# Time series expansion to find solutions of nonlinear systems: an application to enzymatic reactions

Alessandro Borri, Francesco Carravetta, and Pasquale Palumbo

Abstract— Mathematical modeling of enzymatic reactions is among the most known and celebrated results provided by mathematicians to the biological field, and dates back to the early twentieth century. From that point ahead many steps have been done, and nowadays several results, achieved addressing the modeling problem from both the deterministic and stochastic approaches, are available, paving the way to fruitful applications in pharmacokinetics and pharmacodynamics. The common denominator has been the search for analytical, approximate or numerical solutions of the underlying systems, since the intrinsic double time scale of enzymatic reactions prevents the use of standard computational methods. In this note, we exploit recent results allowing to write the solution of an Ordinary Differential Equation (ODE) in terms of a time-series expansion. Here we apply these results to the basic enzymatic reaction scheme and prove its efficacy compared to standard numerical tools. To this end, no approximations are exploited, such as the popular Quasi Steady-State Approximation. Simulations are promising and underline the benefits when standard methods seem to fail.

I. INTRODUCTION

This note investigates the basic enzymatic reaction scheme, which refers to a substrate S that is transformed into a product P by means of the catalytic action of an enzyme E that, first, forms a complex C according to a reversible binding/unbinding reaction and, then, releases the product from the complex. Within this framework, we neglect random fluctuations, thus assuming to have a number of molecules of each species large enough to rely on Ordinary Differential Equation (ODE) models describing average concentrations that vary continuously [16]. This process is summarized in the scheme

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} C \overset{k_2}{\longrightarrow} E + P \qquad (1)$$
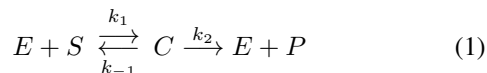
According to standard mass action law, the ODE system associated to (1) is readily written

$$\begin{cases} \dot{s} & = -k_1 es + k_{-1}c \\ \dot{e} & = -k_1 es + (k_{-1} + k_2)c \\ \dot{c} & = k_1 es - (k_{-1} + k_2)c \\ \dot{p} & = k_2 c \end{cases} \qquad (2)$$

where s, e, c, p denote the four species concentrations. Because of the mass conservation laws involving the total

All authors are with the Istituto di Analisi dei Sistemi ed Informatica 'A. Ruberti', Consiglio Nazionale delle Ricerche (IASI-CNR), 00185 Rome, Italy. {alessandro.borri, francesco.carravetta, pasquale.palumbo}@iasi.cnr.it

concentration of substrate and enzyme (S_T and E_T, respectively), the following constraints arise

$$e + c = E_T, \qquad s + c + p = S_T, \qquad (3)$$

so that the ODE (2) simplifies with respect to the independent species as

$$\begin{cases} \dot{s} & = -k_1(E_T - c)s + k_{-1}c \\ \dot{c} & = k_1(E_T - c)s - (k_{-1} + k_2)c \end{cases} \qquad (4)$$

Such a basic chemical reaction network (1) works as a model of any generic enzymatic reaction scheme. This fact explains the huge interest to build up a predictive mathematical model, since enzymes are known to play an important role in many and diverse cellular activities (like metabolism, signal transduction and cell regulation), and their malfunction may lead to the raise of serious diseases [7], [8]. Further interest has recently gained from the practical applications of enzymes as specific catalysts in drug development, food processing and biofuel production [11].

Within this framework, the ODE model (2) has been a matter of intense investigation, strongly related to the so-called Quasi Steady-State Approximation (QSSA), originally developed in the first years of the past century by Michaelis and Menten in [12], and further rigorously derived in [3]. Indeed, according to biologically meaningful parameter assessments, system (4) exhibits a double time scale with consequent difficulties in achieving computationally efficient solutions because of the stiffness of the ODE numerical integration. This drawback arises also when dealing with the stochastic numerical counterpart provided by the Stochastic Simulation Algorithm, [9], [10]. The standard QSSA (and its related modified versions) exploits time-scale separation to project models of biochemical reactions onto lower-dimensional slow manifolds, thus rapidly fluctuating species are not simulated explicitly. Evidently, working hypotheses are provided for the correct applicability of the QSSA (see, e.g. [13], [2], [1]).

