Commun Nonlinear Sci Numer Simulat 131 (2024) 107844

Contents lists available at ScienceDirect

Communications in Nonlinear Science and Numerical Simulation

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

Research paper

Automatic Proper Orthogonal Block Decomposition method for network dynamical systems with multiple timescales

A. Bandera [a,b,*], S. Fernández-García [a,b], M. Gómez-Mármol [a], A. Vidal [c]

[a] *Ecuaciones Diferenciales y Análisis Numérico, Universidad de Sevilla, Calle Tarfia s/n, Seville, 41012, Spain*
[b] *IMUS, Universidad de Sevilla, Calle Tarfia s/n, Seville, 41012, Spain*
[c] *Laboratoire de Mathématiques et Modélisation d'Évry (LAMME), Univ Evry, CNRS, Université Paris-Saclay, IBGBI, 23 Bld de France, Evry, 91037, France*

ARTICLE INFO

Keywords:
Slow–fast dynamics
Coupled oscillators
Synchronization
Network neuron model
Reduced order models
Mixed-mode oscillations

ABSTRACT

In this work, we introduce a novel reduced order model technique, based on the Proper Orthogonal Decomposition method, for dynamical systems with multiple timescales. The main ideas are to retain the structure of the original model, which is lost in the original POD procedure, while producing a competitive reduction in the number of equations and computational time, and to determine the best structure for the reduced system automatically, via a data-driven analysis of the original model data. For these novel techniques, we present some numerical tests for various behaviors of three different neural network models with multiple timescales, which support the use of these new methods.

1. Introduction

Extensive research over the past three decades have been revolutionary by conceptualizing, analyzing and simulating network models of neuronal activity. Concerning the sole formalism of dynamical systems coupled as networks to represent interactions between assemblies of neurons, various paradigms and scales of modeling have been introduced depending on the questioning to be tackled, such as neural mass [1–3], neural fields [4,5], mean-fields [6] and dendritic trees [7], among others. However, common and still unsolved challenges arise in these approaches, especially the difficulty to classify and analyze the features of cellular activity synchronization, and the cost of reliable simulations for realistically sized networks.

The first challenge mentioned above mainly implies results from the different timescales involved in neuronal activities, which are naturally translated into the models by the use of multiple timescale systems (for reproducing the individual cellular activities) coupled as a network. Synchronization mechanisms of multiple timescale systems are quite specific and different from synchronization of harmonic oscillators (especially the impact of the coupling types and strengths on the synchronization). Nevertheless, it is well-known that synchronization of neuron assemblies plays a key role in the neuronal information treatment and diffusion, and overall in the neural functioning. Several works (for instance, [8–17]) have addressed the question of analyzing synchronization mechanisms in the presence of canard phenomena and bursting oscillations, symptomatic of multiple timescale dynamics, in coupled oscillators. But the tremendously large panel of synchronization types and associated conditions remains to be analyzed, especially for large networks, as well as its physio-pathological interpretation.

With this aim, efficient numerical techniques for producing reliable simulations are key tools and need to be thoroughly tested, due to the difficulty to catch structuring dynamical mechanisms specific to multiple timescale systems. As mentioned as a second

[*] Corresponding author at: IMUS, Universidad de Sevilla, Calle Tarfia s/n, Seville, 41012, Spain.
E-mail addresses: abandera@us.es (A. Bandera), soledad@us.es (S. Fernández-García), macarena@us.es (M. Gómez-Mármol), alexandre.vidal@univ-evry.fr (A. Vidal).

https://doi.org/10.1016/j.cnsns.2024.107844
Received 1 September 2023; Received in revised form 22 December 2023; Accepted 8 January 2024
Available online 12 January 2024

challenge above, simulation of network models might be computationally expensive when the number of cells is large. Therefore, intrusive Reduced Order Model (ROM) approximations have been developed [18–28]. For a state-of-the-art summary, we recommend Refs. [29,30]. These techniques aim to obtain an equivalent dynamical system of lower dimension that is a good approximation of the original Full Order Model (FOM), namely, the reduced problem.

Despite the ability of these intrusive methods to approximate closely the solution generated by the original system [31], performing these techniques may make us lose the original structure of the problem, such as, for instance, the slow–fast separation of variables typical in neuroscience problems. This is due to the fact that the usual procedures to obtain the ROM do not consider any separation of variables, nor any given model structure. Therefore, we could be mixing fast and slow behaviors in the reduced model, which is not very natural. This problematic could also impede the direct study of the behaviors in the reduced framework. In order to preserve the structure of the problem, different strategies have been proposed in the literature. For instance, a technique based on Krylov subspaces was developed in the case of electronic circuits [19–21], and a technique to decompose the domain in computational simulations of fluids was introduced in [32]. In this paper, we present a different approach based, on the one hand, on the Proper Orthogonal Decomposition (POD) [18,22–27] and, on the other hand, on a new data-driven procedure based on a collinearity study to determine automatically the best structure for the reduced model, while obtaining a suitable reduction in computational time.

The article outline is the following. In Section 2, we first summarize the POD method. We remark some properties, and we identify the issue of the structure loss. Then, we introduce in Section 2.1 a block approach to preserve the structure of the original problem after the reduction. We also adapt the properties of the POD method to this new block case. In Section 2.2, we present a data-driven procedure to determine the best block structure for the reduced model. After that, in Section 3 we test and compare our methods for three neural network models with multiple timescales. In particular, Section 3.1 considers the case of a coupled network model of the Intracellular Calcium Concentration system, studied previously in [31]; Section 3.2 focuses on the study of a coupled network of Hindmarsh–Rose models [33–36] and Section 3.3 is devoted to the study of a coupled network model of Fold/Hopf bursters, found in pancreatic β-cells, among others [37–40]. We show the reduction of CPU time needed for the numerical integration of the reduced system compared to the original one, the sparsity indexes of the models as a measure of the preservation of the structure and an error comparison for the reduced models. Finally, Section 4 is devoted to present conclusions and perspectives of the present work. Appendix is focused on analyzing the limits of these ROM techniques in some test cases.

2. Reduced order methods based on the proper orthogonal decomposition

In this section, we begin by introducing the POD method. First, in order to build the ROM problem, we need to identify the number of equations. To do that, we can apply the POD method, one of the most popular approaches for reduced problem formulation [18,22,24–27]. Then, once we have the dimension of the reduced problem, the POD method also provides a basis for it, so we can formulate a ROM problem equivalent to the FOM problem.

The POD method is based on the Singular Value Decomposition technique (SVD). It is known that the following property holds for the SVD of a matrix, $\mathbb{S} = (\boldsymbol{w}_1 | \dots | \boldsymbol{w}_{n_s})$, see [41]:

Proposition 2.1. *Let $\mathcal{V}_{N_r} = \{\mathbb{W} \in \mathbb{R}^{N \times N_r} \ : \ \mathbb{W}^T \mathbb{W} = \mathbb{I}_{N_r}\}$ be the set of all orthonormal bases of N_r-dimensional subspaces of $\mathbb{R}^N$. Then,*

$$\sum_{i=1}^{n_s} \|\boldsymbol{w}_i - \mathbb{U}\mathbb{U}^T \boldsymbol{w}_i\|_2^2 = \min_{\mathbb{W}\in\mathcal{V}_{N_r}} \sum_{i=1}^{n_s} \|\boldsymbol{w}_i - \mathbb{W}\mathbb{W}^T \boldsymbol{w}_i\|_2^2 = \sum_{i=N_r+1}^{r} \sigma_i^2,$$

where $\mathbb{U} \in \mathbb{R}^{N\times N_r}$ arises from the SVD of $\mathbb{S}$ and is a full matrix containing the first N_r left singular vectors in columns corresponding to the N_r greatest singular values and r is the number of non-zero singular values.

Therefore, the SVD provides a basis that minimizes the sum of the square of the errors between each snapshot vector $\boldsymbol{w}_i$ and its projection onto the reduced subspace, and the error is equal to the sum of the squares of the neglected singular values.

From now on, we focus on POD applied to dynamical systems. For a general parametric dynamical system $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{\gamma})$, with $\boldsymbol{\gamma}$ the parameter vector, the POD method can be summarized as follows:

1. To solve the original problem for a certain partition of the parameter space, where the solution is qualitatively similar, and store the solutions in time in a matrix $\mathbb{S}$, the so-called snapshot matrix.
2. To compute the SVD of the snapshot matrix $\mathbb{S}$, and sort, in descending order, the singular values and its corresponding left singular vectors.
3. To apply Proposition 2.1 as a measure of the information lost to select the least amount of basis functions that retain the most information, according to a certain fixed tolerance. This will provide a submatrix of $\mathbb{U}$, the matrix of left singular vectors, that contains the basis functions, which we also denote by $\mathbb{U}$.
4. To construct the reduced problem as $\dot{\boldsymbol{z}} = \mathbb{U}^T f(\mathbb{U}\boldsymbol{z}, t)$, where $\boldsymbol{z} = \mathbb{U}^T \boldsymbol{x}$. To recover the solution in the original space, just a left-multiplication by $\mathbb{U}$ of the reduced solution is needed.

However, as said in the introduction, performing this technique may make us to lose the original structure of the problem. This is due to the fact that in the original problem we work with quasi-sparse matrices and vectors, but in the reduction this quasi-sparsity is lost, and we have full matrices and vectors, as the corresponding basis matrix $\mathbb{U}$ is full. Let us provide an illustrative example.

In the following, in order to illustrate the main features of the methods proposed in a clear way, we will consider a dynamical system of $N \in \mathbb{N}$ equations, that could be separated into two groups. However, this analysis can be generalized to $M < N$ groups. Without loss of generality, we can consider that the variables can be sorted by group. A dynamical system in this framework could read,

$$\begin{pmatrix} \dot{\boldsymbol{x}}_1 \\ \dot{\boldsymbol{x}}_2 \end{pmatrix} = \begin{pmatrix} \mathbb{A}_{11} & \mathbb{A}_{12} \\ \mathbb{A}_{21} & \mathbb{A}_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} + \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}, \tag{1}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{b}_i$, $i = 1, 2$, are vectors of dimensions N_1 and N_2, respectively, such that $N = N_1 + N_2$ and $\mathbb{A}_{ij}$, $i, j = 1, 2$, are real matrices with consistent dimensions. For the sake of clarity, we will consider $\mathbb{A}_{11}$ and $\mathbb{A}_{22}$ as the identity matrices, $\mathbb{A}_{21} = 0$ and $\mathbb{A}_{12}$ a full matrix, so

$$\begin{pmatrix} \mathbb{A}_{11} & \mathbb{A}_{12} \\ \mathbb{A}_{21} & \mathbb{A}_{22} \end{pmatrix} = \begin{pmatrix} \mathbb{I} & \mathbb{A}_{12} \\ \mathbb{O} & \mathbb{I} \end{pmatrix}.$$

If we consider system (1) and we let $\mathbb{U}$ be the matrix arising from the POD of some snapshots of the system, then the reduced order system reads,

$$\dot{\boldsymbol{z}} = \tilde{\mathbb{A}} \boldsymbol{z} + \boldsymbol{b}_{\boldsymbol{r}}, \tag{2}$$

where

$$\tilde{\mathbb{A}} = \begin{pmatrix} \mathbb{I} + \mathbb{U}_{11}^T \mathbb{A}_{11} \mathbb{U}_{21} & \mathbb{U}_{11}^T \mathbb{A}_{12} \mathbb{U}_{22} \\ \mathbb{U}_{12}^T \mathbb{A}_{12} \mathbb{U}_{21} & \mathbb{I} + \mathbb{U}_{12}^T \mathbb{A}_{12} \mathbb{U}_{22} \end{pmatrix}, \quad \text{with} \quad \mathbb{U} = \begin{pmatrix} \mathbb{U}_{11} & \mathbb{U}_{12} \\ \mathbb{U}_{21} & \mathbb{U}_{22} \end{pmatrix}, \quad \text{and} \quad \boldsymbol{b}_{\boldsymbol{r}} = \mathbb{U}^T \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix},$$

and we have lost the structural information of the original problem and the possible separation between variables. To solve this issue, we present the Proper Orthogonal Block Decomposition (POBD) method subsequently.

2.1. Proper orthogonal block decomposition

In order to maintain the main properties of the FOM system when we perform the reduction, we have to take care of the structure of the matrix $\mathbb{U}$. Taking into account the separation of variables in the FOM system (1), we aim to build a reduced basis for each set of variables.

With that in mind, we divide the snapshot matrix $\mathbb{S} = (\boldsymbol{w}_1 | \dots | \boldsymbol{w}_{ns})$ into two blocks as follows $\mathbb{S}^T = \left(\mathbb{S}_1^T | \mathbb{S}_2^T\right)$. Now, we perform a SVD for each block, and we obtain a POD basis for each block,

$$\mathbb{S}_1 \approx \sum_{i=1}^{r_1} \sigma_i^{(1)} \boldsymbol{u}_i^{(1)} (\boldsymbol{v}_i^{(1)})^T, \quad \mathbb{S}_2 \approx \sum_{i=1}^{r_2} \sigma_i^{(2)} \boldsymbol{u}_i^{(2)} (\boldsymbol{v}_i^{(2)})^T,$$

where $\boldsymbol{u}_i^{(\cdot)}$ is the ith left singular vector of the corresponding matrix. Proposition 2.1 holds for each block, so as in the original POD method, we have a suitable criterion for the selection of the reduced basis dimension for each variable. We name the reduced dimensions N_{r1}, N_{r2}.

