epsilon_{c1,n3}[\mathrm{Cln3}] * + \epsilon_{c1,k2}[\mathrm{Bck2}] + [\mathrm{Cln2}] + \epsilon_{c1,b5}[\mathrm{Clb5}] + \epsilon_{c1,b2}[\mathrm{Clb2}])$$

$$\frac{\mathrm{d}}{\mathrm{dt}}[\mathrm{Cdc20}]_T = k'_{s,20} + k''_{s,20}[\mathrm{Clb2}] - k_{d,20}[\mathrm{Cdc20}]_T$$

$$\frac{\mathrm{d}}{\mathrm{dt}}[\mathrm{Cdc20}] = k_{a,20}([\mathrm{Cdc20}]_T - [\mathrm{Cdc20}]) - (V_{i,20} + k_{d,20})[\mathrm{Cdc20}]$$

$$V_{i,20} = \begin{cases} k'_{i,20} & \text{for END\_M+12min} < t < \text{START\_S} \\ k''_{i,20} & \text{for START\_S} < t < \text{END\_M} \end{cases}$$

$$\frac{\mathrm{d}}{\mathrm{dt}}[Hct1] = \frac{(k'_{a,t1} + k''_{a,t1}[\mathrm{Cdc20}])([\mathrm{Hct1}]_T - [\mathrm{Hct1}])}{J_{a,t1} + [\mathrm{Hct1}]_T - [\mathrm{Hct1}]} - \frac{V_{i,t1}[\mathrm{Hct1}]}{J_{i,t1} + [\mathrm{Hct1}]}$$

$$V_{i,t1} = k'_{i,t1} + k''_{i,t1}([\mathrm{Cln3}]^* + \epsilon_{i,t1,n2}[\mathrm{Cln2}] + \epsilon_{i,t1,b5}[\mathrm{Clb5}] + \epsilon_{i,t1,b2}[\mathrm{Clb2}])$$

$$\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{mass} = \mu.\mathrm{mass},\ \frac{\mathrm{d}}{\mathrm{dt}}[\mathrm{ORI}] = k_{s,ori}([\mathrm{Clb5}] + \epsilon_{ori,b2}[\mathrm{Clb2}]) - k_{d,ori}[\mathrm{ORI}]$$

$$\frac{\mathrm{d}}{\mathrm{dt}}[\mathrm{BUD}] = k_{s,bud}([\mathrm{Cln2}] + [\mathrm{Cln3}]^* + \epsilon_{bud,d5}[\mathrm{Clb5}]) - k_{d,bud}[\mathrm{BUD}]$$

$$\frac{\mathrm{d}}{\mathrm{dt}}[\mathrm{SPN}] = k_{s,spn}\frac{[\mathrm{Clb2}]}{J_{spn} + [\mathrm{Clb2}]} - k_{d,spn}[\mathrm{SPN}]$$

$$[\mathrm{SBF}] = [\mathrm{MBF}] = G(V_{a,sbf}, k'_{i,sbf} + k''_{i,sbf}[\mathrm{Clb2}], J_{a,sbf}, J_{i,sbf})$$

$$V_{a,sbf} = k_{a,sbf}([\mathrm{Cln2}] + \epsilon_{sbf,n3}([\mathrm{Cln3}]^* + [\mathrm{Bck2}]) + \epsilon_{sbf,d5}[\mathrm{Clb5}]),$$

$$[\mathrm{Mcm1}] = G(k_{a,mcm}[\mathrm{Clb2}], k_{i,mcm}, J_{a,mcm}, J_{i,mcm})$$

$$[\mathrm{Swi5}] = G(k_{a,swi}[\mathrm{Cdc20}], k'_{i,swi} + k''_{i,swi}[\mathrm{Clb2}], J_{a,swi}, J_{i,swi})$$

$$G(a, b, c, d) = \frac{2ad}{b - a + ab + bc + \sqrt{(b - a + ab + bc)^2 - a(b - a)ad}}$$

Data availability

No data was used for the research described in the article.

Acknowledgments

The authors would like to thank the two anonymous referees for their valuable comments and useful suggestions. The first author gratefully acknowledges the support of the National Science Foundation, United States under awards DMS–2012022 and DMS–2309821. The third author is grateful to the Simons foundation for the IMU-Simons African fellowship grant, the University of Lagos for granting him research leave, and the Department of Mathematics and Statistics at Mississippi State University for hosting him during the fellowship period. We also thank Mississippi State University's Center for Computational Science (CCS) for providing computing resources at the High Performance Computing Collaboratory (HPCC).

Table 2
Kinetic constants for the budding yeast model.