This note addresses the problem of finding the exact solution of the original ODE system (4) without any approximating hypotheses. Indeed, our work is based on the application of the quadratic immersion, a tool that allows to embed a wide class of nonlinear systems onto an extended state space according to which the corresponding extended state undergoes a homogeneous quadratic dynamics [4]. Once the system is written in the quadratic form, the analytical solution may be written as a time series expansion, thus providing the exact solution according to a degree of

accuracy which increases with the truncation order of the series. The general theory related to the computation of the coefficients of the time-series expansion has been developed in [5], with the main theorem here reported for the ease of the reader. Numerical simulations are promising, since they show how the proposed approach allows to overcome critical drawbacks that arise in attempting numerical solutions from (4), both according to built-in numerical tools (like Matlab® ODE45 and ODE15s functions) and to a numerical Euler *ad-hoc* implementation.

II. QUADRATIZATION OF THE ENZYMATIC REACTIONS INTO A DRIVER-TYPE ODE

Consider the enzymatic reaction scheme and define $x = [s \quad c]' \in \mathbb{R}^2$, so that the ODE model may be re-written as:

$$\begin{cases} \dot{x}_1 & = -k_1 E_T x_1 + k_1 x_1 x_2 + k_{-1} x_2 \\ \dot{x}_2 & = k_1 E_T x_1 - k_1 x_1 x_2 - (k_{-1} + k_2) x_2 \end{cases} \quad (5)$$

According to [4], [5] (which we refer readers to for details), both dynamics described by (5) can be written as a sum of monomials, i.e.:

$$\dot{x}_i = \sum_{l=1}^{\nu_i} \alpha_{i,l} X_{i,l}, \qquad X_{i,l} = x_1^{p_{i,1}^l} x_2^{p_{i,2}^l}, \qquad i = 1, 2 \quad (6)$$

with $\nu_1 = \nu_2 = 3$,

$$\alpha_{1,1} = -k_1 E_T, \qquad \alpha_{1,2} = k_1, \qquad \alpha_{1,3} = k_{-1},$$
$$\alpha_{2,1} = k_1 E_T, \qquad \alpha_{2,2} = -k_1, \qquad \alpha_{2,3} = -(k_{-1} + k_2) \quad (7)$$

and the exponents $p_{i,j}^l$ characterizing the j-th variable in l-th monomial of the i-th equation reported in Table I.

By further defining vectors:

$$\alpha_1 = [\alpha_{1,1} \quad \alpha_{1,2} \quad \alpha_{1,3}]', \qquad \alpha_2 = [\alpha_{2,1} \quad \alpha_{2,2} \quad \alpha_{2,3}]' \quad (8)$$

and

$$X_1 = [X_{1,1} \quad X_{1,2} \quad X_{1,3}]' = [x_1 \quad x_1 x_2 \quad x_2]', \quad (9)$$
$$X_2 = [X_{2,1} \quad X_{2,2} \quad X_{2,3}]' = X_1,$$

then (6) is rewritten in the more compact form

$$\dot{x}_i = \alpha_i' X_i \qquad i = 1, 2 \quad (10)$$

According to [4], the 'driver' variables

$$Z_{i,l} = \frac{X_{i,l}}{x_i}, \qquad l = 1, \dots, \nu_i \quad (11)$$

are defined for $i = 1, 2$, so that (6) is further rewritten as

$$\dot{x}_i = (\alpha_i' Z_i) x_i \quad (12)$$

with

$$Z_1 = [1 \quad x_2 \quad x_1^{-1} x_2]', \qquad Z_2 = [x_1 x_2^{-1} \quad x_1 \quad 1]'. \quad (13)$$