Now, we focus on the reconstruction of the original snapshot matrix as a whole. We note that each block matrix has the correct number of columns, so no change in the block right singular vectors $\boldsymbol{v}_i^{(\cdot)}$ is needed. However, each block left singular vector $\boldsymbol{u}_i^{(1)}$ lays in $\mathbb{R}^{N_1}$ so we need to complete them in order to have a vector in $\mathbb{R}^N$ as we must, and respectively, for vectors $\boldsymbol{u}_i^{(2)}$.

A direct concatenation of the block left singular vectors does not work, as we do not know how the product $\boldsymbol{u}_i^{(k)} (\boldsymbol{v}_i^{(l)})^T$ behaves, and we have no control over it. A successful approach is to add as many zeros as needed to each block left singular vector, taking into account the position of the block matrix in the original matrix, i.e.,

$$\boldsymbol{u}_i^{(1)} \mapsto \begin{pmatrix} \boldsymbol{u}_i^{(1)} \\ \boldsymbol{0}_{N_2} \end{pmatrix}, \quad \boldsymbol{u}_i^{(2)} \mapsto \begin{pmatrix} \boldsymbol{0}_{N_1} \\ \boldsymbol{u}_i^{(2)} \end{pmatrix}.$$

With this decomposition, we can recover the original matrix as

$$\mathbb{S} \approx \begin{pmatrix} \mathbb{U}_1 & \mathbb{O}_{N_1, N_{r2}} \\ \mathbb{O}_{N_2, N_{r1}} & \mathbb{U}_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & \mathbb{O}_{N_1, N_{r2}} \\ \mathbb{O}_{N_2, N_{r1}} & \Sigma_2 \end{pmatrix} \begin{pmatrix} \mathbb{V}_1^T \\ \mathbb{V}_2^T \end{pmatrix} = \begin{pmatrix} \mathbb{S}_1 \\ \mathbb{S}_2 \end{pmatrix}.$$

If we select the matrix

$$\mathbb{U} = \begin{pmatrix} \mathbb{U}_1 & \mathbb{O}_{N_1, N_{r2}} \\ \mathbb{O}_{N_2, N_{r1}} & \mathbb{U}_2 \end{pmatrix}, \tag{3}$$

as the new basis for the reduced subspace, we are able to maintain the properties of the FOM system (1), as follows:

$$\begin{aligned} \tilde{\mathbb{A}} &= \mathbb{U}^T \mathbb{A} \mathbb{U} = \begin{pmatrix} \mathbb{U}_1^T & \mathbb{O}_{N_2, N_{r1}}^T \\ \mathbb{O}_{N_1, N_{r2}}^T & \mathbb{U}_2^T \end{pmatrix} \begin{pmatrix} \mathbb{I}_{N_1} & \mathbb{A}_{12} \\ \mathbb{O}_{N_2, N_1} & \mathbb{I}_{N_2} \end{pmatrix} \begin{pmatrix} \mathbb{U}_1 & \mathbb{O}_{N_1, N_{r2}} \\ \mathbb{O}_{N_2, N_{r1}} & \mathbb{U}_2 \end{pmatrix} \\ &= \begin{pmatrix} \mathbb{I}_{N_{r1}} & \mathbb{U}_1^T \mathbb{A}_{12} \mathbb{U}_2 \\ \mathbb{O}_{N_{r2}, N_{r1}} & \mathbb{I}_{N_{r2}} \end{pmatrix} = \begin{pmatrix} \mathbb{I}_{N_{r1}} & \tilde{\mathbb{A}}_{12} \\ \mathbb{O}_{N_{r2}, N_{r1}} & \mathbb{I}_{N_{r2}} \end{pmatrix}. \end{aligned}$$

Table 1
Comparison of the sparsity index for between the FOM, POD and POBD systems.

	FOM
Non-zero elements	$\approx N_1 + N_2 + N_1 N_2$
Total elements	$(N_1 + N_2)^2$
Sparsity index	$\approx \dfrac{N_1(N_1-1)+N_2(N_2-1)+N_1N_2}{(N_1+N_2)^2}$

	POD	POBD
Non-zero elements	$\approx N_r^2$	$\approx N_{r1} + N_{r2} + N_{r1} N_{r2}$
Total elements	N_r^2	$(N_{r1} + N_{r2})^2$
Sparsity index	≈ 0	$\approx \dfrac{N_{r1}(N_{r1}-1)+N_{r2}(N_{r2}-1)+N_{r1}N_{r2}}{(N_{r1}+N_{r2})^2}$

We remark that we keep the original structure for our reduced system. We name this method the *Proper Orthogonal Block Decomposition* (POBD).

In order to measure the good behavior of this method, taking into account just the number of equations of the reduced system could be not enough, as we could also need to retain some information about the structure of the original matrix. Bearing that in mind, we propose a new index, that we name the sparsity index, defined as the quotient between the number of zero entries and the number of total elements of the matrix of a system of differential equations. Table 1 shows the comparison between the sparsity index of the FOM, POD and POBD systems for the academic test case (1). We can easily check that, for this test case, the sparsity index of the POBD systems is much more in concordance with the one of the FOM system, while the sparsity index of the POD system is close to zero.

For the original POD method, we are able to obtain an error bound depending on the values of the neglected eigenvalues, see Proposition 2.1. In our case, we are able to obtain a similar error bound applying similar techniques.

Proposition 2.2. *Let $\mathbb{U}$ be an orthonormal matrix built as in (3) for a dynamical system with $N_b \geq 2$ blocks. Then, the next error bound holds*

$$\sum_{i=1}^{n_s} \|\boldsymbol{w}_i - \mathbb{U}^T\mathbb{U}\boldsymbol{w}_i\|_2^2 = \sum_{j=1}^{N_b} \sum_{i=N_{r_j}+1}^{r_j} (\sigma_i^{(j)})^2.$$

Proof. This proof is an adaptation of the proof of Proposition 2.1 in [41] to our case. We detail it for $N_b = 2$.

For a matrix, $\mathbb{A} \in \mathbb{R}^{N\times n_s}$ we have $\|\mathbb{A}\|_F^2 = \sum_{i=1}^{n_s} \|\boldsymbol{a}_i\|_2^2$, where $\boldsymbol{a}_i$ denotes the ith column of $\mathbb{A}$. Thus, in our case we can write

$$\sum_{i=1}^{n_s} \|\boldsymbol{w}_i - \mathbb{U}\mathbb{U}^T\boldsymbol{w}_i\|_2^2 = \|\mathbb{S} - \mathbb{U}\mathbb{U}^T\mathbb{S}\|_F^2. \tag{4}$$

Now, as

$$\mathbb{S} = \begin{pmatrix} \mathbb{S}_1 \\ \mathbb{S}_2 \end{pmatrix} = \begin{pmatrix} \mathbb{S}_1 \\ \mathbb{O}_{N_2,n_s} \end{pmatrix} + \begin{pmatrix} \mathbb{O}_{N_1,n_s} \\ \mathbb{S}_2 \end{pmatrix},$$

and

$$\mathbb{U}\mathbb{U}^T\mathbb{S} = \begin{pmatrix} \mathbb{U}_1\mathbb{U}_1^T\mathbb{S}_1 \\ \mathbb{U}_2\mathbb{U}_2^T\mathbb{S}_2 \end{pmatrix} = \begin{pmatrix} \mathbb{U}_1\mathbb{U}_1^T\mathbb{S}_1 \\ \mathbb{O}_{N_2,n_s} \end{pmatrix} + \begin{pmatrix} \mathbb{O}_{N_1,n_s} \\ \mathbb{U}_2\mathbb{U}_2^T\mathbb{S}_2 \end{pmatrix},$$

along with the facts that $\|S\|_F = \|S^T\|_F$ and $\|(A|B)\|_F^2 = \|A\|_F^2 + \|B\|_F^2$, we can decompose the last term of (4) as

$$\|\mathbb{S} - \mathbb{U}\mathbb{U}^T\mathbb{S}\|_F^2 = \|\mathbb{S}_1 - \mathbb{U}_1\mathbb{U}_1^T\mathbb{S}_1\|_F^2 + \|\mathbb{S}_2 - \mathbb{U}_2\mathbb{U}_2^T\mathbb{S}_2\|_F^2. \tag{5}$$

Finally, applying Proposition 2.1 to each term, we obtain the desired result.

We can remark that this procedure is suitable for any number of blocks and that when considering just 1 block, we recover the classic POD method.

Remark 2.1. We remark that despite the similarities between the error bounds of Propositions 2.1 and 2.2, for a number of blocks $N_b \geq 2$ the latter might not be the optimal for reducing onto a subspace of dimension $\sum_{j=1}^{N_b} N_{r_j}$.

2.2. Automatic proper orthogonal block decomposition

In problems with N_b large, there could be blocks of variables that could be studied as a whole block, allowing us to improve the error estimation presented in Proposition 2.2, as we would need to deal with a lesser number of blocks. In this section, we propose a procedure to identify those blocks that could be merged into a larger block by means of a collinearity analysis of the snapshots. We would name this method *Automatic Proper Orthogonal Block Decomposition* (APOBD). The procedure of the selection of blocks is summarized in the following. We have illustrated the steps with the academic example of the previous Eq. (1),

1. The first step is to compute the correlation matrix of the snapshots $\mathbb{C} \in \mathbb{R}^{N\times N}$. This could also be seen as the first step for the POD, as the singular values of a matrix are the square root of the eigenvalues of the respective correlation matrix,
$$\mathbb{C} = \mathbb{S}\mathbb{S}^T = \begin{pmatrix} \mathbb{S}_1 \\ \mathbb{S}_2 \end{pmatrix} \begin{pmatrix} \mathbb{S}_1^T & \big| & \mathbb{S}_2^T \end{pmatrix} = \begin{pmatrix} \mathbb{S}_1\mathbb{S}_1^T & \mathbb{S}_1\mathbb{S}_2^T \\ \mathbb{S}_2\mathbb{S}_1^T & \mathbb{S}_2\mathbb{S}_2^T \end{pmatrix} = \begin{pmatrix} \mathbb{C}_{11} & \mathbb{C}_{12} \\ \mathbb{C}_{12}^T & \mathbb{C}_{22} \end{pmatrix}. \tag{6}$$
2. Then, we compute the mean correlation of each block with itself and the others, so we end up with a symmetric matrix of dimension equal to the number of blocks. In order to check if the a priori blocks were well located, we can check that the elements of the diagonal of this matrix are close to 1.
$$\mathbb{L} = \begin{pmatrix} \dfrac{\sum_{i,j=1}^{N_1}(\mathbb{C}_{11})_{ij}}{N_1^2} & \dfrac{\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(\mathbb{C}_{12})_{ij}}{N_1N_2} \\ \dfrac{\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(\mathbb{C}_{12})_{ij}}{N_1N_2} & \dfrac{\sum_{i,j=1}^{N_2}(\mathbb{C}_{22})_{ij}}{N_2^2} \end{pmatrix} = \begin{pmatrix} l_{11} & l_{12} \\ l_{12} & l_{22} \end{pmatrix}, \quad \text{with} \quad l_{ii} \approx 1. \tag{7}$$
3. As the third step, for each pair of different blocks (i,j), we compute the collinearity cl_{ij}, that is, the determinant of the 2×2 submatrix of the matrix $\mathbb{L}$ of the previous step (7) corresponding to the blocks studied,
$$cl_{ij} = \det\begin{pmatrix} l_{ii} & l_{ij} \\ l_{ij} & l_{jj} \end{pmatrix}. \tag{8}$$
If this value is lower than a certain fixed tolerance, we can consider the two blocks as a unique block. At most, we would need to compute $N_b(N_b-1)/2$ coefficients as $cl_{ii}=0$ and $cl_{ij}=cl_{ji}$.

3. Test cases

This section is devoted to the validation and numerical study of all the aforementioned methods, POD, POBD and APOBD. We will apply these reduction techniques to problems arising from the mathematical modeling of biological phenomena exhibited in cells or neurons: Intracellular Calcium Concentration model [31,42], Hindmarsh–Rose model [33–36] and pancreatic β-cells models [37,38,40,43]. We will work with network models with a high number of coupled oscillators, and we will compare the reduction techniques by means of some classical indicators, such as speed-up and number of reduced equations, and also through the sparsity index, an indicator of the reproduction of the structure of the original problems based on the sparsity of the reduced matrices associated to each reduced model.

