

Article

The Double Phospho/Dephosphorylation Cycle as a Benchmark to Validate an Effective Taylor Series Method to Integrate Ordinary Differential Equations

Alessandro Borri ^{1,*} , Francesco Carravetta ¹ and Pasquale Palumbo ² 

¹ Institute for Systems Analysis and Computer Science “A. Ruberti”, National Research Council of Italy (CNR-IASI), 00185 Rome, Italy; francesco.carravetta@iasi.cnr.it

² Department of Biotechnology and Biosciences, University of Milano-Bicocca, 20126 Milan, Italy; pasquale.palumbo@unimib.it

* Correspondence: alessandro.borri@iasi.cnr.it

Abstract: The double phosphorylation/dephosphorylation cycle consists of a symmetric network of biochemical reactions of paramount importance in many intracellular mechanisms. From a network perspective, they consist of four enzymatic reactions interconnected in a specular way. The general approach to model enzymatic reactions in a deterministic fashion is by means of stiff Ordinary Differential Equations (ODEs) that are usually hard to integrate according to biologically meaningful parameter settings. Indeed, the quest for model simplification started more than one century ago with the seminal works by Michaelis and Menten, and their Quasi Steady-State Approximation methods are still matter of investigation nowadays. This work proposes an effective algorithm based on Taylor series methods that manages to overcome the problems arising in the integration of stiff ODEs, without settling for model approximations. The double phosphorylation/dephosphorylation cycle is exploited as a benchmark to validate the methodology from a numerical viewpoint.

Keywords: systems biology; enzymatic reactions; quadratization; ODE integration



Citation: Borri, A.; Carravetta, F.; Palumbo, P. The Double Phospho/Dephosphorylation Cycle as a Benchmark to Validate an Effective Taylor Series Method to Integrate Ordinary Differential Equations. *Symmetry* **2021**, *13*, 1684. <https://doi.org/10.3390/sym13091684>

Academic Editor: Lorentz Jäntschi

Received: 30 July 2021

Accepted: 8 September 2021

Published: 13 September 2021

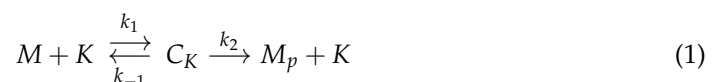
Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

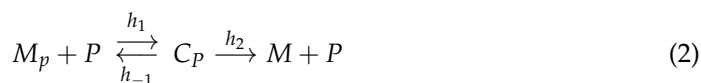
1. Introduction

Protein phosphorylation is a ubiquitous regulatory mechanism for cells, generally working to activate or inactivate molecules [1]. From a biochemical viewpoint, to phosphorylate a molecule consists in the binding of a phosphoryl group PO_4^- [2]. The general model to deal with phosphorylation is to exploit the framework of enzymatic reactions, where the substrate M is supposed to be modified (phosphorylated, actually) into the product M_p by means of the preliminary formation of a complex C provided by the binding of an enzyme (called kinase, K) in charge to catalyze the phosphorylation, see the scheme in (1).

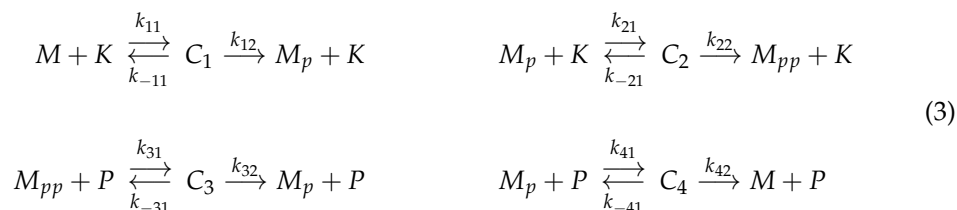


Phosphorylations usually introduce conformational changes that activate/inactivate the enzymatic activity of a protein, or simply prime degradation processes, like the ones involved in the yeast cell cycle (see, e.g., in [3,4] dealing with the degradation of Sic1 and Whi5, respectively) or those involving tyrosine kinase pathways, regulating diverse cellular processes, and whose dysregulation is one of the leading causes of cancer progression [5]. In many important cases, multi-site phosphorylations are required to ensure the correct timing of activation [6,7]. Besides, according to specific conditions and biological frameworks, phosphoryl groups can be removed, making phosphorylation a reversible activation/inactivation mechanism. Moreover, in this case the dephosphorylation may be

treated as a generic enzymatic reaction, with the enzymes that catalyze the reaction called phosphatases P :



Combinations of multiple phosphorylations/dephosphorylations have been selected by nature as devices, providing specific useful biological functions. Within this framework, the Double Phosphorylation/Dephosphorylation Cycle (DPDC) is a symmetric network motif consisting of a double reversible phosphorylation step required to activate a given substrate M :



where the final product is the doubly phosphorylated M_{pp} . Ordinary Differential Equation (ODE) models of DPDC use to combine in a symmetric fashion the four enzymatic reactions [8], providing a bistability regime where the non-phosphorylated or the double-phosphorylated versions of the substrate is predominant [9]. A primary motivation in investigating the computation of the system solutions is that, in a recent paper [10], it has been proven that symmetry breaking in the dynamical solutions of such a network may lead to modify its emergent properties, including concentration robustness of different stationary solutions.

In this context, numerical integration of enzymatic reactions in ODE form has been a matter of investigation for more than a century [11,12], since the seminal works of Michaelis and Menten [13] providing an approximation (the celebrated Quasi Steady-State Approximation (QSSA) [14]) to cope with the double time scale arising whenever biologically meaningful parameters are assigned. Multi-timescale phenomena are very common in biology, see, e.g., the very recent cardiac cell model in [15].

Indeed, in enzymatic networks, the binding/unbinding reactions use to occur at a faster rate, thus leading to stiff ODEs, thus characterized by numerical instability when ordinary numerical schemes are employed for their integration. In particular, in the case of stiff equations, ad hoc integration methods, like the one illustrated in [16] for linear multistep methods, can be employed to approximate the solution efficiently. Like any approximations, there are limitations that may render unfeasible its concrete applicability. Indeed, in [17] we showed how the numerical integration of a basic enzymatic reaction model may create serious problems even to well established procedures like the ones implemented by Matlab in `ode45` and `ode15s` functions. Besides, things become even more crucial in the DPDC, as it has been shown in [18] how the QSSA may miss the bistability property.

In order to deal with stiff ODEs, Taylor Series Methods (TSMs) can be used. These methods build up a polynomial approximation (up to some fixed order k) of the ODE solution around the initial point through Taylor series expansion, which amounts to the recursive calculation of the partial derivatives of the ODE function at the initial point, up to the order k . With respect to standard Runge–Kutta methods, TSMs do not exhibit worse performance in terms of numerical stability, and guarantee a better accuracy in the solution calculation for higher degrees of the approximating polynomial. We refer the interested readers to the works in [19–23] for an in-depth description of TSMs and their numerical properties.