By the exact quadratization theorem [4], the driver dynamics is given by:

$$\dot{Z}_{i,l} = \sum_{j=1}^{2} \pi_{i,j}^l (\alpha_j' Z_j) Z_{i,l} \qquad i = 1, 2, \quad l = 1, ..., \nu_i,$$

that is:

$$\begin{cases} \dot{Z}_{1,1} & = 0 \\ \dot{Z}_{1,2} & = \pi_{1,2}^2 (\alpha_2' Z_2) Z_{1,2} \\ \dot{Z}_{1,3} & = \pi_{1,1}^3 (\alpha_1' Z_1) Z_{1,3} + \pi_{1,2}^3 (\alpha_2' Z_2) Z_{1,3} \\ \dot{Z}_{2,1} & = \pi_{2,1}^1 (\alpha_1' Z_1) Z_{2,1} + \pi_{2,2}^1 (\alpha_2' Z_2) Z_{2,1} \\ \dot{Z}_{2,2} & = \pi_{2,1}^2 (\alpha_1' Z_1) Z_{2,2} \\ \dot{Z}_{2,3} & = 0 \end{cases} \quad (14)$$

with coefficients $\pi_{i,j}^l := p_{i,j}^l - \delta_{i,j}$ written in Table II, where $\delta_{i,j}$ is the Kronecker delta, providing 1 if $i = j$ and 0 elsewhere. By further expanding computations, we get:

$$\begin{cases} \dot{Z}_{1,1} & = 0 \\ \dot{Z}_{1,2} & = \alpha_{2,1} Z_{1,2} Z_{2,1} + \alpha_{2,2} Z_{1,2} Z_{2,2} + \alpha_{2,3} Z_{1,2} Z_{2,3} \\ \dot{Z}_{1,3} & = -\alpha_{1,1} Z_{1,1} Z_{1,3} - \alpha_{1,2} Z_{1,2} Z_{1,3} - \alpha_{1,3} Z_{1,3} Z_{1,3} \\ & \quad + \alpha_{2,1} Z_{1,3} Z_{2,1} + \alpha_{2,2} Z_{1,3} Z_{2,2} + \alpha_{2,3} Z_{1,3} Z_{2,3} \\ \dot{Z}_{2,1} & = \alpha_{1,1} Z_{1,1} Z_{2,1} + \alpha_{1,2} Z_{1,2} Z_{2,1} + \alpha_{1,3} Z_{1,3} Z_{2,1} \\ & \quad - \alpha_{2,1} Z_{2,1} Z_{2,1} - \alpha_{2,2} Z_{2,1} Z_{2,2} - \alpha_{2,3} Z_{2,1} Z_{2,3} \\ \dot{Z}_{2,2} & = \alpha_{1,1} Z_{1,1} Z_{2,2} + \alpha_{1,2} Z_{1,2} Z_{2,2} + \alpha_{1,3} Z_{1,3} Z_{2,2} \\ \dot{Z}_{2,3} & = 0 \end{cases}$$

$$(15)$$

It clearly appears that the driver equations in (15) include the original state variables[1], since $x_1 = Z_{2,2}$, $x_2 = Z_{1,2}$, with their related dynamics involving other driver variables. By defining an augmented state that includes $x_1, x_2, x_3 :=$

---

[1] It is possible to show that the set of times where $x_i = 0$, making (11) undefined, has zero measure, so that $x_1 = Z_{2,2}$ and $x_2 = Z_{1,2}$ do hold everywhere, provided that we extend by continuity the functions $Z_{2,2}, Z_{1,2}$, which are solution of the ODE (14), (15). For more details about this issue we refer readers to [5].

$Z_{1,3} = \frac{x_2}{x_1}$, $x_4 := Z_{2,1} = \frac{x_1}{x_2}$ and $x_5 := Z_{1,1} = Z_{2,3} = 1$, the original ODE system (5) is now embedded in the following extended system evolving in $\mathbb{R}^5$:

$$
\begin{cases}
\dot{x}_1 &= \alpha_{1,2}x_1x_2 + \alpha_{1,3}x_1x_3 + \alpha_{1,1}x_1x_5 \\
\dot{x}_2 &= \alpha_{2,2}x_2x_1 + \alpha_{2,1}x_2x_4 + \alpha_{2,3}x_2x_5 \\
\dot{x}_3 &= \alpha_{2,2}x_3x_1 - \alpha_{1,2}x_3x_2 - \alpha_{1,3}x_3^2 \\
&\quad + \alpha_{2,1}x_3x_4 + (\alpha_{2,3} - \alpha_{1,1})x_3x_5 \\
\dot{x}_4 &= -\alpha_{2,2}x_4x_1 + \alpha_{1,2}x_4x_2 + \alpha_{1,3}x_4x_3 \\
&\quad - \alpha_{2,1}x_4^2 + (\alpha_{1,1} - \alpha_{2,3})x_4x_5 \\
\dot{x}_5 &= 0
\end{cases}
\tag{16}
$$