3.1. Intracellular calcium concentration network model

In this section, we test the aforementioned methods on the Intracellular Calcium Concentration model studied in [31,42]. In realistic problems, we have more than the two cells originally considered in those works. We consider here the case where we have $N \in \mathbb{N}$ different cells and symmetric coupling. We can model the system with N coupled oscillators as
$$O_i \begin{cases} \dot{x}_i = \tau(-y_i + f(x_i) - \phi_f(z_i)), \\ \dot{y}_i = \tau\varepsilon k_i(a_0x_i + a_1y_i + a_2 + \dfrac{1}{N/2}\displaystyle\sum_{j=1}^{N} c_{ij}(x_i - x_j)), \\ \dot{z}_i = \tau\varepsilon\left(\phi_r(x_i) - \dfrac{z_i - z_b}{\tau_z}\right), \end{cases} \tag{9}$$
where $i=1,2,\dots,N$, O_i denotes the ith oscillator formed by (x_i,y_i,z_i) and $c_{ij}\in[-1,1]$ with $c_{ij}=c_{ji}$ and
$$\phi_f(z) = \frac{\mu z}{z+z_0}, \phi_r(x) = \frac{\lambda}{1+\exp(-\rho(x-x_{on}))}. \tag{10}$$
In the following section, for parameters that are fixed among the simulations, we have kept the values used in [31,42]. Therefore, we consider $f(x) = -x^3+4x$ and
$$\begin{array}{llllll} a_1 = -0.1, & a_2 = 0.8, & \varepsilon = 0.06, & \tau = 37, & \mu = 2.4, & z_0 = 5, \\ z_b = 1, & \tau_z = 2, & x_{on} = -0.45, & \lambda = 1.75, & \rho = 4.5. & \end{array} \tag{11}$$

Thus, x would be the fast variable and y, z the slow ones. In a first approach, we consider the cells separated following a modular network structure with two clusters, namely, first cluster, are the cells with indexes $i=1,\dots,N_-$, and second cluster, the cells with indexes $i=N_-+1,\dots,N$. This choice is based on the motoneurons network properties studied in the zebrafish embryo in [44], and has been considered previously in [31]. We set $N_- = \lfloor N/2 \rfloor$, that is, the greatest integer lower than or equal to $N/2$, and $N_+ = \lceil N/2 \rceil$, that is, the lowest integer greater than or equal to $N/2$.

3.1.1. A priori study

In this section, we build the algebraic form of the original network problem (9). We denote $\boldsymbol{w} = (x_1, \dots, x_N, y_1, \dots, y_N, z_1, \dots, z_N)$, and we can write problem (9) as

$$\dot{\boldsymbol{w}} = f(\boldsymbol{w}, \boldsymbol{\gamma}), \text{ with } \boldsymbol{w}(0) = \boldsymbol{w}_0, \tag{12}$$

where $\boldsymbol{\gamma} = (c_\alpha, c_\beta, \boldsymbol{k})$, with k_i the heterogeneity parameters, and c_α, c_β the intra-cluster and inter-cluster coupling, respectively.

We note that we have changed the order of the variables in order to help the reader to understand the relationship between them, and that it has no consequences in the following study. First, for the sake of clarity, we introduce some notations.

Notation 3.1.

We denote by $\mathbf{0}_N$ the null column vector with N components, $\mathbf{1}$ the column vector with N components, and all its components 1 and $\boldsymbol{k}$ the column vector whose entries are the heterogeneity parameters. On the other hand, $\mathbb{I}_N$ is the identity matrix of order N, $\mathbb{O}_N$ is the null matrix of order N and $\mathbb{K} = \mathrm{diag}(\boldsymbol{k})$ is a diagonal matrix whose entries are the heterogeneity parameters.

Also, we define the following parameters,

$$\Sigma_i = \begin{cases} (N_- - 1)c_\alpha + N_+ c_\beta, & \textit{for } i = 1, \dots, N_-, \\ (N_+ - 1)c_\alpha + N_- c_\beta, & \textit{for } i = N_- + 1, \dots, N. \end{cases}$$

Note that, when N is even, $\Sigma_i = \Sigma = (N/2 - 1)c_\alpha + N c_\beta/2, \forall i = 1, \dots, N$.

Finally, we consider,

$$\mathbb{M}(c_\alpha, c_\beta) = \left(\begin{array}{cccc|cccc} \Sigma_1 & -c_\alpha & \dots & -c_\alpha & -c_\beta & \dots & \dots & -c_\beta \\ -c_\alpha & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & -c_\alpha & \vdots & & & \vdots \\ -c_\alpha & \dots & -c_\alpha & \Sigma_{N_-} & -c_\beta & \dots & \dots & -c_\beta \\ \hline -c_\beta & \dots & \dots & -c_\beta & \Sigma_{N_-+1} & -c_\alpha & \dots & -c_\alpha \\ \vdots & & & \vdots & -c_\alpha & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \ddots & \ddots & -c_\alpha \\ -c_\beta & \dots & \dots & -c_\beta & -c_\alpha & \dots & -c_\alpha & \Sigma_N \end{array} \right)$$

Thus, top left submatrix is $N_- \times N_-$, top right is $N_- \times N_+$, bottom left is $N_+ \times N_-$, and bottom right is $N_+ \times N_+$. We remark here that the matrix $\mathbb{M}(c_\alpha, c_\beta)$ admits an affine decomposition depending on c_α, c_β and Σ_i.

Injecting the previous notation in Eq. (12), we can rewrite system (9) as

$$\dot{\boldsymbol{w}} = \tau((\mathbb{A}_{np} + \mathbb{A}_p(\boldsymbol{k}) + \mathbb{B}(c_\alpha, c_\beta, \boldsymbol{k}))\boldsymbol{w} + \boldsymbol{b}_{np} + \boldsymbol{b}_p(\boldsymbol{k}) + \boldsymbol{b}_{nl}(\boldsymbol{w})), \tag{13}$$

where $\mathbb{A}_{np}$ and $\boldsymbol{b}_{np}$ represent the non-parametric part, $\mathbb{A}_p(\boldsymbol{k})$ and $\boldsymbol{b}_p(\boldsymbol{k})$ the parametric part depending on the heterogeneity parameters, $\mathbb{B}(c_\alpha, c_\beta)$ is the parametric part relative to the coupling and $\boldsymbol{b}_{nl}(\boldsymbol{w})$ denotes the nonlinear part of system (9). They can be expressed as follows

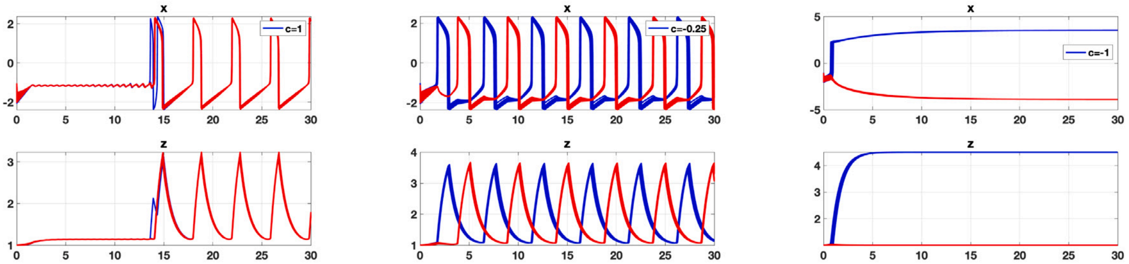
$$\mathbb{A}_{np} = \left(\begin{array}{c|c|c} 4\mathbb{I}_N & -\mathbb{I}_N & \mathbb{O}_N \\ \hline \mathbb{O}_N & \mathbb{O}_N & \mathbb{O}_N \\ \hline \mathbb{O}_N & \mathbb{O}_N & -\varepsilon/\tau_z \mathbb{I}_N \end{array} \right), \qquad \mathbb{A}_p(\boldsymbol{k}) = \varepsilon \left(\begin{array}{c|c|c} \mathbb{O}_N & \mathbb{O}_N & \mathbb{O}_N \\ \hline a_0\mathbb{K} & a_1\mathbb{K} & \mathbb{O}_N \\ \hline \mathbb{O}_N & \mathbb{O}_N & \mathbb{O}_N \end{array} \right),$$

$$\mathbb{B}(c_\alpha, c_\beta) = \frac{\varepsilon}{N/2} \left(\begin{array}{c|c|c} \mathbb{O}_N & \mathbb{O}_N & \mathbb{O}_N \\ \hline \mathbb{K}\mathbb{M}(c_\alpha, c_\beta) & \mathbb{O}_N & \mathbb{O}_N \\ \hline \mathbb{O}_N & \mathbb{O}_N & \mathbb{O}_N \end{array} \right),$$

$$\boldsymbol{b}_{np} = \frac{\varepsilon z_b}{\tau_z} \begin{pmatrix} \mathbf{0}_N \\ \mathbf{0}_N \\ \mathbf{1}_N \end{pmatrix}, \quad \boldsymbol{b}_p(\boldsymbol{k}) = \varepsilon \begin{pmatrix} \mathbf{0}_N \\ a_2\boldsymbol{k} \\ \mathbf{0}_N \end{pmatrix}, \quad \boldsymbol{b}_{nl}(\boldsymbol{w}) = \begin{pmatrix} -x_1^3 - \phi_f(z_1) \\ \vdots \\ -x_N^3 - \phi_f(z_N) \\ \mathbf{0}_N \\ \varepsilon\phi_r(x_1) \\ \vdots \\ \varepsilon\phi_r(x_N) \end{pmatrix}.$$

With a first look into the model, we can easily see that system (13) depends on many parameters. In the following study, we set random heterogeneity parameters $k_i \in [1, 1.5]$ following a truncated normal distribution $\mathcal{N}(1.25, 0.25)$, fixed for all the simulations, and fix $c_\alpha = 1$, that is, we consider that the cells within each cluster are coupled in an excitatory fashion. Therefore, we perform our reduction with respect to the inter-cluster coupling parameter c_β, that is, the coupling between cells of different clusters. This selection is made to be in concordance with previous works [31], and to avoid chaos and bizarre behaviors. As a matter of fact, chaotic behaviors are tremendously sensitive to perturbations. Consequently, a test case for any reduction process aiming at accelerating the simulations cannot be successful in the presence of chaos. Nevertheless, we comment this case and show corresponding simulations in Appendix A.1.

Before proceeding to the computational study, we need to set which behaviors we are trying to model, along with its range on the parameter c_β and also the number of blocks for the POBD. We will focus on three main behaviors:



(a) AiP synchronization behavior of system (13) for $c_\beta = 1$.

(b) AS behavior of system (13) for $c_\beta = -0.25$.

(c) TOD behavior of system (13) for $c_\beta = -1$.

Fig. 1. Representation of the different behaviors to study. In blue, the cells belonging to the first cluster and in red, the cells belonging to the second cluster. We only represent x and z, for the sake of clarity.

1. **Almost-in-Phase synchronization (AiP).** For $c_\beta \in [0.1, 1]$.
   In this range of the parameter, each cell reaches asymptotically an orbit similar to the cycle existing for the relaxation oscillation of the uncoupled case without Mixed Mode Oscillations (MMOs). All the 3D oscillators are synchronized with a small phase-shift or change of frequency depending on its heterogeneity parameter. This behavior is shown in Fig. 1(a).
2. **Antiphase Synchronization (AS).** For $c_\beta \in [-0.45, -0.05]$.
   In this range of the parameter, each cell within each cluster reach asymptotically the same orbit, and the phase-shift between the clusters is close to half the period with a small phase-shift or change of frequency depending on its heterogeneity parameter. This behavior is shown in Fig. 1(b).
3. **Total Oscillation Death (TOD).** For $c_\beta \in [-1, -0.55]$.
   In this range of the parameter, each cell within each cluster reaches asymptotically a stable equilibrium. This behavior is shown in Fig. 1(c).

As we have two clusters with three types of variables we will perform the POBD with 3 and 6 blocks, the first POBD will have one block per variable $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, and we will name it POBD3, and the second POBD will have one block per cluster variable $(\boldsymbol{x}_I, \boldsymbol{x}_{II}, \boldsymbol{y}_I, \boldsymbol{y}_{II}, \boldsymbol{z}_I, \boldsymbol{z}_{II})$, where subindex I denotes the variables relatives to the first cluster and subindex II denotes the variables relatives to the second cluster. We will name this method POBD6. As APOBD groups the blocks automatically, we start with the maximum number of possible blocks, in our case 6, one per cluster and variable.

3.1.2. Numerical results

In this section, we will perform various numerical tests for different number of cells, namely, $N = 200, 500, 1000, 2000$.

In every test case, we will let the original system (13) compute from $T = 0$ to $T = 50$ via a fourth order Runge–Kutta solver with fixed time step of $h = 10^{-3}$. We perform every method with the data from $T = 35$ to $T = 50$ as we need that the system is almost at an asymptotic behavior.

In the construction of reduced systems, the tolerance criteria for the collinearity analysis in the APOBD method is set at 0.2, that give us two blocks in AiP behavior, one for the fast variables $(\boldsymbol{x})$ and another one for the slow ones $(\boldsymbol{y}, \boldsymbol{z})$, four blocks in AS behavior, $(\boldsymbol{x}_I)$, $(\boldsymbol{x}_{II})$, $(\boldsymbol{y}_I, \boldsymbol{z}_I)$ and $(\boldsymbol{y}_{II}, \boldsymbol{z}_{II})$, that is, one block per timescale and cluster and just one block for TOD behavior, that is, all the variables mixed together.

The tolerance criteria of the selection of the reduced dimension in POD methods is set to 10^{-9} for AiP behavior and 10^{-6} for AS and TOD behaviors. Nonlinear terms of (13) are addressed through the Empirical Interpolation Method [45,46] (from now on, EIM) with a tolerance of 10^{-6}. This is a technique that allows us to compute the nonlinearities of the system as a linear combination of some basis.