The main issue with TSMs is that, as the approximation order k increases, the calculation of all the required derivatives becomes too cumbersome, and some preliminary transformations need to be applied to the original problem in order to simplify the deriva-

tives calculation. To this end, in the present work, we employ some recent technical results published in [24,25], according to which ODE systems can be embedded into higher-order quadratic equations which, in spite of their dimensionality, allow for more efficient differentiation, which in turn can be exploited for numerical integration via TSM. These results have been already exploited in [17] for the class of simple enzymatic reactions (two differential equations), and are here extended to the more challenging DPDC case (seven differential equations). Simulations are promising as the numerical results show a higher qualitative accuracy of the method with respect to standard off-the-shelf solvers for an appropriate choice of the reaction parameters which make the equations stiff.

The paper is organized as follows. Section 2 reviews the methods involving quadratization and approximate integration of a class of differential equations. Section 3 adapts the framework to the context of Double Phosphorylation/Dephosphorylation Cycle. Section 4 provides some numerical simulations of the DPDC system. Finally, Section 5 offers concluding remarks and ideas for further developments.

2. Exact Quadratization and Approximate Integration of $\sigma\pi$ Differential Equations

This section recollects and adapts some results mainly developed in [24,25] for more general classes of dynamical systems than those developed in this work. Except where differently specified, we adopt the following vector and indices convention: a vector v is always a column vector, and v' is its transpose. With respect to indices, given a vector v , the scalar v_j denotes its j -th entry; instead, in case of double subscripts, like in $Z_{i,j}$, when dealing with a vector we mean a nested notation, where $Z_{i,j}$ is the j -th scalar/vector component of vector Z_i , and vector Z_i is the i -th vector component of vector Z ; on the other hand, when dealing with a matrix, $Z_{i,j}$ refers to the usual scalar entry in row i and column j .

2.1. Exact Quadratization of $\sigma\pi$ -ODEs into Driver-Type Differential Equations

We consider a first-order ODE system

$$\dot{x} = f(x), \quad x \in \mathbb{R}^n, \tag{4}$$

where the function f is a formal polynomial of x , i.e., a polynomial writing where the exponents are allowed to be any real number. More formally, the components of f are in the form

$$f_i(x) = \sum_{l=1}^{v_i} \bar{v}_{i,l} X_{i,l}(x); \quad X_{i,l}(x) = \prod_{r=1}^n x_r^{p_{i,r}^l}, \tag{5}$$

where $p_{i,r}^l$ are real exponents, the v_i quantities $X_{i,l}$ in (5) are named monomials and $\bar{v}_{i,l}$ are real coefficients. We refer to this kind of functions as $\sigma\pi$ -functions, and to the associated system of differential equations as $\sigma\pi$ -ODE.

For $\sigma\pi$ -ODEs, the following theorem holds, by virtue of which the system (4) and (5) can be densely embedded in a higher-dimensional quadratic system.

Theorem 1 (Exact Quadratization Theorem, adapted from the work in [24]). Any $\sigma\pi$ -ODE in the form (4) and (5), with domain $V' \subset \mathbb{R}^n$, is quadratizable on the non empty, open and dense subset $U' = V' \setminus \mathbf{S}$ (with \mathbf{S} denoting the set of all coordinate hyperplanes in \mathbb{R}^n), and the quadratization is given by the following homogeneous Riccati equation in the indeterminates x_i and $Z_{i,l}$:

$$\dot{x}_i = (\bar{v}'_i Z_i) x_i \tag{6}$$

$$\dot{Z}_{i,l} = \sum_{r=1}^n \pi_{i,r}^l (\bar{v}'_r Z_r) Z_{i,l} \tag{7}$$

where $\pi_{i,r}^l = p_{i,r}^l - \delta_{i,r}$, and $\delta_{i,r}$ being the Kronecker symbol: $\delta_{i,r} = 1$ (resp: $\delta_{i,r} = 0$) if $i = r$ (resp: $i \neq r$).

It clearly appears that the specific Riccati equation constituting the quadratization (6) and (7) includes the original state variables, so it is possible to define an augmented state such that the original ODE system (4) is now embedded in the following extended system evolving in \mathbb{R}^m , with $m \geq n$:

$$\dot{x}_i = \sum_{j=1}^m v_{i,j} x_j x_j = (v'_i x) x_i, \tag{8}$$

where, with a little abuse, we keep the notation of x for a vector now living in \mathbb{R}^m , with the first n components provided by the n components of vector x in (4), and the other $m - n$ components provided by vector Z in (7). The coefficients $v_{i,j}$ are suitable linear functions of the coefficients $\bar{v}_{i,l}$ in (5). Equation (8) is called ‘Driver-type’ ODE form, while the matrix $\mathbf{V} \in \mathbb{R}^{m \times m}$ collecting the entries $v_{i,j}$, $i, j = 1, \dots, m$, is called ‘frame’.

The interested reader is referred to the works in [24,25] for further notation (including multi-indices), technical details and full proofs.

2.2. Approximate Taylor Series Integration Method

With a slight abuse of notation, let $t \mapsto x(t)$ be the solution of (4) with initial condition $x(t_0)$, and consider the Taylor expansion of a generic scalar component $x_i(\cdot)$ with respect to the initial time instant t_0 :

$$x_i(t) = \sum_{k=0}^{\infty} c_k(i) \frac{(t - t_0)^k}{k!}, \quad c_k(i) = x_i^{(k)}(t_0). \tag{9}$$

Numerical integration techniques based on the application of the truncated series in (9) to compute a solution of (4) are called Taylor Series Methods (TSMs). Unfortunately, such a series expansion cannot be straightforwardly applied to compute the solution for general systems as it requires the explicit computation of the derivatives of the solution at the initial point, which usually reveals to be too cumbersome. Below is reported a Theorem that allows to compute the coefficients of the Taylor expansion for a system in the ‘Driver-type’ ODE form (8).

Theorem 2 (from [25]). *Consider a ‘Driver-type’ ODE in the form (8). Then, the coefficients $c_k(i)$, $k > 0$, in the Taylor expansion (9) are given by*

$$c_k(i) = \sum_{i_1, \dots, i_k \in \mathcal{S}} v_{i, i_1, \dots, i_k}^{k+1} \cdot x_{i_1}(t_0) x_{i_1}(t_0) \cdots x_{i_k}(t_0), \tag{10}$$

where $i_s \in \mathcal{S} = \{1, \dots, n\}$, for $s = 0, 1, \dots, k$ (we set $i_0 = i$), and the (constant) coefficients $v_{i, i_1, \dots, i_k}^{k+1}$ are given by the following recursive equation

$$v_{i, i_1, \dots, i_k}^{k+1} = v_{i, i_1, \dots, i_{k-1}}^k \left(\sum_{j=0}^{k-1} v_{i_j, i_k} \right), \quad v_i^1 = 1, \tag{11}$$

where the coefficients $v_{i,j}$, $i, j = 1, \dots, m$, are defined in (8).