or, in the more compact 'Driver-type' ODE form, see [5]:

$$
\dot{x}_i = \sum_{i=1}^{5} v_{i,j}x_ix_j = (v_i'x)x_i
\tag{17}
$$

with

$$
\begin{aligned}
v_1 &= \begin{bmatrix} 0 & \alpha_{1,2} & \alpha_{1,3} & 0 & \alpha_{1,1} \end{bmatrix}' \\
v_2 &= \begin{bmatrix} \alpha_{2,2} & 0 & 0 & \alpha_{2,1} & \alpha_{2,3} \end{bmatrix}' \\
v_3 &= \begin{bmatrix} \alpha_{2,2} & -\alpha_{1,2} & -\alpha_{1,3} & \alpha_{2,1} & (\alpha_{2,3} - \alpha_{1,1}) \end{bmatrix}' \\
v_4 &= \begin{bmatrix} -\alpha_{2,2} & \alpha_{1,2} & \alpha_{1,3} & -\alpha_{2,1} & (\alpha_{1,1} - \alpha_{2,3}) \end{bmatrix}' \\
v_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}'.
\end{aligned}
\tag{18}
$$

or, in terms of the original parameters,

$$
\begin{aligned}
v_1 &= \begin{bmatrix} 0 & k_1 & k_{-1} & 0 & -k_1 E_T \end{bmatrix}' \\
v_2 &= \begin{bmatrix} -k_1 & 0 & 0 & k_1 E_T & -(k_{-1} + k_2) \end{bmatrix}' \\
v_3 &= \begin{bmatrix} -k_1 & -k_1 & -k_{-1} & k_1 E_T & (k_1 E_T - k_{-1} - k_2) \end{bmatrix}' \\
v_4 &= \begin{bmatrix} k_1 & k_1 & k_{-1} & -k_1 E_T & -(k_1 E_T - k_{-1} - k_2) \end{bmatrix}' \\
v_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}'.
\end{aligned}
\tag{19}
$$

Remark 1: It is worth noting that the inclusion of a state variable with a trivial dynamics, $x_5 = 1$, is required to have a homogeneous second-order nonlinear transformation in the right-hand-side of (17). This fact will be exploited in the solution formula provided in the next section.

III. MAIN RESULTS

A. The solution formula

For a first order ODE system

$$
\dot{\chi} = f(\chi), \qquad \chi \in \mathbb{R}^n
\tag{20}
$$

let $t \mapsto \chi(t)$ be the solution of (20) with initial condition $\chi(t_0)$, and consider the Taylor expansion of a generic scalar component $\chi_i(\cdot)$ with respect to the initial time instant t_0:

$$
\chi_i(t) = \sum_{k=0}^{\infty} c_k(i)\frac{(t - t_0)^k}{k!}, \qquad c_k(i) = \chi_i^{(k)}(t_0).
\tag{21}
$$

Unfortunately, the series expansion cannot be straightforwardly applied to compute the solution or to build up a numerical integration algorithm, since it requires the derivatives of the solution. Below is reported a Theorem (see [5] for the proof and any details) that allows to compute the coefficients

of the Taylor expansion for a system in the 'Driver-type' ODE form (17).

Theorem 2: Consider a 'Driver-type' ODE in the form (17). Then the coefficients $c_k(i)$, $k > 0$, in the Taylor expansion (21) are given by

$$
c_k(i) = \sum_{i_1,\ldots,i_k \in \mathcal{S}} v_{i,i_1,\ldots,i_k}^{k+1} \cdot x_i(t_0)x_{i_1}(t_0) \cdots x_{i_k}(t_0),
\tag{22}
$$

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

$$
v_{i,i_1,\ldots,i_k}^{k+1} = v_{i,i_1,\ldots,i_{k-1}}^{k}\left(\sum_{j=0}^{k-1}v_{i_j,i_k}\right), \qquad v_i^1 = 1.
\tag{23}
$$

Coefficients $v_{i,j}$, $i,j = 1,\ldots,5$, in (23) refer to the j-th entry of v_i in (18)-(19).