In the following, we compare the number of equations, the sparsity of the reduced model via the sparsity index, and the computational time spent to solve the reduced system from $T = 50$ to $T = 70$, via the speed-ups, defined as the quotient between the computational time of the original system (13) and the computational time of the reduced system.

The objectives of this test case are the following: First, to validate the new proposed methods. In order to do so, we compare all the methods with $N = 200$ cells in the three behaviors mentioned above. Then, in order to compare in more detail the methods, we will test them with a more large amount of cells, that is, $N = 500, 1000, 2000$ for AiP and AS behaviors, as the TOD behavior has not a major interest, as it reproduces the system in a steady state and the reduction is straightforward. In Table 2, we show the simulation data of the original system (13) for every test case.

Validation of the methods for $N = 200$

The results of the reductions are summarized in Table 3. We see that for this test, POD performs better in terms of speed-up just for AiP behavior, but in terms of sparsity POBD6 provides the best index in the three cases, that is, the best separation of variables. We see that both POBD3 and APOBD provide a well trade-off between sparsity and speed-up. For AS behavior, POBD6 and APOBD

Table 2
Data for system (13) for different test cases.

Test case	N = 200	N = 500	N = 1000	N = 2000
Number of equations	600	1500	3000	6000
Sparsity index	0.8867	0.8880	0.8884	0.8887
CPU time	≈6.2 s	≈14 s	≈126 s	≈475 s

Table 3
Comparison of the data for the reduced system for the original system (13) with N = 200. Notation a+...+c indicates the number of equations in each block. In bold fashion, the best results for each behavior.

Behavior	Method	POD	POBD3	POBD6	APOBD
AiP	Number of equations	33	26+12+10	21+23+10+11+8+9	26+16
	Sparsity index	0	0.6510	**0.7518**	0.3685
	Speed-up	**6.04**	4.20	2.70	4.24
	Error $(\cdot 10^{-3})$	5.06	4.46	6.95	**4.35**
AS	Number of equations	57	53+14+8	27+26+7+7+4+4	27+26+9+8
	Sparsity index	0	0.6874	**0.7739**	0.6937
	Speed-up	2.02	1.67	**2.20**	**2.17**
	Error $(\cdot 10^{-3})$	**3.59**	7.80	6.24	6.98
TOD	Number of equations	1	1+1+1	1+1+1+1+1+1	1
	Sparsity index	0	0.4444	**0.6667**	0
	Speed-up	34	15.32	9.29	**40.06**
	Error $(\cdot 10^{-4})$	3.09	0.40	**0.21**	3.09

Table 4
Comparison of the number of equations of the reduced systems for the original system (13) in AiP behavior.

Number of cells	POD	POBD3	POBD6	APOBD
500	36	29+14+11	25+27+12+13+10+11	29+18
1000	39	32+14+12	28+31+12+14+10+12	32+18
2000	42	36+15+12	35+34+15+14+12+11	36+18

are the methods that provide the best speed-ups and sparsity indexes. For TOD behavior, APOBD provides the best speed-up, while POBD6 provides the best separation of variables and error.

For illustration purposes, in Fig. 2 we show the first 3 basis obtained for each method and each block in AS behavior. Note that the basis for POD (Fig. 2(a)) and POBD3 (Fig. 2(b)) are much more erratic, and we cannot identify neither the separation between variables nor clusters, while for POBD6 (Fig. 2(c)) and APOBD (Fig. 2(d)) this separation is clearer, due to the higher sparsity index. This means that, we could not need to go back to the full-dimensional space to analyze the data, allowing us to save more computational time.

At this point, we can affirm that this validation test is finished successfully. The reduced methods proposed are able to reproduce the solution of the original problem while performing a reduction in computational times. It should be noted that while all the methods are comparable in terms of sparsity and error, the novel methods also provide us with a new feature, the perseverance of the structure of the original system. Now, we are able to continue our study. In next sections, we will focus just on the AiP and AS behaviors. As mentioned in the previous, the TOD behavior is not of significant interest as it simply replicates the system in a steady state, and the reduction process is straightforward.

Comparison for larger number of cells in AiP behavior

In this section, we study in detail the reduction in the AiP behavior of system (13) for larger numbers of cells, $N = 500, 1000, 2000$.

The results are summarized in Table 4, which shows the number of variables of each cell number case, and Figs. 3(a)–3(c), which represent the evolution of sparsity, speed-up and error, respectively. From Table 4, we can see that APOBD minimizes the number of equations with respect to the POBD methods, while maintaining the structure of the system.

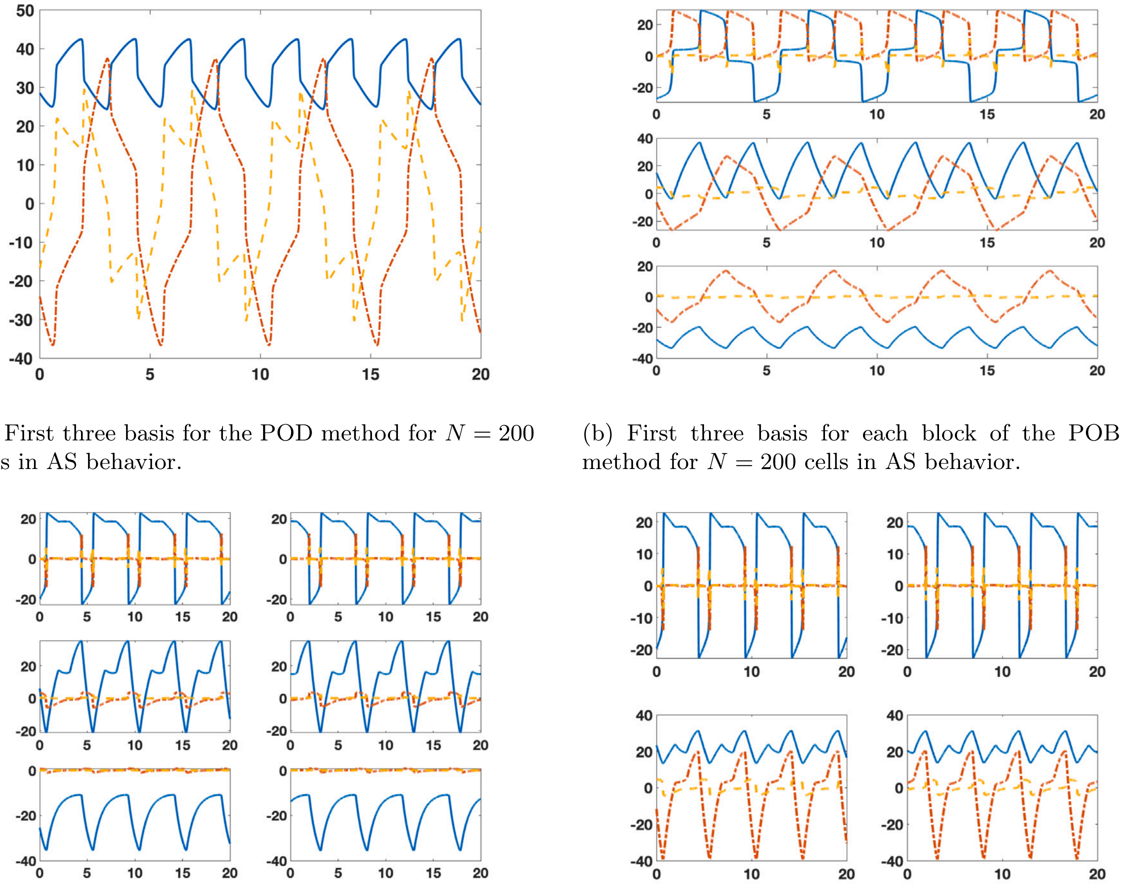
The best method in terms of sparsity is POBD6 in every case, see Fig. 3(a), however, a better sparsity index does not imply a better speed-up, as it is also the worst method in terms of speed-up, see Fig. 3(b). It seems that the best method in terms of the tradeoff between sparsity and speed-up is APOBD. We can observe that the relative errors are comparable, as they have the same order, see Fig. 3(c).

Comparison for larger number of cells in AS behavior

In this section, we study in detail the reduction in the AS behavior of system (13) for larger numbers of cells, $N = 500, 1000, 2000$.

The results are summarized in Table 5, which shows the number of variables of each cell number case, and Figs. 4(a)–4(c), which represent the evolution of sparsity, speed-up and error, respectively. From Table 5, we can see again that APOBD minimizes the number of equations with respect to the POBD methods, while maintaining the structure of the system.

The best method in terms of sparsity is POBD6 in every case, see Fig. 4(a), however, a better sparsity index does not imply a better speed-up, as it is not the best method in terms of speed-up, see Fig. 4(b). We can see that methods POBD3 and APOBD



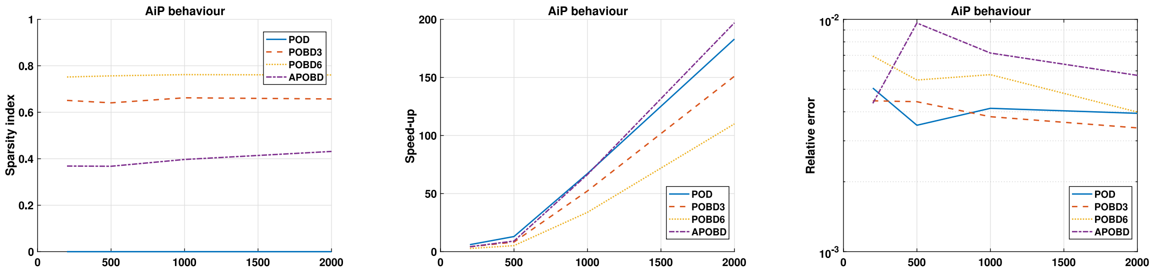
(a) First three basis for the POD method for $N = 200$ cells in AS behavior.

(b) First three basis for each block of the POBD3 method for $N = 200$ cells in AS behavior.

(c) First three basis for each block of the POBD6 method for $N = 200$ cells in AS behavior.

(d) First three basis for each block of the APOBD method for $N = 200$ cells in AS behavior.

Fig. 2. Comparison of the basis for each method. In all the cases, the first basis is in blue continuous fashion, the second in red dash-dotted fashion and the third basis in yellow dashed fashion. (a) POD. (b) POBD3. (c) POBD6. (d) APOBD.



(a) Comparison of the sparsity index for each method in AiP behavior.

(b) Comparison of the speed-up for each method in AiP behavior.

(c) Comparison of the relative error for each method in AiP behavior.

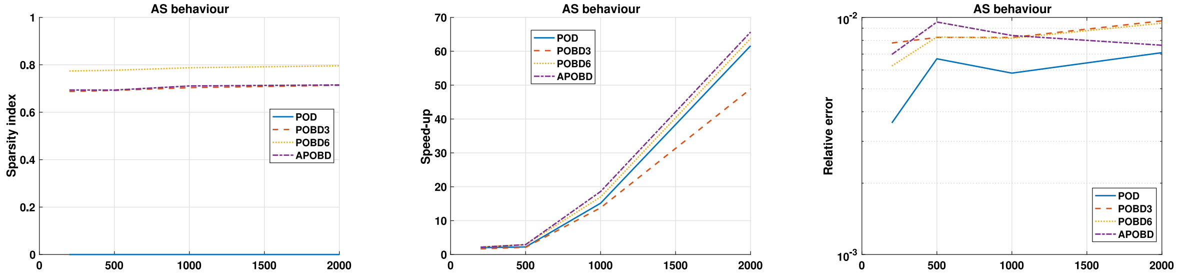
Fig. 3. Comparisons of the results for each method in AiP behavior of system (13).

provide a similar sparsity index, but APOBD is much better in terms of speed-up. Again, it seems that the best method in terms of the tradeoff between sparsity and speed-up is APOBD. Again, the relative errors are comparable between the methods, see Fig. 4(c).

It is clear that our proposed method APOBD gives the best results in terms of the trade-off between model reduction, saving of computational time and explainability of the reduced variables, as it needs a number of equations that is very low in comparison

Table 5
Comparison of the number of equations of the reduced systems for the original system (13) in AS behavior.

Number of cells	POD	POBD3	POBD6	APOBD
500	66	61+16+ 9	31+30+ 8+ 8+ 5+ 4	31+30+10+ 9
1000	71	68+26+ 8	33+35+ 8+ 8+ 4+ 5	33+35+ 9+10
2000	75	72+16+ 8	36+37+ 8+ 8+ 5+ 4	36+37+10+10



(a) Comparison of the sparsity index for each method in AS behavior.

(b) Comparison of the speed-up for each method in AS behavior.

(c) Comparison of the relative error for each method in AS behavior.

Fig. 4. Comparisons of the results for each method in AS behavior of system (13).

with the original system, and the computational times reductions and errors are competitive for all the behaviors studied. We can also note that APOBD performs better the larger the number of equations of the original problem are. Furthermore, it also gives us a new feature, explainability of the reduced variables, that could also help the post-analysis of the results as we do not need to go back to the full-dimensional space.