Approximate numerical integration based on Theorem 2 can be readily performed by truncating the series (9) at a finite order \bar{k} , provided that the integration formula is reinitialized frequently enough to prevent numerical instability. This readily leads to the following iteration. Let $\hat{x}_i(j\Delta)$, $i = 1, \dots, n$, for $j = 0, 1, \dots$, be the approximate value of x_i

at time $j\Delta$ provided by the algorithm, where $t_0 = 0$ and a fixed sampling time $\Delta > 0$ are assumed for simplicity. Then, the approximate solution at all times $t \geq 0$ is given by

$$\begin{cases} \hat{x}_i(t) = \sum_{k=0}^{\bar{k}} c_{k,j}(i) \frac{(t-j\Delta)^k}{k!}, & t \in [j\Delta, (j+1)\Delta), \\ c_{k,j}(i) = \sum_{i_1, \dots, i_k \in \mathcal{S}} v_{i, i_1, \dots, i_k}^{k+1} \cdot \hat{x}_i(j\Delta) \hat{x}_{i_1}(j\Delta) \cdots \hat{x}_{i_k}(j\Delta). \end{cases} \quad (12)$$

Note that, for $\bar{k} = 1$, the proposed integration scheme coincides with the forward Euler method.

3. The Double Phosphorylation–Dephosphorylation Cycle

According to the mass action law, the system of equations governing the dynamics of substrates and complexes in the DPDC scheme in (3) is

$$\begin{aligned} \frac{dM}{dt} &= -k_{11}MK + k_{-11}C_1 + k_{42}C_4 \\ \frac{dM_p}{dt} &= -k_{21}M_pK + k_{-21}C_2 + k_{-41}C_4 - k_{41}M_pP + k_{32}C_3 + k_{12}C_1 \\ \frac{dM_{pp}}{dt} &= -k_{31}M_{pp}P + k_{-31}C_3 + k_{22}C_2 \\ \frac{dC_1}{dt} &= k_{11}MK - (k_{-11} + k_{12})C_1 \\ \frac{dC_2}{dt} &= k_{21}M_pK - (k_{-21} + k_{22})C_2 \\ \frac{dC_3}{dt} &= k_{31}M_{pp}P - (k_{-31} + k_{32})C_3 \\ \frac{dC_4}{dt} &= k_{41}M_pP - (k_{-41} + k_{42})C_4 \end{aligned} \quad (13)$$

where the following conservation laws hold:

$$M + M_p + M_{pp} + C_1 + C_2 + C_3 + C_4 = M_T, \quad (14)$$

$$K + C_1 + C_2 = K_T, \quad P + C_3 + C_4 = P_T. \quad (15)$$

Enzymes K and P can be replaced in (13) according to (15), so that the overall dynamics is not redundant and the system reduces to the following 7-dimensional ODE with the constraint (14):

$$\begin{aligned} \frac{dM}{dt} &= -k_{11}M(K_T - C_1 - C_2) + k_{-11}C_1 + k_{42}C_4 \\ \frac{dM_p}{dt} &= -k_{21}M_p(K_T - C_1 - C_2) + k_{-21}C_2 + k_{-41}C_4 \\ &\quad - k_{41}M_p(P_T - C_3 - C_4) + k_{32}C_3 + k_{12}C_1 \\ \frac{dM_{pp}}{dt} &= -k_{31}M_{pp}(P_T - C_3 - C_4) + k_{-31}C_3 + k_{22}C_2 \\ \frac{dC_1}{dt} &= k_{11}M(K_T - C_1 - C_2) - (k_{-11} + k_{12})C_1 \\ \frac{dC_2}{dt} &= k_{21}M_p(K_T - C_1 - C_2) - (k_{-21} + k_{22})C_2 \\ \frac{dC_3}{dt} &= k_{31}M_{pp}(P_T - C_3 - C_4) - (k_{-31} + k_{32})C_3 \\ \frac{dC_4}{dt} &= k_{41}M_p(P_T - C_3 - C_4) - (k_{-41} + k_{42})C_4 \end{aligned} \quad (16)$$

Summing up all left hand sides of the system (16) yields zero, which, on the other hand, is entailed by (14), accounting that M_T (the total mass) is constant over time. The algebraic equation (14) defines a six-dimensional manifold in \mathbb{R}^7 , invariant with respect to system (16), which means that, if one takes the initial value on this manifold, the evolution of system (16) remains on the same manifold at all times.

Remark 1. *Indeed, it can be readily proved the stronger result that the DPDC system is positive [26], as each non-positive term on the right-hand side of any equation in (16) multiplies the variable differentiated on the left-hand side of the same equation; in short, removal terms are linear in the variable of interest. This prevents the crossing of the coordinate hyperplane \mathbf{S} already defined in Theorem 1, on which the aforementioned terms are zeroed; this will make the DPDC ODEs quadratizable (see the remainder of this section) on the same domain $V' \subseteq \mathbb{R}_{\geq 0}^7$ of the original variables, where the latter symbol denotes the non-negative orthant in \mathbb{R}^7 .*

By setting

$$(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (M, M_p, M_{pp}, C_1, C_2, C_3, C_4),$$

it is readily seen that Equation (16) defines a $\sigma\pi$ -ODE (see Section 2.1), with state dimension $n = 7$, where the exponents in (5) are integer. As a consequence, Theorem 1 can be applied so that the system is exactly quadratized, with the quadratization that can be expressed in the 'Driver-type' ODE form (8), with augmented state dimension $m = 24$, by extending the system with the following adjoint variables, whose dynamics is defined in (7):

$$\begin{aligned} x_8 &= x_1^{-1}x_4; & x_9 &= x_1^{-1}x_7; & x_{10} &= x_2^{-1}x_5; & x_{11} &= x_2^{-1}x_7; & x_{12} &= x_2^{-1}x_6; \\ x_{13} &= x_2^{-1}x_4; & x_{14} &= x_3^{-1}x_6; & x_{15} &= x_3^{-1}x_5; & x_{16} &= x_4^{-1}x_1 & x_{17} &= x_4^{-1}x_1x_5 \\ x_{18} &= x_5^{-1}x_2; & x_{19} &= x_5^{-1}x_2x_4 & x_{20} &= x_6^{-1}x_3; & x_{21} &= x_6^{-1}x_3x_7 \\ x_{22} &= x_7^{-1}x_2; & x_{23} &= x_7^{-1}x_2x_6 & x_{24} &= 1. \end{aligned} \quad (17)$$