We below report the explicit computation of the coefficients provided by Theorem 2 for the cases $k = 1,2$, in addition to the trivial case $k = 0$, provided by

$$
c_0(i) = x_i^{(0)}(t_0) = x_i(t_0), \qquad \forall i \in \mathcal{S} = \{1,\ldots,5\}
\tag{24}
$$

For $k = 1$:

$$
\begin{aligned}
v_{i,i_1}^2 &= v_i^1 v_{i,i_1} = v_{i,i_1} \qquad i,i_1 \in \mathcal{S} \\
c_1(i) &= x_i^{(1)}(t_0) = \sum_{i_1 \in \mathcal{S}} v_{i,i_1}^2 \cdot x_i(t_0)x_{i_1}(t_0) \\
&= \sum_{i_1 \in \mathcal{S}} v_{i,i_1} \cdot x_i(t_0)x_{i_1}(t_0) = \left(v_i'x(t_0)\right)x_i(t_0) \qquad i \in \mathcal{S}
\end{aligned}
$$

For $k = 2$:

$$
\begin{aligned}
v_{i,i_1,i_2}^3 &= v_{i,i_1}^2\left(v_{i,i_2} + v_{i_1,i_2}\right) \\
&= v_{i,i_1}\left(v_{i,i_2} + v_{i_1,i_2}\right)
\end{aligned}
$$

$$
\begin{aligned}
c_2(i) &= x_i^{(2)}(t_0) = \sum_{i_1,i_2 \in \mathcal{S}} v_{i,i_1,i_2}^3 \cdot x_i(t_0)x_{i_1}(t_0)x_{i_2}(t_0) \\
&= \sum_{i_1,i_2 \in \mathcal{S}} v_{i,i_1}\left(v_{i,i_2} + v_{i_1,i_2}\right) \cdot x_i(t_0)x_{i_1}(t_0)x_{i_2}(t_0)
\end{aligned}
$$

B. The numerical integration algorithm

A fixed step numerical integration algorithm is here provided, based on Theorem 2 results. In the following t_j, $j = 0,1,\ldots$ denote the time steps, with $t_{j+1} - t_j = \Delta$, for any $j = 0,1,\ldots$, whilst $\hat{x}_i^k(t_j)$ denote the estimate of x_i, $i \in \mathcal{S}$, at step t_j, according to a given order k of the series expansion. The numerical integration algorithm is below reported. To this end, coefficients $v_{i,i_1,\ldots,i_k}^{k+1}$ are computed off-line, so they do not affect substantially the computational burden.

0) t_j and $\hat{x}_i^k(t_j)$ are given;
1) compute coefficients $c_k(i)$ in t_j from (22);
2) update the state estimate $\hat{x}_i^k(t_{j+1})$ in t_{j+1} according to the series expansion in (21) in the neighborhood of t_j, truncated at k-th order and computed in $t = t_{j+1} = t_j + \Delta$;
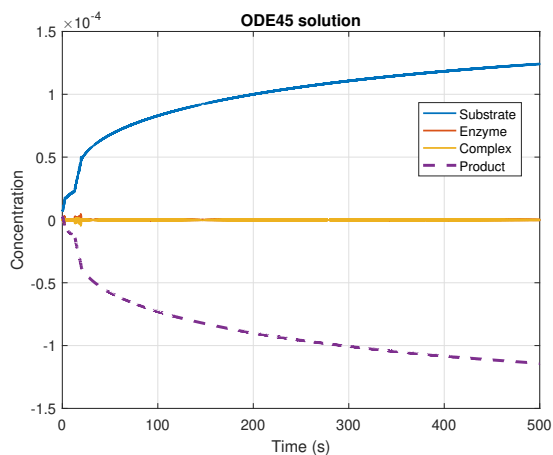
Fig. 1. Solution of system (4) via Matlab® ODE45. Parameters are taken from [13]. Note that the product concentration p (purple dashed line) is negative throughout the whole simulation time.

Fig. 2. Solution of system (4) via Matlab® ODE15s. Parameters are taken from [13].

Remark 3: For $k = 1$, it readily comes that the proposed numerical integration algorithm coincides with the Euler algorithm. This fact is actually true for any nonlinear system (not only for the enzymatic reaction scheme here investigated).