$k'_{s,n2} = 0$	$k'_{s,n2} = 0.05$	$k_{d,n2} = 0.1$	$k'_{s,b2} = 0.002$	$k''_{s,b2} = 0.05$
$k'_{d,b2} = 0.01$	$k''_{d,b2} = 2$	$k'''_{d,b2} = 0.05$	$k'_{s,b5} = 0.006$	$k''_{s,b5} = 0.05$
$k'_{d,b5} = 0.1$	$k''_{d,b5} = 0.25$	$k'_{s,c1} = 0.02$	$k''_{s,c1} = 0.1$	$k_{d1,c1} = 0.01$
$k_{d2,c1} = 0.3$	$k_{as,b2} = 50$	$k_{as,b5} = 50$	$k_{di,b2} = 0.05$	$k_{di,b5} = 0.05$
$k'_{s,20} = 0.005$	$k''_{s,20} = 0.06$	$k_{d,20} = 0.08$	$k_{a,20} = 1$	$k'_{i,20} = 0.1$
$k''_{i,20} = 10$	$k'_{a,t1} = 0.04$	$k''_{a,t1} = 2$	$k'_{i,t1} = 0$	$k''_{i,t1} = 0.64$
$k_{s,ori} = 2$	$k_{s,bud} = 0.3$	$k_{s,spn} = 0.08$	$k_{d,ori} = 0.06$	$k_{d,bud} = 0.06$
$k_{d,spn} = 0.06$	$k_{a,sbf} = 1$	$k_{a,mcm} = 1$	$k_{a,swi} = 1$	$k'_{i,sbf} = 0.5$
$k''_{i,sbf} = 6$	$k'_{i,swi} = 0.3$	$k''_{i,swi} = 0.2$	$k_{i,mcm} = 0.15$	$\mu = 0.005776$
$[\text{Cln3}]_{\max} = 0.02$	$[\text{Bck2}]^0 = 0.0027$	$[\text{Hct1}]_T = 1$	$J_{spn} = 0.2$	$J_{d2,c1} = 0.05$
$J_{a,sbf} = 0.01$	$J_{i,sbf} = 0.01$	$J_{a,mcm} = 1$	$J_{i,mcm} = 1$	$J_{a,swi} = 0.1$
$J_{i,swi} = 0.1$	$J_{a,t1} = 0.05$	$J_{i,t1} = 0.05$		
$\epsilon_{c1,n3} = 20$	$\epsilon_{c1,k2} = 2$	$\epsilon_{c1,b2} = 0.067$	$\epsilon_{c1,b5} = 1$	$\epsilon_{i,t1,n2} = 1$
$\epsilon_{i,t1,b2} = 1$	$\epsilon_{i,t1,b5} = 0.5$	$\epsilon_{ori,b2} = 0.4$	$\epsilon_{bud,d5} = 1$	$\epsilon_{sbf,n3} = 75$
$\epsilon_{sbf,b5} = 0.5$	$f = 0.433$	$J_{n3} = 6$	$D_{n3} = 1$	

References

[1] A. Pandey, M. Mann, Proteomics to study genes and genomes, Nature 405 (6788) (2000) 837–846.

[2] H. De Jong, Modeling and simulation of genetic regulatory systems: a literature review, J. Comput. Biol. 9 (1) (2002) 67–103.

[3] H. De Jong, J. Geiselmann, Modeling and simulation of genetic regulatory networks by ordinary differential equations, Genom. Signal Process. Stat. (2005) 201239.

[4] P. Smolen, D.A. Baxter, J.H. Byrne, Mathematical modeling of gene networks, Neuron 26 (3) (2000) 567–580.

[5] T. Chen, H.L. He, G.M. Church, Modeling gene expression with differential equations, in: Biocomputing'99, World Scientific, 1999, pp. 29–40.

[6] A. Polynikis, S. Hogan, M. di Bernardo, Comparing different ODE modelling approaches for gene regulatory networks, J. Theoret. Biol. 261 (4) (2009) 511–530.

[7] Y. Deng, Parameter optimization of linear ordinary differential equations with application in gene regulatory network inference problems, 2014.

[8] B.C. Goodwin, Oscillatory behavior in enzymatic control processes, Adv. Enzyme Regul. 3 (1965) 425–437.

[9] R. Thomas, R. d'Ari, Biological Feedback, CRC Press, 1990.

[10] M.B. Elowitz, S. Leibler, A synthetic oscillatory network of transcriptional regulators, Nature 403 (6767) (2000) 335–338.

[11] C.P. Fall, E.S. Marland, J.M. Wagner, J.J. Tyson, Computational Cell Biology, Springer, New York, 2002.

[12] S. Widder, J. Schicho, P. Schuster, Dynamic patterns of gene regulation I: simple two-gene systems, J. Theoret. Biol. 246 (3) (2007) 395–419.

[13] K. Iwamoto, H. Hamada, Y. Eguchi, M. Okamoto, Mathematical modeling of cell cycle regulation in response to DNA damage: exploring mechanisms of cell-fate determination, Biosystems 103 (3) (2011) 384–391.