Therefore, system (16) rewrites as follows:

$$\begin{aligned} \dot{x}_1 &= (k_{11}x_4 + k_{11}x_5 + k_{-11}x_8 + k_{42}x_9 - k_{11}K_Tx_{24})x_1 \\ \dot{x}_2 &= (k_{21}x_4 + k_{21}x_5 + k_{41}x_6 + k_{41}x_7 + k_{-21}x_{10} + k_{-41}x_{11} + k_{32}x_{12} + k_{12}x_{13} \\ &\quad - (k_{21}K_T + k_{41}P_T)x_{24})x_2 \\ \dot{x}_3 &= (k_{31}x_6 + k_{31}x_7 + k_{-31}x_{14} + k_{22}x_{15} - k_{31}P_Tx_{24})x_3 \\ \dot{x}_4 &= (-k_{11}x_1 + k_{11}K_Tx_{16} - k_{11}x_{17} - (k_{-11} + k_{12})x_{24})x_4 \\ \dot{x}_5 &= (-k_{21}x_2 + k_{21}K_Tx_{18} - k_{21}x_{19} - (k_{-21} + k_{22})x_{24})x_5 \\ \dot{x}_6 &= (-k_{31}x_3 + k_{31}P_Tx_{20} - k_{31}x_{21} - (k_{-31} + k_{32})x_{24})x_6 \\ \dot{x}_7 &= (-k_{41}x_2 + k_{41}P_Tx_{22} - k_{41}x_{23} - (k_{-41} + k_{42})x_{24})x_7 \end{aligned}$$

while we can use the chain rule to compute the dynamics of the adjoint variables, resulting, after some substitutions, into the following additional quadratic differential equations:

$$\begin{aligned}
\dot{x}_8 &= (-k_{11}x_1 - k_{11}x_4 - k_{11}x_5 - k_{-11}x_8 - k_{42}x_9 + k_{11}K_Tx_{16} - k_{11}x_{17} \\
&\quad + (k_{11}K_T - k_{-11} - k_{12})x_{24})x_8 \\
\dot{x}_9 &= (-k_{41}x_2 - k_{11}x_4 - k_{11}x_5 - k_{-11}x_8 - k_{42}x_9 + k_{41}P_Tx_{22} - k_{41}x_{23} \\
&\quad + (k_{11}K_T - k_{-41} - k_{42})x_{24})x_9 \\
\dot{x}_{10} &= (-k_{21}x_2 - k_{21}x_4 - k_{21}x_5 - k_{41}x_6 - k_{41}x_7 - k_{-21}x_{10} - k_{-41}x_{11} - k_{32}x_{12} - k_{12}x_{13} \\
&\quad + k_{21}K_Tx_{18} - k_{21}x_{19} + (k_{41}P_T + k_{21}K_T - k_{-21} - k_{22})x_{24})x_{10} \\
\dot{x}_{11} &= (-k_{41}x_2 - k_{21}x_4 - k_{21}x_5 - k_{41}x_6 - k_{41}x_7 - k_{-21}x_{10} - k_{-41}x_{11} - k_{32}x_{12} - k_{12}x_{13} \\
&\quad + k_{41}P_Tx_{22} - k_{41}x_{23} + (k_{41}P_T + k_{21}K_T - k_{-41} - k_{42})x_{24})x_{11} \\
\dot{x}_{12} &= (-k_{31}x_3 - k_{21}x_4 - k_{21}x_5 - k_{41}x_6 - k_{41}x_7 - k_{-21}x_{10} - k_{-41}x_{11} - k_{32}x_{12} - k_{12}x_{13} \\
&\quad + k_{31}P_Tx_{20} - k_{31}x_{21} + (k_{21}K_T + k_{41}P_T - k_{-31} - k_{32})x_{24})x_{12} \\
\dot{x}_{13} &= (-k_{11}x_1 - k_{21}x_4 - k_{21}x_5 - k_{41}x_6 - k_{41}x_7 - k_{-21}x_{10} - k_{-41}x_{11} - k_{32}x_{12} - k_{12}x_{13} \\
&\quad + k_{11}K_Tx_{16} - k_{11}x_{17} + (k_{21}K_T + k_{41}P_T - k_{-11} - k_{12})x_{24})x_{13} \\
\dot{x}_{14} &= (-k_{31}x_3 - k_{31}x_6 - k_{31}x_7 - k_{-31}x_{14} - k_{22}x_{15} + k_{31}P_Tx_{20} - k_{31}x_{21} \\
&\quad + (k_{31}P_T - k_{-31} - k_{32})x_{24})x_{14} \\
\dot{x}_{15} &= (-k_{21}x_2 - k_{31}x_6 - k_{31}x_7 - k_{-31}x_{14} - k_{22}x_{15} + k_{21}K_Tx_{18} - k_{21}x_{19} \\
&\quad + (k_{31}P_T - k_{-21} - k_{22})x_{24})x_{15} \\
\dot{x}_{16} &= (k_{11}x_1 + k_{11}x_4 + k_{11}x_5 + k_{-11}x_8 + k_{42}x_9 - k_{11}K_Tx_{16} + k_{11}x_{17} \\
&\quad + (k_{-11} + k_{12} - k_{11}K_T)x_{24})x_{16} \\
\dot{x}_{17} &= (k_{11}x_1 - k_{21}x_2 + k_{11}x_4 + k_{11}x_5 + k_{-11}x_8 + k_{42}x_9 - k_{11}K_Tx_{16} + k_{11}x_{17} \\
&\quad + k_{21}K_Tx_{18} - k_{21}x_{19} + (k_{-11} + k_{12} - k_{-21} - k_{22} - k_{11}K_T)x_{24})x_{17} \\
\dot{x}_{18} &= (k_{21}x_2 + k_{21}x_4 + k_{21}x_5 + k_{41}x_6 + k_{41}x_7 + k_{-21}x_{10} + k_{-41}x_{11} + k_{32}x_{12} + k_{12}x_{13} \\
&\quad - k_{21}K_Tx_{18} + k_{21}x_{19} + (k_{-21} + k_{22} - k_{21}K_T - k_{41}P_T)x_{24})x_{18} \\
\dot{x}_{19} &= (-k_{11}x_1 + k_{21}x_2 + k_{21}x_4 + k_{21}x_5 + k_{41}x_6 + k_{41}x_7 + k_{-21}x_{10} + k_{-41}x_{11} + k_{32}x_{12} \\
&\quad + k_{12}x_{13} + k_{11}K_Tx_{16} - k_{11}x_{17} - k_{21}K_Tx_{18} + k_{21}x_{19} \\
&\quad + (k_{-21} + k_{22} - k_{-11} - k_{12} - k_{21}K_T - k_{41}P_T)x_{24})x_{19} \\
\dot{x}_{20} &= (k_{31}x_3 + k_{31}x_6 + k_{31}x_7 + k_{-31}x_{14} + k_{22}x_{15} - k_{31}P_Tx_{20} + k_{31}x_{21} \\
&\quad + (k_{-31} + k_{32} - k_{31}P_T)x_{24})x_{20} \\
\dot{x}_{21} &= (-k_{41}x_2 + k_{31}x_3 + k_{31}x_6 + k_{31}x_7 + k_{-31}x_{14} + k_{22}x_{15} - k_{31}P_Tx_{20} + k_{31}x_{21} \\
&\quad + k_{41}P_Tx_{22} - k_{41}x_{23} + (k_{-31} + k_{32} - k_{-41} - k_{42} - k_{31}P_T)x_{24})x_{21} \\
\dot{x}_{22} &= (k_{41}x_2 + k_{21}x_4 + k_{21}x_5 + k_{41}x_6 + k_{41}x_7 + k_{-21}x_{10} + k_{-41}x_{11} + k_{32}x_{12} + k_{12}x_{13} \\
&\quad - k_{41}P_Tx_{22} + k_{41}x_{23} + (k_{-41} + k_{42} - k_{21}K_T - k_{41}P_T)x_{24})x_{22} \\
\dot{x}_{23} &= (k_{41}x_2 - k_{31}x_3 + k_{21}x_4 + k_{21}x_5 + k_{41}x_6 + k_{41}x_7 + k_{-21}x_{10} + k_{-41}x_{11} + k_{32}x_{12} \\
&\quad + k_{12}x_{13} + k_{31}P_Tx_{20} - k_{31}x_{21} - k_{41}P_Tx_{22} + k_{41}x_{23} \\
&\quad + (k_{-41} + k_{42} - k_{-31} - k_{32} - k_{21}K_T - k_{41}P_T)x_{24})x_{23}
\end{aligned}$$