3.2. Hindmarsh-Rose network model

In this section, we will consider another type of single-cell dynamics used for building a network model. The Hindmarsh–Rose model formalizes the dynamics of neuronal activity in a single neuron and was initially introduced in [36]. We quickly describe how it is built. The model involves three state variables: the membrane potential x, a recovery variable y controlling the neuron ability to recover from depolarization and a gating variable z regulating the dynamics of the neuron's spiking activity. The membrane potential is mainly driven by a nonlinear voltage-dependent conductance term that captures the neuron's response to input currents: a cubic polynomial function introduces bistability and the potential for oscillations. Recovery and gating variables are driven by linear intrinsic terms. Additionally, the coupling term of y upon x-dynamics is linear and influences the excitability by modulating the rate at which the recovery variable affects the membrane potential: an increase in the recovery variable y leads to an increase in the membrane potential x, enhancing the excitability of the neuron. The coupling term of the gating variable onto the potential dynamics is also linear, but inhibitory. Feedback loops are introduced by an excitatory (resp. inhibitory) coupling from the membrane potential onto the recovery variable dynamics (resp. gating variable). Several types of feedback terms have been considered in modified versions of Hindmarsh–Rose model (see, for instance, [8,47,48]): in the following, we consider a quadratic term for the feedback of x upon y, ensuring that the feedback is always inhibitory, and a linear one upon z. Moreover, in its slow–fast version, one currently considers the gating variable dynamics to be slow compared to the membrane potential and recovery variable dynamics, which is materialized by parameter ε, assumed to be small and positive, in the z-dynamics.

The Hindmarsh–Rose model exhibits a wide range of dynamical behaviors depending on the parameter values and therefore is able to capture a wide panel of neuronal activity features, e.g. sustained oscillations (or regular spiking), various types of bursting oscillations, chaotic oscillations, etc. In the following, we will focus on two dynamical behaviors of the single-cell dynamics: sustained oscillations and plateau-like bursting that will be further described.

We will test the POD and the APOBD on the following network of Hindmarsh–Rose Network system that has been introduced and studied in [33–35]:

$$O_i \begin{cases} \dot{x}_i = l(x_i) + y_i - z_i + I + \dfrac{1}{N-1}\displaystyle\sum_{i=1}^{N} c_{ij}(x_j - x_i), \\ \dot{y}_i = c + m(x_i) - y_i, \\ \dot{z}_i = \varepsilon(k_i(x_i - x_0) - z_i), \end{cases} \tag{14}$$

where $i = 1, 2, \ldots, N$, O_i denotes the ith oscillator formed by (x_i, y_i, z_i), $l(x) = -ax^3 + bx^2$ and $m(x) = -dx^2$. In the following, x represents the membrane potential and y and z take into account the transport of ions across the membrane through the ion channels. The timescale separation parameter ε fulfills $0 < \varepsilon \ll 1$, parameters $k_i > 0$ has been introduced so that the outputs fit realistic biological evolution patterns and parameter I stands for the external exciting current. Finally, a, b, c and d are positive parameters, and $x_0 < 0$.

In this test, we consider a normalized version of the coupling, with $c_{ij} = 1, \forall i \neq j$, and $c_{ii} = 0$. We also add some heterogeneity in the slow equation via $k_i \in [3.9, 4.1]$ following a uniform distribution.

3.2.1. *A priori study*

Note that, if we consider $\mathbb{D} = (d_{ij})$, with

$$d_{ij} = \frac{1}{N-1}\left(\delta_{ij}\sum_{s=1}^{N} c_{sj} - c_{ij}\right),$$

where δ_{ij} is the Kronecker delta, the first equation can be rewritten as

$$\dot{x}_i = l(x_i) + y_i - z_i + I + \sum_{j=1}^{N} d_{ij}x_j.$$

This allows us to rewrite the system in vector notation

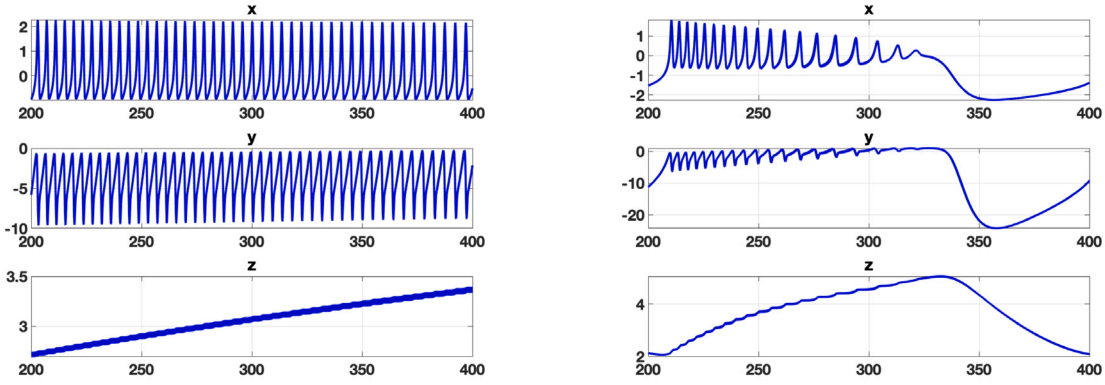
$$\boldsymbol{w} = \mathbb{A}\boldsymbol{w} + \boldsymbol{b} + \boldsymbol{b}_{nl}(\boldsymbol{w}), \tag{15}$$

where

$$\mathbb{A} = \left(\begin{array}{c|c|c} \mathbb{D} & \mathbb{I}_N & -\mathbb{I}_N \\ \hline \mathbb{O}_N & -\mathbb{I}_N & \mathbb{O}_N \\ \hline \varepsilon\mathbb{K} & \mathbb{O}_N & -\varepsilon\mathbb{I}_N \end{array}\right), \quad \boldsymbol{b} = \begin{pmatrix} I\mathbf{1}_N \\ c\mathbf{1}_N \\ -\varepsilon x_0\boldsymbol{k} \end{pmatrix}, \quad \boldsymbol{b}_n l(\boldsymbol{w}) = \begin{pmatrix} \boldsymbol{l}(\boldsymbol{x}) \\ \boldsymbol{m}(\boldsymbol{x}) \\ \mathbf{0}_N \end{pmatrix}.$$

Before the computational study, as we did in the previous test case, we need to set which behavior we are trying to model, along with its parameters values and also the number of initial blocks for the APOBD. In this case, we will focus on two different behaviors:

1. **Sustained Oscillation (SO).**
   This behavior was considered in [34]. It is obtained for parameters with classical values $a = 1$, $b = 3$, $c = 1$ and $d = 5$. The separation between timescales is set to $\varepsilon = 10^{-3}$ and the external exciting current is set to $I = 5$.
   For this selection of the parameters, each cell is spiking with no quiescent phase, and all 3D oscillators are strongly synchronized with a slight phase shift. This behavior is shown in Fig. 5(a).
2. **Plateau-like Bursting (PB).**
   This behavior was studied in [33]. It is obtained for parameter values $a = 1$, $b = 2.52$, $c = 1$ and $d = 5$. The separation between timescales is set to $\varepsilon = 10^{-2}$ and the external exciting current is set to $I = 4$.
   Plateau-like bursting is characterized by repeated cycles of spiking followed by a prolonged depolarized state known as plateau. During the bursting phase, the membrane potential x gradually increases due to the inward current, leading to depolarization and spike trigger. However, under the impact of the recovery variable y, the neuron does not immediately return to its resting state. Instead, y inhibits the repolarization process, prolonging the depolarized state and leading to the characteristic plateau. This is achieved through the negative feedback mechanism, where an increase in the membrane potential promotes the recovery variable, which in turn inhibits the repolarization process. Once the gating variable gradually decreases or reaches a critical value, it counterbalances the impact of the recovery variable. As a result, the membrane potential rapidly returns to its resting state, initiating the next cycle of bursting.
   The dynamical mechanism underlying this feature can be roughly explained (see [48,49] for more details) using slow–fast approximation. The critical manifold is an S-shaped curve, and its two folds correspond to saddle–node bifurcations of the fast dynamics (x, y): the one corresponding to the lowest value of z is responsible for the burst trigger after the plateau. On the opposite branch of the critical manifold, the singular point of the fast dynamics undergoes a Hopf bifurcation (for low value of z), which gives birth to a stable limit cycle persisting for higher z values, and eventually disappears through a homoclinic bifurcation. The second saddle–node bifurcation appears for a higher value of z. Hence, an orbit generating Plateau-like Bursting alternatively follows the attractive branch of the critical manifold and the family of fast limit cycles. The transition from spiking regime to quiescence phase results from the homoclinic connection, i.e., the disappearance of all attractive invariant for higher value of z except the attractive singular point associated with the lowest x value.
   We have deliberately chosen the parameter values mentioned above for obtaining a homoclinic bifurcation close to the higher saddle–node bifurcation. In that case, we obtain synchronization/desynchronization between cells in the network model. More precisely, all the cells starts their spiking regime at the same time, gradually desynchronize their spikes during the spiking regime, and resynchronize during the quiescence phase. Such behavior is therefore sufficiently non-trivial for testing the POD and APOBD methods.



(a) SO behavior of system (14) for $N = 100$ cells.

(b) PB behavior of system (14) for $N = 100$ cells.

Fig. 5. Representation of the different behaviors to study of system (14).

Table 6
Simulation data for system (15).

Test case	$N = 100$	$N = 500$	$N = 1000$
Number of equations	300	1500	3000
Sparsity index	0.8833	0.8878	0.8883
CPU time	≈5.5 s	≈100 s	≈415 s

Table 7
Comparison of the number of equations of the reduced systems for the original system (15) in SO behavior.

Number of cells	POD	APOBD
100	15	7+7+3
500	15	7+7+3
1000	16	7+7+3

3.2.2. Numerical results

In the following study, we will consider various instances for the number of cells $N = 100, 500, 1000$. In every test case, we will let the original system (15) compute from $T = 0$ to $T = 400$ via a fourth order Runge–Kutta solver with fixed time step of $h = 10^{-2}$. And we perform both reduction methods with the data from $T = 200$ to $T = 400$ as we need that the system is almost at an asymptotic behavior. This will be enough to capture the main behavior of the system, as shown in Fig. 5.

In the construction of reduced systems, the tolerance criteria for the collinearity analysis in the APOBD method is set at 0.2, that gives us three blocks for the SO and PB behaviors. The tolerance criteria of the selection of the reduced dimension in POD methods is set to 10^{-12} for every behavior. Again, nonlinear parts of (15) are addressed by means of the EIM [45,46], now with a tolerance of 10^{-12}.

Table 6 summarizes the simulation data of the original system (15) for both test cases.

As done for the previous tests, we compare the number of equations, the sparsity index, and the speed-ups, from $T = 400$ to $T = 1000$, for the original model and the reduced ones. Furthermore, as an additional way to compare the reduced solutions with the original one, we compute the following a-posteriori error estimator,

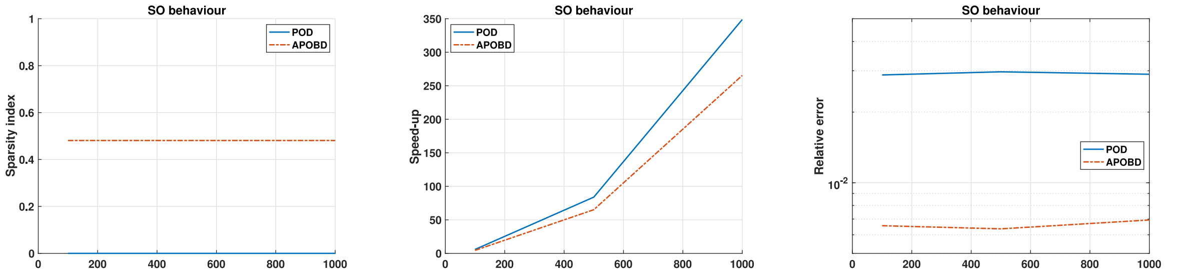
$$\Delta_j = \max_i \min_k |t_j^{(i)} - \tilde{t}_k^{(i)}|, \tag{16}$$

where $t_j^{(i)}$ is the jth time when the original model has a local maximum in the x variable for the cell i, and $\tilde{t}_k^{(i)}$ is the kth time when the reduced model has a local maximum in the x variable for the cell i. This estimator tells us how shifted are the reduced peaks in comparison to the original one. Next, we summarize the results obtained for each behavior.

Numerical results for SO behavior

In this section, we study in detail the reduction in the SO behavior of system (15) for $N = 100, 500, 1000$.

The results are summarized in Table 7, which shows the number of variables of each cell number case, and Figs. 6(a)–6(c), which represent the evolution of sparsity, speed-up and error, respectively. From Table 7, we can see that the reduction is stronger when the number of original equations rises, as the number of reduced equations remains almost the same.

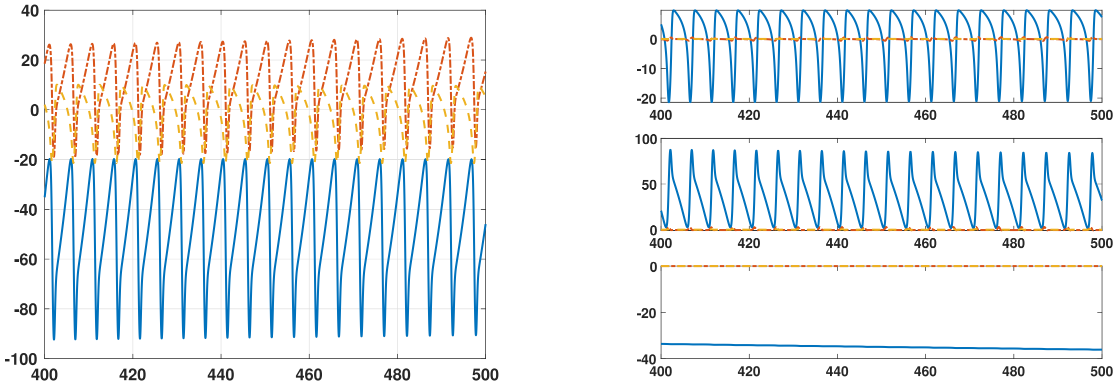


(a) Comparison of the sparsity index for each method in SO behavior.