IV. SIMULATION RESULTS

Numerical simulations of the system (4) have been performed in the Matlab® suite. In the first set of simulations, we consider the parameters taken from [13]:

$$k_1 = 4 \cdot 10^6 \mathrm{M}^{-1}\mathrm{s}^{-1}, \qquad k_{-1} = 25\mathrm{s}^{-1}, \qquad k_2 = 15\mathrm{s}^{-1}, \tag{25}$$

with total amounts (3)

$$S_T = \frac{k_{-1} + k_2}{k_1}, \qquad E_T = 10^{-3} \cdot S_T. \tag{26}$$

We start from initial conditions

$$s(0) = 0.75 S_T, \qquad c(0) = 0.25 E_T. \tag{27}$$

Fig. 1 shows the numerical solution obtained by means of the standard ODE45 Matlab® solver, based on the Dormand-Prince Runge-Kutta method [14]. It is readily seen from the plot of concentrations that, because of the stiffness of the ODE system (4), the method fails since the product level is negative during the whole time horizon.

Another off-the-shelf method is the ODE15s Matlab® variable-order solver for stiff differential equations [15], which provides meaningful results (see Fig. 2) for the same parameter set. Alternatively, a solution can be produced by means of the method described in this paper. We choose the truncation order $k = 3$ in the series expansion (21) and the sampling time $\Delta = 0.035$s in the algorithm provided in Section III.B. The solution in Fig. 3 looks indistinguishable from the ODE15s solution. For the same time step Δ, the solutions for $k = 1$ (Euler method) and $k = 2$ are unstable (the concentrations diverge).
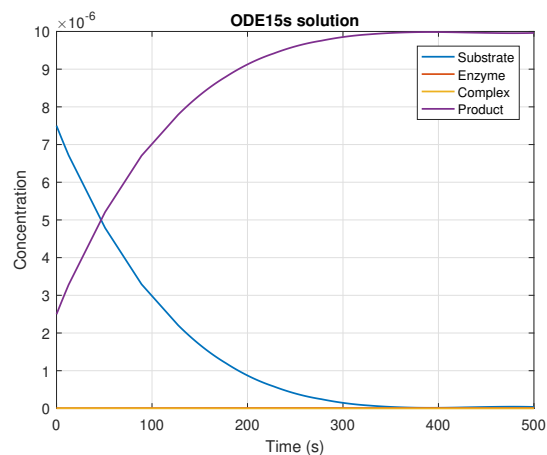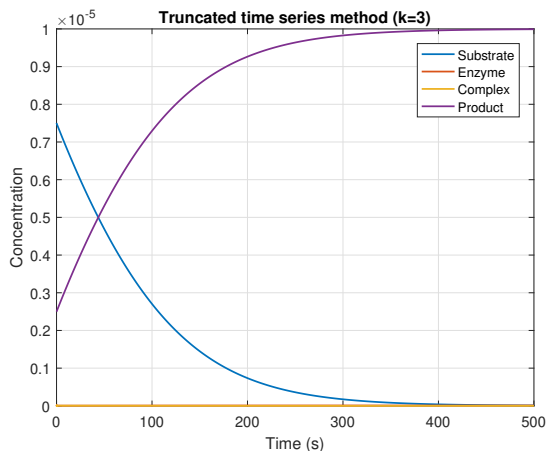

Fig. 3. Solution of system (4) via the truncated time series expansion described in this paper. Parameters are taken from [13]. The truncation order has been set to $k = 3$.

In the second set of simulations, we increase k_1 to the value $k_1 = 4 \cdot 10^7 \mathrm{M}^{-1}\mathrm{s}^{-1}$. This fact worsens the stiffness of the ODE. While, quite predictably, the Matlab ODE45 solver confirms the issues described above for the first simulation, it is quite surprising that also the stiff solver ODE15s exhibits an inconsistent behavior, with the substrate concentration getting negative (see Fig. 4). Instead, the solution by means of the time series method, truncated at order $k = 2$ and with $\Delta = 0.005$s, still provides a meaningful solution (in Fig. 5).

Finally, we further increase k_1 to the value $k_1 = 10^8 \mathrm{M}^{-1}\mathrm{s}^{-1}$. In this case ODE45 simulations with default settings are aborted by Matlab for not being able to meet integration tolerances. The ODE15s stiff solver still exhibits an inconsistent behavior with a substrate trajectory crossing zero, as shown in Fig. 6. Instead, a simulation with our method for $k = 2$ and $\Delta = 0.0025$s provides consistent results (see Fig. 7).