Finally, we obviously have

$$\dot{x}_{24} = 0$$

initialized to $x_{24}(0) = 1$, implying $x_{24}(t) = 1$ for all $t \geq 0$, which allows to turn linear into quadratic terms in the right-hand side of any $\sigma\pi$ -ODEs.

4. Simulation Results

Numerical simulations of the system (16), quadratized in the form (8) as developed in the previous section, have been performed in the Matlab® suite. Inspired by the parameter choices in [14,17], we consider the following parameters:

$$k_{j1} = 4 \cdot 10^8 \text{M}^{-1} \text{s}^{-1}, \quad k_{-j1} = 25 \text{s}^{-1}, \quad k_{j2} = 15 \text{s}^{-1}, \quad j = 1, 2, 3, 4, \quad (18)$$

with total amounts (14) and (15) equal to

$$M_T = \frac{k_{-11} + k_{12}}{k_{11}}, \quad K_T = P_T = 10^{-3} \cdot M_T. \quad (19)$$

We start from the initial conditions (at time $t_0 = 0$)

$$x_1(0) = 0.70M_T, \quad x_4(0) = 0.25M_T, \quad x_i(0) = 0.01M_T \quad \text{for } i = 2, 3, 5, 6, 7, \quad (20)$$

where the initial states of the adjoint variables $x_j(0)$, for $j = 8, \dots, 24$, are uniquely determined from (17) and (20). The sampling interval has been set to $\Delta = 0.015 \text{ s}$ and the overall simulation time is 1 s. For the given choice of parameters and initial conditions, simulations performed according to the standard ODE45 Matlab® solver, based on the Dormand–Prince Runge–Kutta method [27] with default settings, are aborted by Matlab for not being able to meet integration tolerances.

Figures 1–7 show the numerical solution obtained for variables x_1, \dots, x_7 by means of the ODE15s Matlab® variable-order solver for stiff differential equations [28] compared with the method of quadratization and truncated TSM integration proposed in this paper. It is clearly observed that the substrate trajectories provided by ode15s oscillate, while our approach provides a smoother behavior. Regards to the complexes, ode15s trajectories become negative, which is qualitatively inconsistent by virtue of Remark 1, as we know that the system (16) is positive. The same occurs with C_1 for the TSM method with truncation order $\bar{k} = 1$ (Euler), see Figure 4, and with C_2 for the TSM method with $\bar{k} = 2$, see Figure 5. Instead, the TSM method applied to the quadratized system with truncation order $\bar{k} = 3$ exhibits non-negative solutions for all the variables.

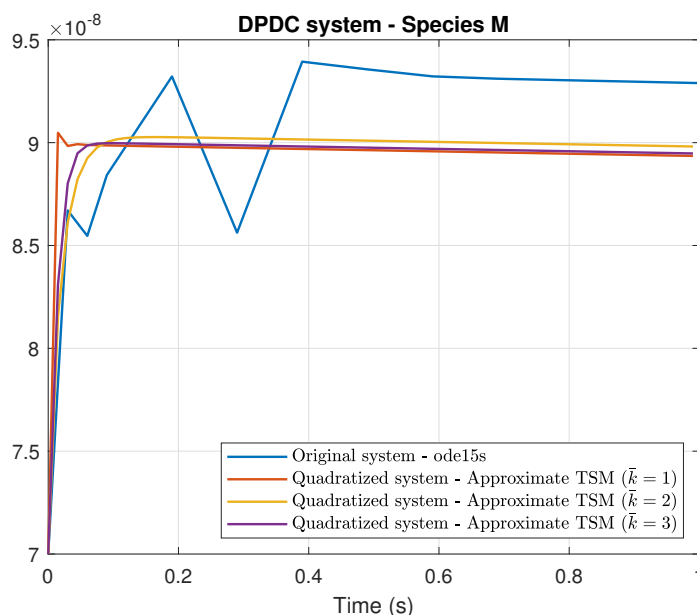


Figure 1. Trajectory of species M via quadratization of Equation (16) and approximate TSM method compared with the Matlab® ODE15s solution. The maximum truncation order has been set to $\bar{k} = 3$.

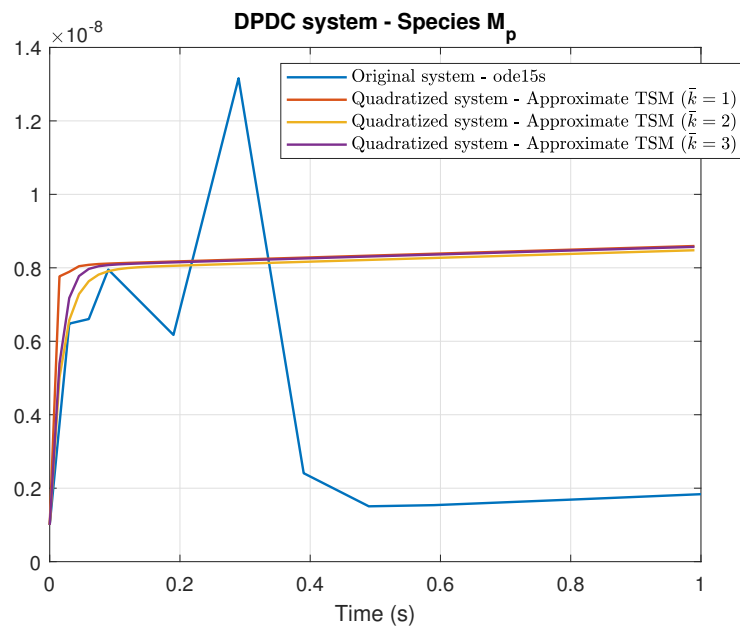


Figure 2. Trajectory of species M_p via quadratization of Equation (16) and approximate TSM method compared with the Matlab® ODE15s solution. The maximum truncation order has been set to $\bar{k} = 3$.