(b) Comparison of the speed-up for each method in SO behavior.

(c) Comparison of the relative error for each method in SO behavior.

Fig. 6. Comparisons of the results for each method in SO behavior of system (15).



(a) First basis for the POD method for $N = 100$ cells in sustained oscillation behavior.

(b) First basis for each block of the APOBD method for $N = 100$ cells in sustained oscillation behavior.

Fig. 7. Comparisons of the basis for each method obtained in the sustained oscillation behavior. In all the cases, the first basis is in blue continuous fashion, the second in red dash-dotted fashion and the third basis in yellow dashed fashion. (a) POD. (b) APOBD with 3 blocks.

Table 8
Comparison of the number of equations of the reduced systems for the original system (15) in PB behavior.

Number of cells	POD	APOBD
100	15	7+6+4
500	15	7+6+4
1000	15	7+6+4

We can see that although the speed-up is better for the POD method (see Fig. 6(b)), APOBD provides an additional order in the relative error, Fig. 6(c).

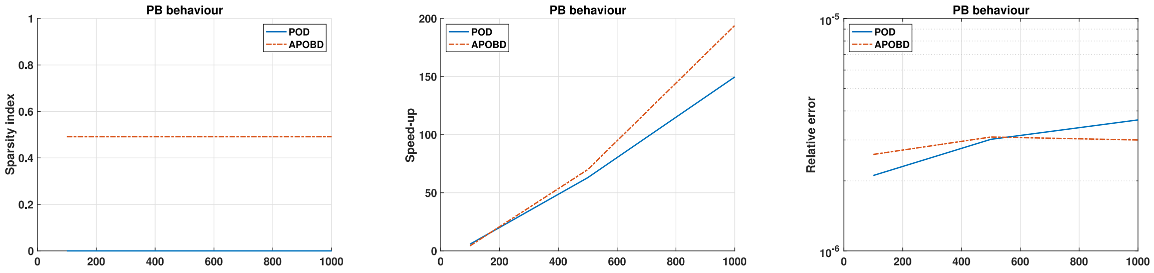
For illustration purposes, in Fig. 7, we show the first 3 basis obtained for each method and each block in SO behavior the basis for POD (Fig. 7(a)), where we cannot see a separation in the timescales, and APOBD (Fig. 7(b)), where this separation is clearer, due to the higher sparsity index.

Numerical results for PB behavior

In this section, we study in detail the reduction in the PB behavior of system (15) for $N = 100, 500, 1000$.

The results are summarized in Table 8, which shows the number of variables of each cell number case, and Figs. 8(a)–8(c), which represent the evolution of sparsity, speed-up and error, respectively. From Table 8, we can see again that the reduction is stronger when the number of original equations rises, as the number of reduced equations remains almost the same.

It is clear that the APOBD method provides a better speed-up in comparison with the POD for this behavior, maintaining an error of the same order as the one of the POD method.

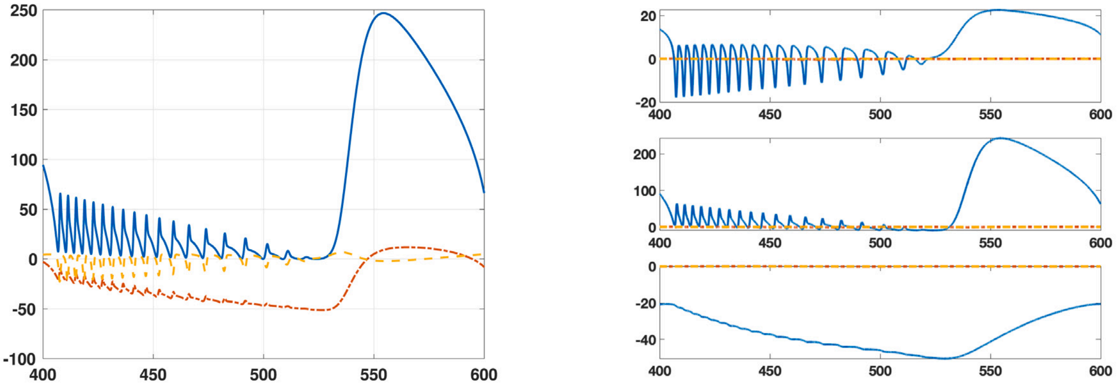


(a) Comparison of the sparsity index for each method in PB behavior.

(b) Comparison of the speed-up for each method in PB behavior.

(c) Comparison of the relative error for each method in PB behavior.

Fig. 8. Comparisons of the data obtained for each method in PB behavior of system (15).



(a) First basis for the POD method for $N = 100$ cells in PB behavior.

(b) First basis for each block of the APOBD method for $N = 100$ cells in PB behavior.

Fig. 9. Comparisons of the basis for each method obtained in the PB behavior. In all the cases, the first basis is in blue continuous fashion, the second in red dash-dotted fashion and the third basis in yellow dashed fashion. (a) POD. (b) APOBD.

For illustration purposes, we show in Fig. 9 a comparison between the first 3 basis obtained for each method and block in the PB behavior for POD (Fig. 9(a)) and APOBD (Fig. 9(b)). In the POD case, we cannot observe a separation in the timescales, while in the APOBD case this separation is clearer, due to the higher sparsity index.

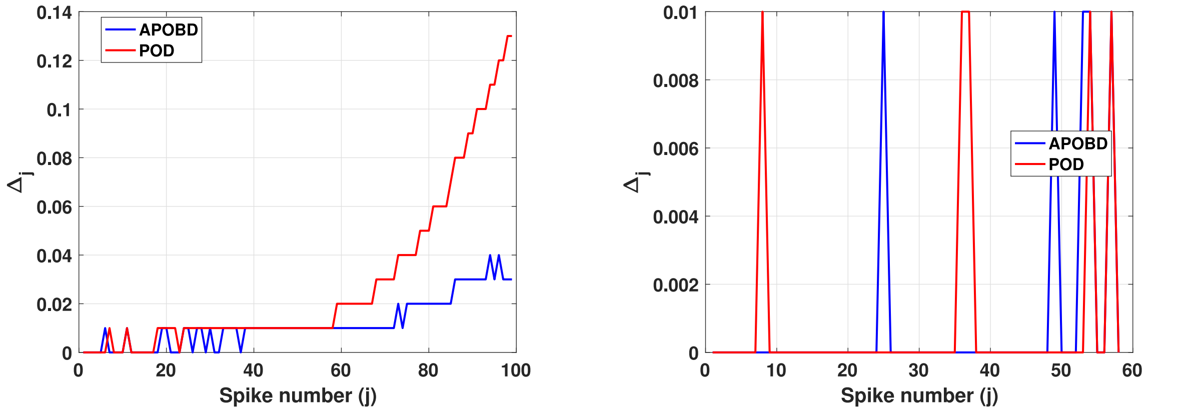
In Fig. 10, we show the comparison of the error estimators (16) for both behaviors in the case with $N = 100$, as it is similar for $N = 500$ and $N = 1000$. We see that for the integrating time, APOBD and POD have a great precision in terms of reproducing the spiking times correctly for the PB behavior as the error is at most the integrating time step h, see Fig. 10(b). However, for the SO behavior, as the time increases this precision is lost. This could be due to the transition to a qualitatively different behavior after a stable transient, as it will be discussed in Appendix. Nevertheless, according to Fig. 10(a), APOBD performs a more accurate approximation of the spiking times and bursting duration, which is consistent with the relative error shown in Fig. 6(c).

It seems that APOBD method performs better when there is a switch between the fast behavior and the slow transient. To test it more in detail, we will compare it with the POD method in another model that exhibits this kind of behavior.

3.3. Fold/Hopf bursting network model

In this section, we test the POD and APOBD methods in a case of "Fold/Hopf" (also called "tapered") bursting studied, for instance, in [38]. As reported in [38], this type of bursting has been found in models of insulin-producing pancreatic β-cells [37,39,40] and in models of some enzymatic systems [50,51].

In this type of burster, the stable equilibrium corresponding to the resting state disappears through a fold bifurcation and the limit cycle attractor corresponding to the spiking state shrinks to a point though a supercritical Hopf bifurcation. Moreover, the fast subsystem undergoes another bifurcation while in the excited state, a fold bifurcation corresponding to the transition from the excited equilibrium to the resting equilibrium, closing the hysteresis loop.



(a) Comparison of the error estimators for sustained oscillation behavior of system (14) with $N = 100$ cells.

(b) Comparison of the errors for PB behavior of system (14) with $N = 100$ cells.

Fig. 10. Comparison of errors of the reduced solutions of system (14) with $N = 100$, in red fashion, POD, and in blue fashion, APOBD. (a) SO. (b)PB.

The Fold/Hopf Bursting network model can be written as follows,

$$O_i \begin{cases} \dot{x}_i = x_i - x_i^3/3 - y_i + \dfrac{1}{N-1}\displaystyle\sum_{i=1}^{N} c_{ij}(x_j - x_i), \\ \dot{y}_i = \varepsilon(a + x_i - S(y_i, z_i)), \\ \dot{z}_i = \mu k_i x_i, \end{cases} \tag{17}$$

where $i = 1, 2, \dots, N$, O_i denotes the ith oscillator formed by (x_i, y_i, z_i), $S(y, z) = b/(1 + \exp((z - y)/d))$, $a = 1.55$, $b = 2.5$, $d = 0.1$, and the timescale separation parameters are set so that the system has three timescales, $\varepsilon = 0.5$ and $\mu = 0.01$. Note that, then, this model is different from the previous ones in terms of separation timescales as is the only one featuring three timescales $1, \epsilon$ and μ.

In this test, we consider again a normalized version of the coupling, with $c_{ij} = 1, \forall i \neq j$, and $c_{ii} = 0$. We also add some heterogeneity in the slow equation via $k_i \in [3.9, 4.1]$ following a uniform distribution.

3.3.1. *A priori study*

Note that, if we consider, as in the previous case, $\mathbb{D} = (d_{ij})$, with

$$d_{ij} = \frac{1}{N-1}\left(\delta_{ij}\sum_{s=1}^{N} c_{sj} - c_{ij}\right),$$

the first equation can be rewritten as

$$\dot{x}_i = x_i - x_i^3/3 - y_i + \sum_{j=1}^{N} d_{ij} x_j.$$

This allows us to rewrite the system in vector notation

$$\boldsymbol{w} = \mathbb{A}\boldsymbol{w} + \boldsymbol{a} + \boldsymbol{b}_{nl}(\boldsymbol{w}), \tag{18}$$

where

$$\mathbb{A} = \left(\begin{array}{c|c|c} \mathbb{I}_N + \mathbb{D} & -\mathbb{I}_N & \mathbb{O}_N \\ \hline \varepsilon\mathbb{1}_N & \mathbb{O}_N & \mathbb{O}_N \\ \hline \mu\mathbb{K} & \mathbb{O}_N & \mathbb{O}_N \end{array}\right), \quad \boldsymbol{b} = \begin{pmatrix} \boldsymbol{0}_N \\ \varepsilon a\boldsymbol{1}_N \\ \boldsymbol{0}_N \end{pmatrix}, \quad \boldsymbol{b}_n l(\boldsymbol{w}) \begin{pmatrix} \boldsymbol{x}^3/3 \\ \frac{1}{1+\exp(z-y)/d} \\ \boldsymbol{0}_N \end{pmatrix}.$$

Before proceeding to do the computational study, as we did in the previous test cases, we need to set which behavior we are trying to model and also the number of initial blocks for the APOBD, which we set to three, one per variable. In this case, we will focus on the behavior already described:

1. **Fold/Hopf Bursting (FHB).** The resting state disappears through a fold bifurcation, and the spiking state disappears through a supercritical Hopf bifurcation. Moreover, the hysteresis loop is closed through another fold bifurcation. See Fig. 11.

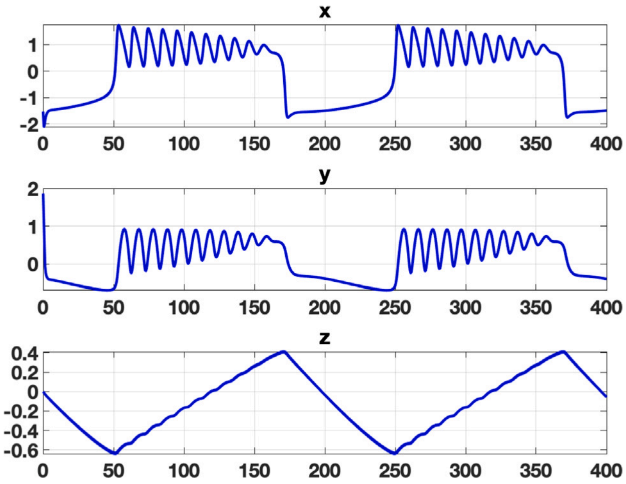


Fig. 11. FHB behavior of system (18) for $N = 100$ cells.