[14] X. You, X. Liu, I.H. Musa, Splitting strategy for simulating genetic regulatory networks, Comput. Math. Methods Med. 2014 (2014) 683235.

[15] Z. Chen, J. Li, R. Zhang, X. You, Exponentially fitted two-derivative Runge–Kutta methods for simulation of oscillatory genetic regulatory systems, Comput. Math. Methods Med. 2015 (2015) 689137.

[16] J.D. Lawson, Generalized Runge–Kutta processes for stable systems with large Lipschitz constants, SIAM J. Numer. Anal. 4 (3) (1967) 372–380.

[17] R. Zhang, J.O. Ehigie, X. Hou, X. You, C. Yuan, Steady-state-preserving simulation of genetic regulatory systems, Comput. Math. Methods Med. 2017 (2017) 2729683.

[18] U. Alon, An Introduction to Systems Biology: Design Principles of Biological Circuits, Chapman and Hall/CRC, 2006.

[19] A.V. Hill, The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves, J. Physiol. 40 (1910) 4–7.

[20] L. Chen, K. Aihara, Stability of genetic regulatory networks with time delay, IEEE Trans. Circuits Syst. I 49 (5) (2002) 602–608.

[21] L. Chen, R. Wang, C. Li, K. Aihara, Modeling Biomolecular Networks in Cells: Structures and Dynamics, Springer Science & Business Media, 2010.

[22] M. Xiao, J. Cao, Genetic oscillation deduced from hopf bifurcation in a genetic regulatory network with delays, Math. Biosci. 215 (1) (2008) 55–63.

[23] M. Hochbruck, A. Ostermann, Explicit exponential Runge–Kutta methods for semilinear parabolic problems, SIAM J. Numer. Anal. 43 (2005) 1069–1090.

[24] V.T. Luan, A. Ostermann, Explicit exponential Runge–Kutta methods of high order for parabolic problems, J. Comput. Appl. Math. 256 (2014) 168–179.

[25] V.T. Luan, Efficient exponential Runge–Kutta methods of high order: construction and implementation, BIT Numer. Math. 61 (2) (2021) 535–560.

[26] M. Hochbruck, A. Ostermann, J. Schweitzer, Exponential Rosenbrock-type methods, SIAM J. Numer. Anal. 47 (2009) 786–803.

[27] V.T. Luan, A. Ostermann, Exponential Rosenbrock methods of order five–construction, analysis and numerical comparisons, J. Comput. Appl. Math. 255 (2014) 417–431.

[28] V.T. Luan, A. Ostermann, Parallel exponential Rosenbrock methods, Comput. Math. Appl. 71 (2016) 1137–1150.

[29] V.T. Luan, Fourth-order two-stage explicit exponential integrators for time-dependent PDEs, Appl. Numer. Math. 112 (2017) 91–103.

[30] C. Clancy, J. Pudykiewicz, On the use of exponential time integration methods in atmospheric models, Tellus A 65 (2013).

[31] V.T. Luan, J.A. Pudykiewicz, D.R. Reynolds, Further development of efficient and accurate time integration schemes for meteorological models, J. Comput. Phys. 376 (2019) 817–837.

[32] D.L. Michels, V.T. Luan, M. Tokman, A stiffly accurate integrator for elastodynamic problems, ACM Trans. Graph. 36 (4) (2017) 116.

[33] Y.J. Chen, U. Ascher, D. Pai, Exponential Rosenbrock-Euler integrators for elastodynamic simulation, IEEE Trans. Vis. Comput. Graphics (2017).

[34] K. Pieper, K.C. Sockwell, M. Gunzburger, Exponential time differencing for mimetic multilayer ocean models, J. Comput. Phys. 398 (2019) 817–837.

[35] V.T. Luan, D.L. Michels, Efficient exponential time integration for simulating nonlinear coupled oscillators, J. Comput. Appl. Math. 391 (2021) 113429.

[36] N.J. Higham, The scaling and squaring method for the matrix exponential revisited, SIAM J. Matrix Anal. Appl. 26 (4) (2005) 1179–1193.

[37] S.M. Cox, P.C. Matthews, Exponential time differencing for stiff systems, J. Comput. Phys. 176 (2) (2002) 430–455.

[38] R.P. Chan, A.Y. Tsai, On explicit two-derivative Runge-Kutta methods, Numer. Algorithms 53 (2) (2010) 171–194.

[39] Ž. Pušnik, M. Mraz, N. Zimic, M. Moškon, Computational analysis of viable parameter regions in models of synthetic biological systems, J. Biol. Eng. 13 (1) (2019) 1–21.

[40] K.C. Chen, et al., Kinetic analysis of a molecular model of the budding yeast cell cycle, Mol. Biol. Cell 11 (1) (2000) 369–391.