Summarizing the results illustrated above, preliminary simulations seem to show that the time series method can
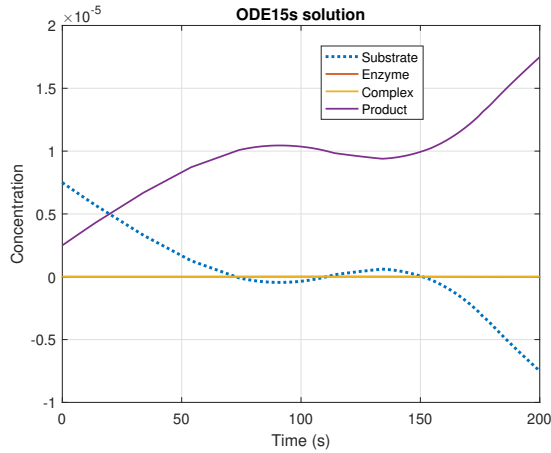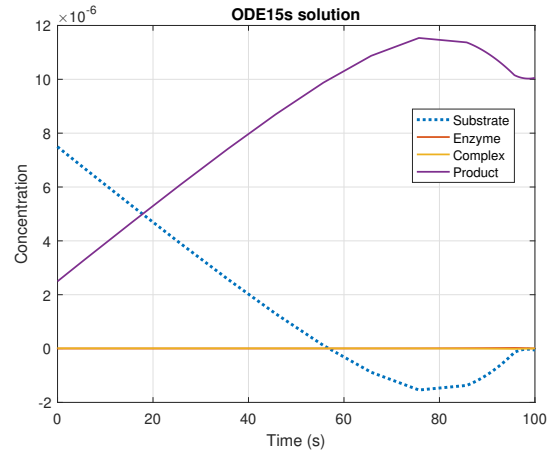
Fig. 4. Solution of system (4) via Matlab® ODE15s, with parameters taken from [13] except for k_1, which is increased to the value $4 \cdot 10^7 \mathrm{M}^{-1}\mathrm{s}^{-1}$. Note that the substrate concentration s (cyan dotted line) becomes negative, thus meaningless.

Fig. 6. Solution of system (4) via Matlab® ODE15s, with parameters taken from [13] except for k_1, which is increased to the value $10^8 \mathrm{M}^{-1}\mathrm{s}^{-1}$. Note that the substrate concentration s (cyan dotted line) becomes negative, thus meaningless.
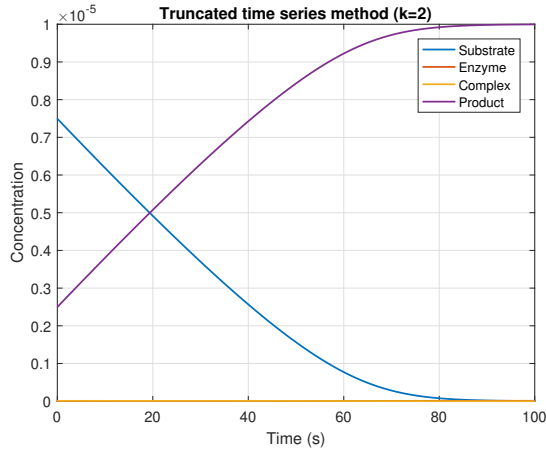

Fig. 5. Solution of system (4) via the truncated time series expansion described in this paper, with parameters taken from [13] except for k_1, which is increased to the value $4 \cdot 10^7 \mathrm{M}^{-1}\mathrm{s}^{-1}$. The truncation order has been set to $k = 2$.
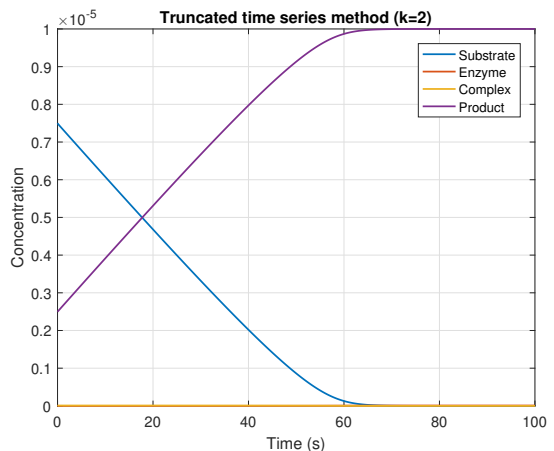