Table 9
Simulation data for system (18).

Test case	$N = 100$	$N = 500$	$N = 1000$
Number of equations	300	1500	3000
Sparsity index	0.8833	0.8867	0.8887
CPU time	≈4 s	≈65 s	≈280 s

Table 10
Comparison of the number of equations of the reduced systems for the original system (18) in FHB behavior.

Number of cells	POD	APOBD
100	3	2+2+2
500	3	2+2+2
1000	3	2+2+2

3.3.2. Numerical results

In the following study, we will consider various instances for the number of cells $N = 100, 500, 1000$. In every test case, we will let the original system (18) compute from $T = 0$ to $T = 400$ via a fourth order Runge–Kutta solver with fixed time step of $h = 10^{-2}$. And we perform both reduction methods with the data from $T = 200$ to $T = 400$ as we need that the system is almost at an asymptotic behavior, this will be enough to capture the main behavior of the system as shown in Fig. 11.

In the construction of reduced systems, the tolerance criteria for the collinearity analysis in the APOBD method is set at 0.2, that gives us three blocks in the behavior of the study. The tolerance criteria of the selection of the reduced dimension in POD methods is set to 10^{-6} for every behavior, and nonlinear parts of (18) are addressed by means of the EIM with a tolerance of 10^{-6}.

Table 9 summarizes the simulation data of the original system (18) for the test case.

Now, we compare the number of equations, the sparsity index, and the speed-ups, from $T = 400$ to $T = 800$, for the original model and the reduced ones. The results are summarized in Table 10, which shows the number of variables of each cell number case, and Figs. 12(a)–12(c), which represent the evolution of sparsity, speed-up and error, respectively.

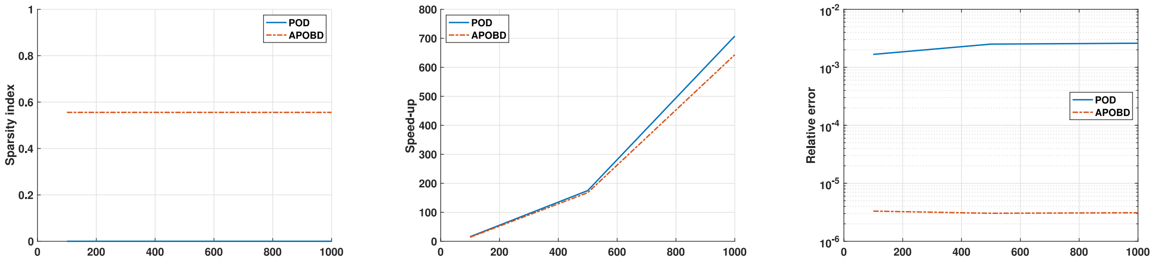
From Table 10, we can see again that the reduction is stronger when the number of original equations rises, as the number of reduced equations remains the same.

We can again see, that although the speed-up is better for the POD method (see Fig. 12(b)), APOBD provides three additional orders in the relative error, Fig. 12(c). So, for almost the same computational time, we obtain a much more precise approximation of the solution of the original problem (18).

For illustration purposes, in Fig. 13 we show the first basis obtained for each method and each block in the studied behavior, in the three basis for POD (Fig. 13(a)), where we cannot see a separation in the timescales, and in the basis for APOBD (Fig. 13(b)), where this separation is clearer, thanks to the higher sparsity index.

4. Conclusions and perspectives

In this work, we have developed new ROM techniques based on the POD method for dynamical systems with multiple timescales. They are able to maintain the structure of the original model, while producing a competitive speed-up. In this case, we have

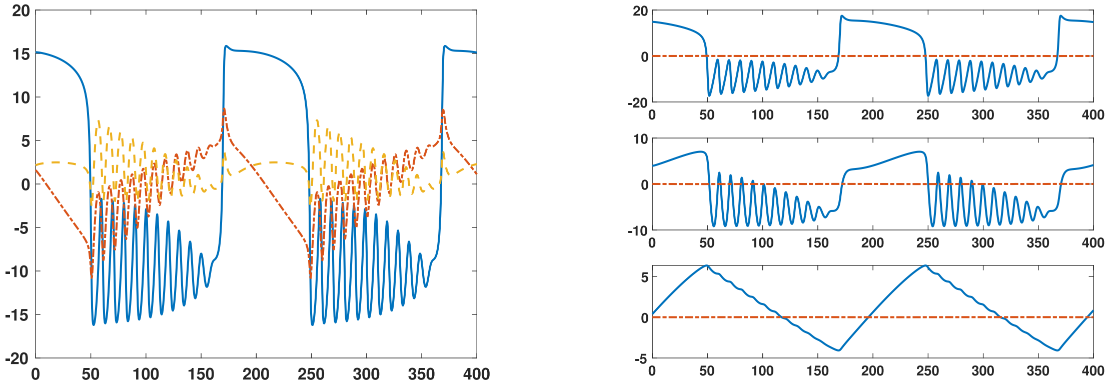


(a) Comparison of the sparsity index for each method in FHB behavior.

(b) Comparison of the speed-up for each method in FHB behavior.

(c) Comparison of the relative error for each method in FHB behavior.

Fig. 12. Comparisons of the results for each method in FHB behavior of system (18).



(a) First basis for the POD method for $N = 100$ cells in FHB.

(b) First basis for each block of the APOBD method for $N = 100$ cells in FHB.

Fig. 13. Comparisons of the basis for each method obtained in the studied behavior. In all the cases, the first basis is in blue continuous fashion, the second in red dash-dotted fashion and the third basis in yellow dashed fashion, this one only for the POD method. (a) POD. (b) APOBD.

performed an a priori analysis of how the variables of the original model could be grouped and then apply a data-driven analysis based on the correlation between these groups of variables in order to reduce the number of blocks to consider, while maintaining the most part of the structure of the original problem. We have also provided an error estimator of the information missed when performing these methods, depending on the number of considered blocks.

Furthermore, we have tested these methods for several biological neural models with multiple timescales that exhibit a rich variety of different qualitative behaviors. These new methods have been proven to produce an accurate approximation of the original solution in much less computational time, and we obtain results in speed-up and error that are better, or at least, as good as, the original POD procedure to build the reduced models. These new methods also provide a new feature, they keep the structure of the original model, which could be useful in the sense that we can see the different timescales in the new basis of the reduced model and this allows us to exploit the separation of the timescales. Furthermore, we have studied the evolution of the performance of these methods augmenting the number of cells in the original models, and they seem to perform better the more cells are considered. We have also noticed that these new techniques seem to perform better when there are more fast variables than slow variables and where there is a switch between the fast and slow behaviors.

Therefore, these new techniques based on the POD and a separation of variables to keep the structure of the original model are a useful tool for numerical study of network models, as they are able to reproduce complex behaviors that these models may exhibit.

In future works, we aim to apply these new ROM techniques considering various biological parameters; we also intend to apply them to perform a numerical analysis of the repartition of the behaviors of the models presented, as well as to apply these new techniques to different problems arising from other fields, like physical problems and computational fluid dynamics problems. Furthermore, we intend to study how these reductions can be performed in the transitions between different behaviors.

CRediT authorship contribution statement

A. Bandera: Writing – original draft, Writing – review & editing. **S. Fernández-García:** Writing – original draft, Writing – review & editing. **M. Gómez-Mármol:** Writing – original draft, Writing – review & editing. **A. Vidal:** Writing – original draft, Writing – review & editing.

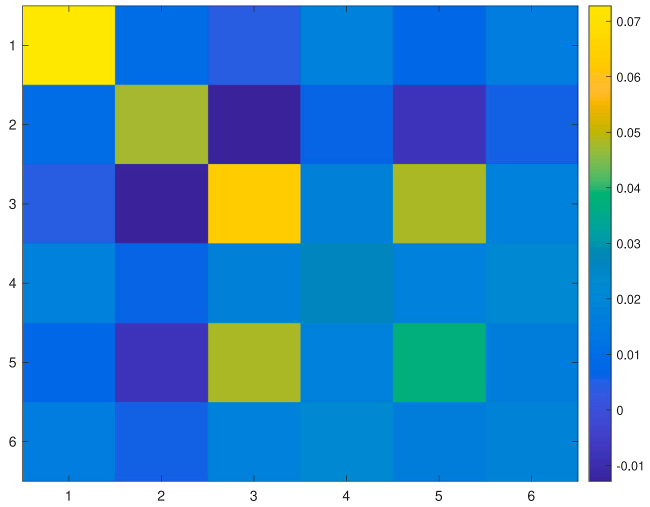


Fig. 14. Matrix $\mathbb{L}$ arising after computing the mean correlation of each block of variables of system (13) for $N = 100$ with $c_\alpha = -0.5$ and $c_\beta \in [-0.25, 0.25]$.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

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

Acknowledgment

This work has been supported by the Spanish Government Project PID2021-123153OB-C21.

Appendix. Limitations of the reduced order modeling techniques

The methods presented in the previous sections have been proven to provide competitive results in terms of maintaining the structure of the original problem, studied via the sparsity index, providing a reduction in the computational time, studied via the speed-up and being true to the original solution, studied via the relative error. However, there are some limitations to these techniques, both older and novel. Here, we discuss the limitations of the methods presented in this article.

A.1. Limitations in the network model of intracellular calcium concentration

As said in the presentation of the APOBD method, we could check if the block were well located, just checking if the elements of the diagonal of the matrix were close to one. A natural question arises: What happens in the case where there is no correlation inside each block of variables? In order to illustrate the results in this case, we consider system (13) for $N = 100$ with $c_\alpha = -0.5$, that is, inhibitory coupling within each cluster, and $c_\beta \in [-0.25, 0.25]$.

For the study, we consider the same parameters test as the ones used for behavior AS, and we compare POD, POBD3, POBD6, APOBD. The APOBD pre-analysis provides the matrix $\mathbb{L}$ shown in Fig. 14. It is easy to see that the elements of the diagonal are far from zero, so the POBD methods and the APOBD method, would not provide a faithful approximation. We checked that this is also the case for the POD method.

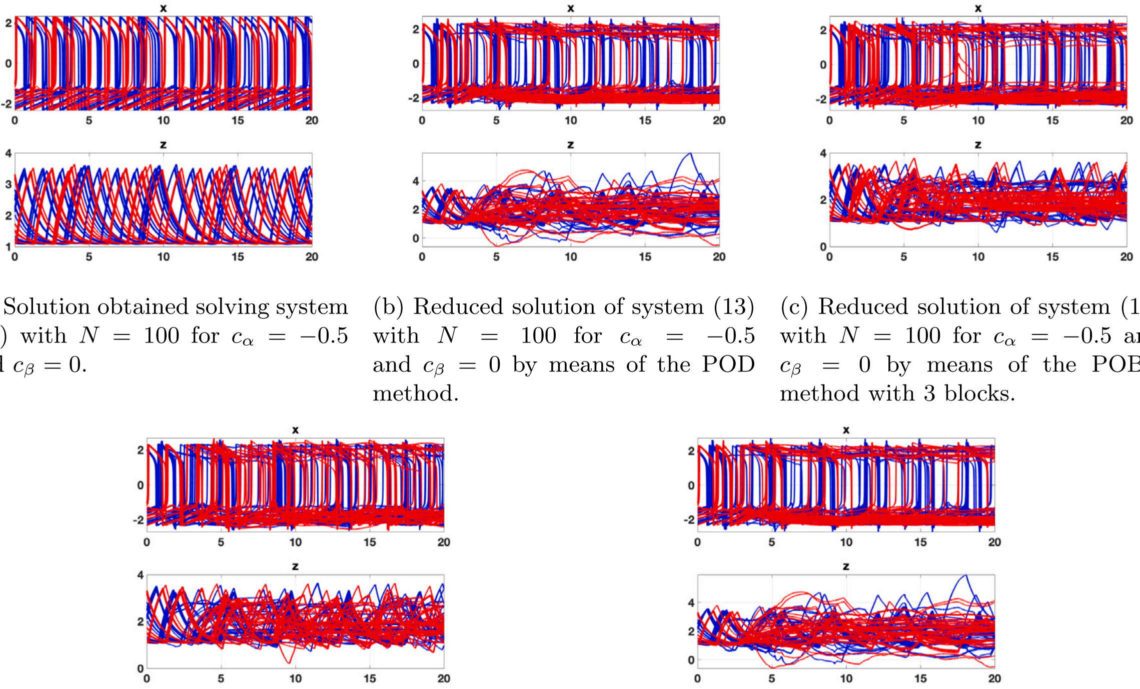
Table 11 shows the computational data obtained for the reductions. We can clearly see that the number of equations of the reduced models are higher than the ones obtained for the behaviors studied in the previous, as well no speed-up is obtained and the errors are not assumable.

Fig. 15 shows a comparison of the solutions obtained for the methods presented above. It is clear that the reduced solutions do not provide a good approximation of the behavior studied, and that the reduced behavior is very erratic.