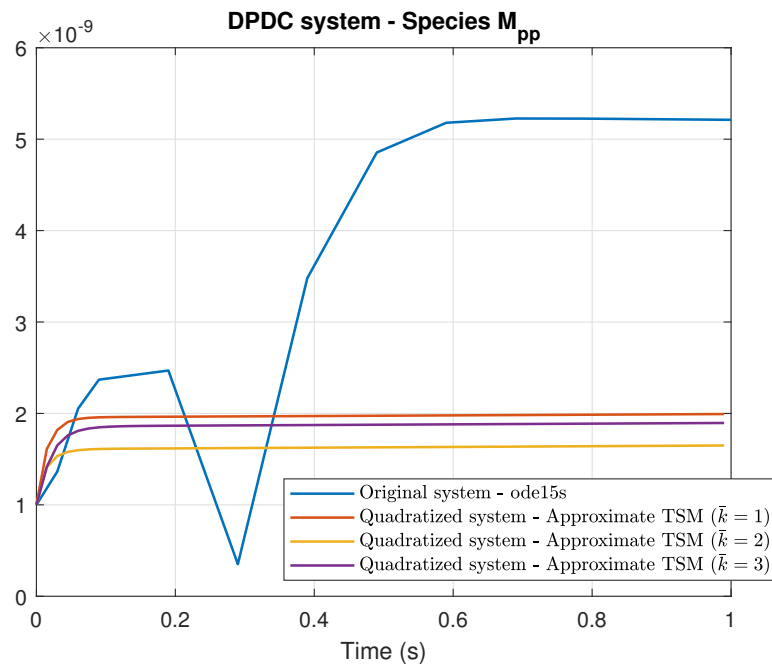


Figure 3. Trajectory of species M_{pp} via quadratization of Equation (16) and approximate TSM method compared with the Matlab® ODE15s solution. The maximum truncation order has been set to $\bar{k} = 3$.

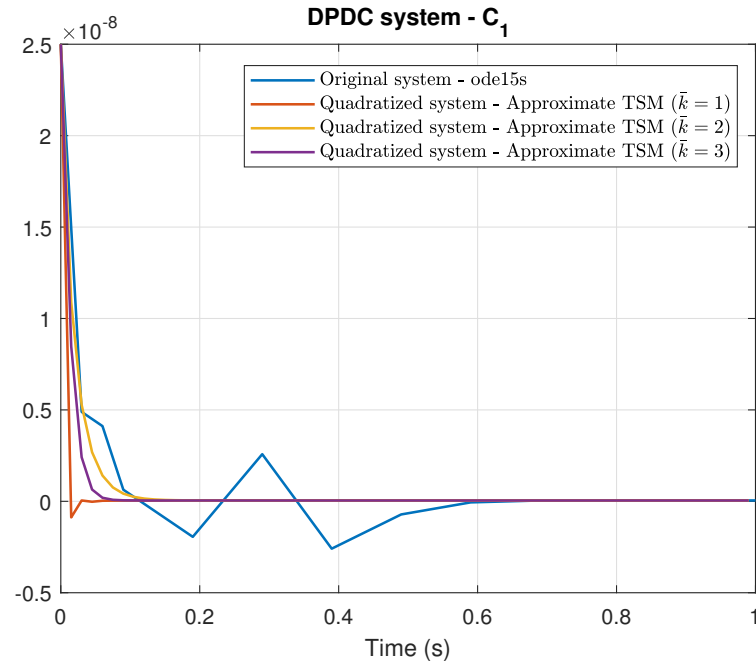


Figure 4. Trajectory of species C_1 via quadratization of Equation (16) and approximate TSM method compared with the Matlab® ODE15s solution. The maximum truncation order has been set to $\bar{k} = 3$.

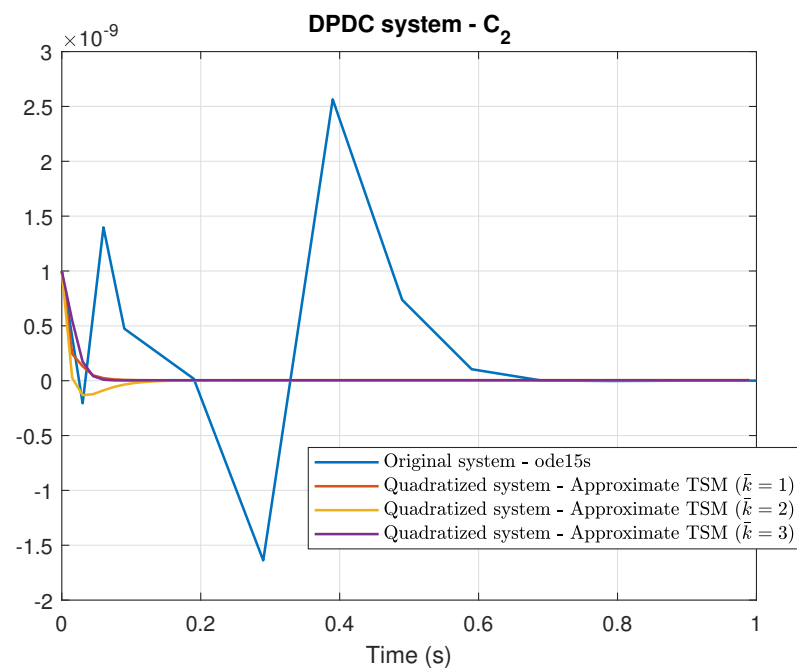


Figure 5. Trajectory of species C_2 via quadratization of Equation (16) and approximate TSM method compared with the Matlab® ODE15s solution. The maximum truncation order has been set to $\bar{k} = 3$.