Fig. 7. Solution of system (4) via the truncated time series expansion described in this paper, with parameters taken from [13] except for k_1, which is increased to the value $10^8 \mathrm{M}^{-1}\mathrm{s}^{-1}$. The truncation order has been set to $k = 2$.

be useful to provide meaningful integration of systems for which standard numerical methods may fail. This is obtained with a limited computational burden for low truncation orders, since our simulations lasted very few seconds. It is worth noting that the parameter setting has been taken from the experimental literature [13], thus motivating the use of the proposed methodology in real frameworks.

V. CONCLUSIONS

In this work, we proposed a novel solution to the problem of approximating the solution of well-known enzymatic reaction systems exhibiting an apparent double time-scale separation. While such a kind of systems poses some integrability issues by means of classical numerical solvers, the illustrated scheme is based on a time-series-expansion approach, where the exact solution formula needs to be truncated for practical computability. Although the complexity of the approach

grows exponentially with the chosen truncation order, it is shown by preliminary simulations that a relatively low order offers a compromise between accuracy and efficiency. Since some problems can be hardly solvable with low-order truncation methods, future work will be devoted to problem-dependent adaptations of the general procedure given in this work, in order to overcome the curse of dimensionality.

REFERENCES

[1] A.M. Bersani, E. Bersani, G. Dell'Acqua, M.G. Pedersen, New trends and perspectives in nonlinear intracellular dynamics: one century from Michaelis–Menten paper, Continuum Mech. Termodyn., 27, 659–684, 2015.

[2] Borghans, J. and de Boer, R. and Segel, L.: Extending the quasi-steady state approximation by changing variables, Bull. Math. Biol., 58, 43–63, 1996.

[3] Briggs, G. E., Haldane, J. B. S.: A note on the kinetics of enzyme action, Biochem. J., 19, 338–339, 1925.

[4] F. Carravetta: Global Exact Quadratization of Continuous-Time Non-linear Control Systems. *SIAM J. Control Optim.* 53, 235-261, 2015.

[5] F. Carravetta: On the Solution Calculation of Nonlinear Ordinary Differential Equations via Exact Quadratization. *arXive:1903.06073*, 2019, http://arxiv.org/abs/1903.06073

[6] A. Cornish-Bowden, One hundred years of Michaelis-Menten kinetics, *Perspectives in Science*, 4, 3–9, 2015.

[7] R. Chang, Physical chemistry for the chemical and biological sciences, *University Science Books*, 2000.

[8] G. Cooper, The Cell: A molecular approach, *USA: Sinauer Associates*, 2000.

[9] D. T. Gillespie, Exact Stochastic Simulation of Coupled Chemical Reactions, *The Journal of Physical Chemistry* 81(25), 2340–2361, 1977.

[10] D. T. Gillespie, Approximate Accelerated Stochastic Simulation of Chemically Reacting Systems, *The Journal of Chemical Physics* 115(4), 1716–1733, 2001.

[11] O. Kirk, T.V. Borchert, C.C. Fuglsang, Industrial enzyme applications, *Curr. Opin. Biotechnol.*, 13, 345–351, 2002.

[12] Michaelis L., Menten M., Kinetics of invertase action, *Biochem. Z*, 49, 333–369, 1913.

[13] Segel, L.: On the validity of the steady state assumption of enzyme kinetics. Bull. Math. Biol. 50, 579–593 (1988)

[14] J.R. Dormand, P.J. Prince, A family of embedded Runge-Kutta formulae, *Journal of computational and applied mathematics*, 6(1), 19-26, 1980.

[15] L.F. Shampine, M. W. Reichelt, The MATLAB ODE Suite, *SIAM Journal on Scientific Computing*, Vol. 18, pp. 1–22, 1997.

[16] N.G. van Kampen, *Stochastic Processes in Physics and Chemistry*, North Holland, third edition, 2007.