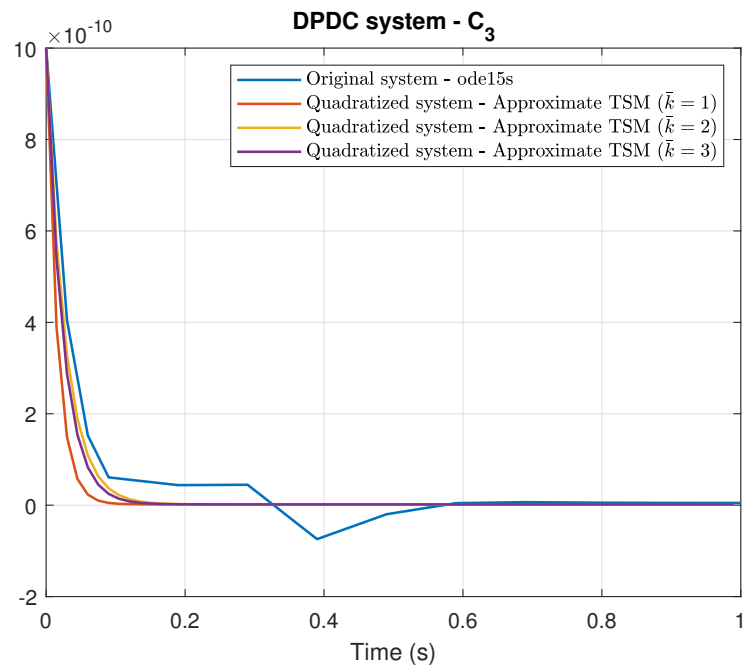


Figure 6. Trajectory of species C_3 via quadratization of Equation (16) and approximate TSM method compared with the Matlab® ODE15s solution. The maximum truncation order has been set to $\bar{k} = 3$.

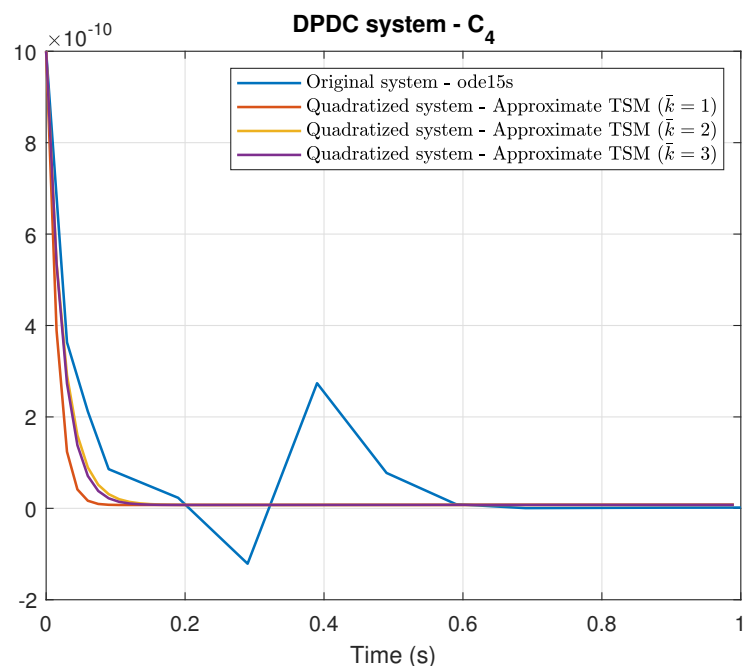


Figure 7. Trajectory of species C_4 via quadratization of Equation (16) and approximate TSM method compared with the Matlab® ODE15s solution. The maximum truncation order has been set to $\bar{k} = 3$.

Figure 8 shows the trajectories of complexes C_1 (top panel) and C_2 (bottom panel), obtained with sampling time $\Delta = 0.015$ s, compared with the same trajectories computed with a doubled sample interval $\Delta = 0.03$ s, and the same truncation order $\bar{k} = 3$ of the TSM method. It is apparent that the choice of a sufficiently small sampling interval, jointly with a sufficiently high truncation order, is crucial for the numerical stability of the algorithm. In particular, both C_1 and C_2 violate again the non-negativity constraint in the case $\Delta = 0.03$ s, as already observed for $\Delta = 0.015$ s and low truncation orders in Figures 4 and 5, with a notable oscillating behavior and an evident initial overshoot of species C_2 . Note that the pattern of damped oscillations in Figures 4, 6, and 8 is not surprising in numerical

simulation and identification in biochemical and biological contexts, for instance, a similar behavior may indeed result from the optimization of the power of the error in multiple linear regression under the assumption of generalized Gauss-Laplace distribution [29].

In summary, the method proposed in this paper seems to be able to return meaningful solutions in the numerical simulation of particular biological conditions when standard solvers may fail.

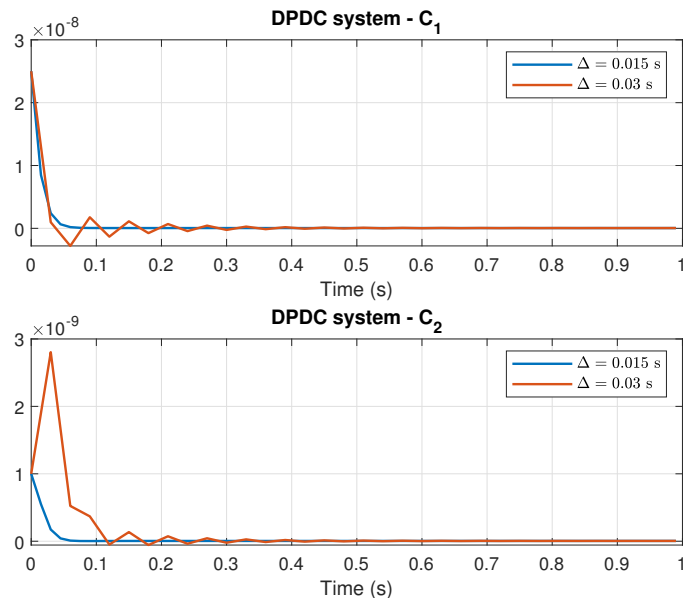


Figure 8. Trajectories of species C_1 (top panel) and C_2 (bottom panel) via quadratization of Equation (16) and approximate TSM method, with truncation order set to $\bar{k} = 3$, for two different values of the sampling interval Δ .

5. Discussion

In this work, we proposed a novel approach to the problem of integrating the solution of biological systems expressed by stiff differential equations (for example, those exhibiting an apparent double time-scale separation). To overcome typical numerical issues related to the numerical integration of this kind of system by means of existing solvers, we here rely on recent work exploiting the so-called *quadratization* of ODEs, which allows embedding the original dynamics in a higher-dimensional space where the system is quadratic and differentiation formulae of any order are computable by means of simple recursions. Such derivatives are exploited within a truncated Taylor Series expansion to build an approximate simple integration scheme, which is proved to work accurately in the *in silico* simulation of the Double Phosphorylation/Dephosphorylation Cycle (DPDC), which is an important regulatory mechanism present in cells. Ongoing and future work is focusing on the construction of a numerical scheme that might overcome the *curse of dimensionality* due to the computation of coefficients of the high-order terms in the Taylor series expansion for the augmented quadratized dynamics.

Author Contributions: Conceptualization, investigation, writing—review and editing, resources, A.B., F.C. and P.P.; writing—original draft preparation, formal analysis, A.B. and P.P.; methodology, F.C.; software, A.B.; supervision, P.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cooper, G. *The Cell, a Molecular Approach*; Sinauer Associates: Sunderland, MA, USA, 2000.
2. Chang, R. *Physical Chemistry for the Chemical and Biological Sciences*; University Science Books: New York, NY, USA, 2000.
3. Nash, P.; Tang, X.; Orlicky, S.; Chen, Q.; Gertler, F.B.; Mendenhall, M.D.; Sicheri, F.; Pawson, T.; Tyers, M. Multisite phosphorylation of a CDK inhibitor sets a threshold for the onset of DNA replication. *Nature* **2001**, *414*, 514–521. [[CrossRef](#)]
4. Palumbo, P.; Vanoni, M.; Cusimano, V.; Busti, S.; Marano, F.; Manes, C.; Alberghina, L. Whi5 phosphorylation embedded in the G1/S network dynamically controls critical cell size and cell fate. *Nat. Commun.* **2016**, *7*, 11372. [[CrossRef](#)]
5. Ionescu, A.E.; Mentel, M.; Munteanu, C.V.A.; Sima, L.E.; Martin, E.C.; Necula-Petrareanu, G.; Szedlacsek, S.E. Analysis of EYA3 Phosphorylation by Src Kinase Identifies Residues Involved in Cell Proliferation. *Int. J. Mol. Sci.* **2019**, *20*, 6307. [[CrossRef](#)]
6. Salazar, C.; Höfer, T. Multisite protein phosphorylation—From molecular mechanisms to kinetic models. *FEBS J.* **2009**, *276*, 3177–3198. [[CrossRef](#)]
7. Salazar, C.; Brümmer, A.; Alberghina, L.; Höfer, T. Timing control in regulatory networks by multisite protein modifications. *Trends Cell Biol.* **2010**, *20*, 634–641. [[CrossRef](#)] [[PubMed](#)]
8. Ortega, F.; Garcés, J.L.; Mas, F.; Kholodenko, B.N.; Cascante, M. Bistability from double phosphorylation in signal transduction. *FEBS J.* **2006**, *273*, 3915–3926. [[CrossRef](#)]
9. Bersani, A.M.; Borri, A.; Carravetta, F.; Mavelli, G.; Palumbo, P. On a stochastic approach to model the double phosphorylation/dephosphorylation cycle. *Math. Mech. Complex Syst.* **2020**, *8*, 261–285. [[CrossRef](#)]
10. Ramesh, V.; Krishnan, J. Symmetry breaking meets multisite modification. *Elife* **2021**, *10*, e65358. [[CrossRef](#)] [[PubMed](#)]
11. Bersani, A.M.; Bersani, E.; Dell’Acqua, G.; Pedersen, M.G. New trends and perspectives in nonlinear intracellular dynamics: One century from Michaelis–Menten paper. *Contin. Mech. Thermodyn.* **2015**, *27*, 659–684. [[CrossRef](#)]
12. Cornish-Bowden, A. One hundred years of Michaelis–Menten kinetics. *Perspect. Sci.* **2015**, *4*, 3–9. [[CrossRef](#)]
13. Michaelis, L.; Menten, M. Kinetics of invertase action. *Biochem. Z.* **1913**, *49*, 333–369.
14. Segel, L. On the validity of the steady state assumption of enzyme kinetics. *Bull. Math. Biol.* **1988**, *50*, 579–593. [[CrossRef](#)]
15. Barrio, R.; Martínez, M.A.; Pérez, L.; Pueyo, E. Bifurcations and Slow-Fast Analysis in a Cardiac Cell Model for Investigation of Early Afterdepolarizations. *Mathematics* **2020**, *8*, 880. [[CrossRef](#)]
16. Mohd Ijam, H.; Ibrahim, Z.B. Diagonally Implicit Block Backward Differentiation Formula with Optimal Stability Properties for Stiff Ordinary Differential Equations. *Symmetry* **2019**, *11*, 1342. [[CrossRef](#)]
17. Borri, A.; Carravetta, F.; Palumbo, P. Time series expansion to find solutions of nonlinear systems: An application to enzymatic reactions. In Proceedings of the 2020 European Control Conference (ECC), St. Petersburg, Russia, 12–15 May 2020; pp. 749–754.
18. Bersani, A.M.; Dell’Acqua, G.; Tomassetti, G. On stationary states in the double phosphorylation-dephosphorylation cycle. In Proceedings of the AIP Conference Proceedings, College Park, MD, USA, 28–31 July 2011; American Institute of Physics: College Park, MD, USA, 2011; Volume 1389.
19. Barton, D.; Willers, I.M.; Zahar, R.V.M. Taylor Series Methods for Ordinary Differential Equations—An Evaluation. In *Mathematical Software*; Rice, J.R., Ed.; Academic Press: New York, NY, USA, 1972; pp. 369–389.
20. Corliss, F.; Chang, Y.F. Solving ordinary differential equations using Taylor Series. *ACM Trans. Math. Softw.* **1994**, *8*, 209–233. [[CrossRef](#)]
21. Corliss, G.F.; Kirlinger, G. On implicit Taylor series method for stiff ODEs. In Proceedings of the SCAN 91: International Symposium on Computer Arithmetic and Scientific Computing, Oldenburg, Germany, 1–4 October 1991.
22. Barton, D. On Taylor series and stiff equations. *ACM Trans. Math. Softw.* **1980**, *6*, 280–294. [[CrossRef](#)]
23. Barrio, R. Sensitivity analysis of ODEs/DAEs using the Taylor series method. *SIAM J. Sci. Comput.* **2006**, *6*, 1929–1947. [[CrossRef](#)]
24. Carravetta, F. Global Exact Quadraticization of Continuous-Time Nonlinear Control Systems. *SIAM J. Control Optim.* **2015**, *53*, 235–261. [[CrossRef](#)]
25. Carravetta, F. On the Solution Calculation of Nonlinear Ordinary Differential Equations via Exact Quadraticization. *J. Differ. Equ.* **2020**, *269*, 11328–11365. [[CrossRef](#)]
26. Farina, L.; Rinaldi, S. *Positive Linear Systems: Theory and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2000.
27. Dormand, J.R.; Prince, P.J. A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.* **1980**, *6*, 19–26. [[CrossRef](#)]
28. Shampine, L.F.; Reichelt, M.W. The MATLAB ODE Suite. *SIAM J. Sci. Comput.* **1997**, *18*, 1–22. [[CrossRef](#)]
29. Jäntschi, L.; Bálint, D.; Bolboacă, S.D. Multiple Linear Regressions by Maximizing the Likelihood under Assumption of Generalized Gauss-Laplace Distribution of the Error. *Comput. Math. Methods Med.* **2016**, *2016*, 8578156. [[CrossRef](#)] [[PubMed](#)]