This loss in accuracy could be due to the fact that the original system (13) exhibits chaotic-like behavior for this range of parameters, where each cell is independent of the others and its behavior cannot be expressed as a linear combination of the behaviors of the other cells for the whole length of the integrating range.

Table 11
Comparison of the results obtained, considering system (13) for $N = 100$ with $c_\alpha = -0.5$ and $c_\beta \in [-0.25, 0.25]$.

	POD	POBD3	POBD6	APOBD
Number of equations	109	73+33+31	37+36+19+16+17+15	109
Sparsity index	0	0.6797	0.7676	0
Speed-ups	<1	<1	<1	<1
Relative error	0.96	0.93	0.90	0.96



(a) Solution obtained solving system (13) with $N = 100$ for $c_\alpha = -0.5$ and $c_\beta = 0$.

(b) Reduced solution of system (13) with $N = 100$ for $c_\alpha = -0.5$ and $c_\beta = 0$ by means of the POD method.

(c) Reduced solution of system (13) with $N = 100$ for $c_\alpha = -0.5$ and $c_\beta = 0$ by means of the POBD method with 3 blocks.

(d) Reduced solution of system (13) with $N = 100$ for $c_\alpha = -0.5$ and $c_\beta = 0$ by means of the POBD method with 6 blocks.

(e) Reduced solution of system (13) with $N = 100$ for $c_\alpha = -0.5$ and $c_\beta = 0$ by means of the APOBD method.

Fig. 15. Representation of the solutions obtained for the original system and the reduced ones. (a) Original, (b) POD, (c) POBD with 3 blocks, (d) POBD with 6 blocks and (e) APOBD.

A.2. Limitations in the case of nonlinear coupling

Another limitation could be faced if the original model exhibit too many nonlinearities. In that case, the EIM could not be able to reproduce all the variability of the nonlinear terms, and we should need to go back to the original space to compute the nonlinearities. In the following example, we present a modified version of model (9) with nonlinear coupling:

$$O_i \begin{cases} \dot{x}_i = \tau(-y_i + f(x_i) - \phi_f(z_i)), \\ \dot{y}_i = \tau\epsilon k_i(a_0 x_i + a_1 y_i + a_2 + \dfrac{1}{N/2}\displaystyle\sum_{j=1}^{N} c_{ij}\sigma(x_i - x_j)), \\ \dot{z}_i = \tau\epsilon\left(\phi_r(x_i) - \dfrac{z_i - z_b}{\tau_z}\right), \end{cases} \tag{19}$$

with the same parameters as for (9), and $\sigma(x) = \frac{2}{1+\exp(-2x)} - 1$.

The selection of the nonlinear coupling has been made so we keep qualitatively similar behaviors as for the original model (9). In particular, we will be focusing on the AiP behavior with $c_\beta = -0.25$.

As done for the previous test cases, we present in Table 12 the number of equations for the reduced problem obtained with each method. Furthermore, in Fig. 16, we represent the evolution of sparsity, speed-up and error, respectively.

Table 12
Comparison of the number of equations of the reduced systems for the original system (19) in AiP behavior.

Number of cells	POD	APOBD
100	47	19+16+13+12
500	55	21+22+13+13
1000	58	23+23+14+14
2000	60	23+24+14+14

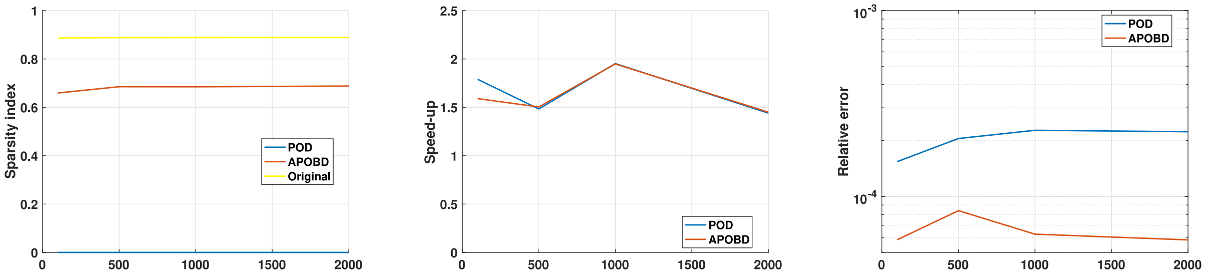
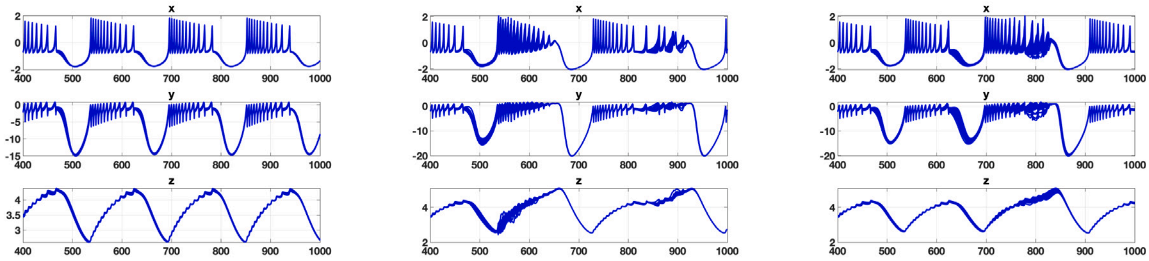


Fig. 16. Comparisons of the results obtained for each method in the AiP behavior of model (19). The left panel shows the sparsity index, the central panel shows the speed-ups, and the right panel shows the relative error.



(a) Solution for the original system (15) for $N = 500$ cells in (SW) behavior.

(b) Solution for the POD reduced system (15) for $N = 500$ cells in (SW) behavior.

(c) Solution for the APOBD reduced system (15) for $N = 500$ cells in (SW) behavior.

Fig. 17. Comparisons of the solutions obtained in the (SW) behavior. (a) Original. (b) POD. (c) APOBD with 3 blocks.

We can see that although that we obtain better results in terms of the sparsity index and relative error in comparison with the POD method, the speed-up is similar to the POD, but very low in comparison with the previous tests. This is due to the bottleneck of computing the nonlinear coupling in the original space.

A.3. Limitations in the Hindmarsh–Rose network model

In the case of the Hindmarsh–Rose Network model, we have a limitation when trying to reduce the system in the Square-Wave Bursting (SW) behavior ($I = 3$, $b = 2.6$, $\varepsilon = 0.01$), see Fig. 17(a). In this type of bursters, the stable equilibrium corresponding to the resting state disappears through a fold bifurcation and the limit cycle attractor corresponding to the spiking state disappears through a Homoclinic bifurcation. The presence of this Homoclinic bifurcation makes the system to be near to chaotic behavior.

In this case, the reduced models are not able to truly reproduce the behavior of the original system (14) as they exhibit some erratic behavior after a certain amount of time, see Figs. 17(b) and 17(c), due to the presence of the Homoclinic bifurcation.

References

[1] Ton R, Deco G, Daffertshofer A. Structure-function discrepancy: inhomogeneity and delays in synchronized neural networks. PLoS Comput Biol 2014;10(7):e1003736.

[2] Daffertshofer A, Pietras B. Phase synchronization in neural systems. Synergetics 2020;221–33.

[3] Coombes S. Next generation neural population models. Front Appl Math Stat 2023;9:1128224.

[4] Touboul J. Mean-field equations for stochastic firing-rate neural fields with delays: Derivation and noise-induced transitions. Physica D 2012;241(15):1223–44.

[5] Faugeras O, Grimbert F, Slotine J-J. Absolute stability and complete synchronization in a class of neural fields models. SIAM J Appl Math 2008;69(1):205–50.
[6] Byrne Á, Ross J, Nicks R, Coombes S. Mean-field models for EEG/MEG: from oscillations to waves. Brain Topogr. 2022;35(1):36–53.
[7] Traub RD, Kopell N, Bibbig A, Buhl EH, LeBeau FE, Whittington MA. Gap junctions between interneuron dendrites can enhance synchrony of gamma oscillations in distributed networks. J Neurosci 2001;21(23):9478–86.
[8] Semenov DM, Fradkov AL. Adaptive synchronization in the complex heterogeneous networks of Hindmarsh–Rose neurons. Chaos Solitons Fractals 2021;150:111170.
[9] Somers D, Kopell N. Waves and synchrony in networks of oscillators of relaxation and non-relaxation type. Physica D 1995;89(1–2):169–83.
[10] Drover J, Rubin J, Su J, Ermentrout B. Analysis of a canard mechanism by which excitatory synaptic coupling can synchronize neurons at low firing frequencies. SIAM J Appl Math 2004;65(1):69–92.
[11] Ermentrout B, Wechselberger M. Canards, clusters, and synchronization in a weakly coupled interneuron model. SIAM J Appl Dyn Syst 2009;8(1):253–78.
[12] Köksal Ersöz E, Desroches M, Krupa M. Synchronization of weakly coupled canard oscillators. Physica D 2017;349:46–61.
[13] Rotstein HG, Kuske R. Localized and asynchronous patterns via canards in coupled calcium oscillators. Physica D 2006;215(1):46–61.
[14] Zhabotinsky AM, Rotstein HG, Epstein IR, Kopell N. A canard mechanism for localization in systems of globally coupled oscillators. SIAM J Appl Math 2003;63(6):1998–2019.
[15] Best J, Borisyuk A, Rubin J, Terman D, Wechselberger M. The dynamic range of bursting in a model respiratory pacemaker network. SIAM J Appl Dyn Syst 2005;4(4):1107–39.
[16] Izhikevich EM. Synchronization of elliptic bursters. SIAM Rev 2001;43(2):315–44
[17] Coombes S, Wedgwood KC. Weakly coupled oscillator networks. In: Neurodynamics. Springer; 2023, p. 227–91.
[18] Chapelle D, Gariah A, Sainte-Marie J. Galerkin approximation with proper orthogonal decomposition: new error estimates and illustrative examples. ESAIM Math Model Numer Anal 2012;46(4):731–57.
[19] Freund RW. Krylov-subspace methods for reduced-order modeling in circuit simulation. J Comput Appl Math 2000;123(1–2):395–421.
[20] Freund RW. Model reduction methods based on krylov subspaces. Acta Numer 2003;12:267–319.
[21] Freund RW. The SPRIM algorithm for structure-preserving order reduction of general RCL circuits. Model Reduct Circuit Simul 2011;25–52.
[22] Kahlbacher M, Volkwein S. Galerkin proper orthogonal decomposition methods for parameter dependent elliptic systems. Discuss Math Differ Incl Control Optim 2007;27(1):95–117.
[23] Kerschen G, Golinval J-c, Vakakis AF, Bergman LA. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. Nonlinear Dyn 2005;41:147–69.
[24] Kostova-Vassilevska T, Oxberry GM. Model reduction of dynamical systems by proper orthogonal decomposition: Error bounds and comparison of methods using snapshots from the solution and the time derivatives. J Comput Appl Math 2018;330:553–73.
[25] Kunisch K, Volkwein S. Galerkin proper orthogonal decomposition methods for parabolic problems. Numer Math 2001;90:117–48.
[26] Rathinam M, Petzold LR. A new look at proper orthogonal decomposition. SIAM J Numer Anal 2003;41(5):1893–925.
[27] Volkwein S. Proper orthogonal decomposition: Theory and reduced-order modelling. Lecture notes, vol. 4, (no. 4):University of Konstanz; 2013, p. 1–29.
[28] Xu K-L, Jiang Y-L, Li Z, Li L. Model reduction of discrete time-delay systems based on Charlier polynomials and high-order krylov subspaces. Linear Algebra Appl 2023.
[29] Benner P, Gugercin S, Willcox K. A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev 2015;57(4):483–531.
[30] Kumar R, Ezhilarasi D. A state-of-the-art survey of model order reduction techniques for large-scale coupled dynamical systems. Int J Dyn Control 2023;900–16.
[31] Bandera A, Fernández-García S, Gómez-Mármol M, Vidal A. A multiple timescale network model of intracellular calcium concentrations in coupled neurons: Insights from ROM simulations. Math Model Nat Phenom 2022;17:11.
[32] Anderson S, White C, Farhat C. Space-local reduced-order bases for accelerating reduced-order models through sparsity. Internat J Numer Methods Engrg 2022.
[33] Barrio R, Shilnikov A. Parameter-sweeping techniques for temporal dynamics of neuronal systems: case study of Hindmarsh-Rose model. J Math Neurosci 2011;1(1):1–22.
[34] Bonaventura L, Fernández-García S, Gómez-Mármol M. Efficient implicit solvers for models of neuronal networks. 2022, arXiv preprint arXiv:2210.01697.
[35] Etémé AS, Tabi CB, Mohamadou A. Long-range patterns in Hindmarsh-Rose networks. Commun Nonlinear Sci Numer Simul 2017;43:211–9.
[36] Hindmarsh JL, Rose R. A model of neuronal bursting using three coupled first order differential equations. Proc R Soc Lond Ser B Biol Sci 1984;221(1222):87–102.
[37] De Vries G. Multiple bifurcations in a polynomial model of bursting oscillations. J Nonlinear Sci 1998;8:281–316.
[38] Izhikevich EM. Neural excitability, spiking and bursting. Int J Bifurcation Chaos 2000;10(06):1171–266.
[39] Pernarowski M. Fast subsystem bifurcations in a slowly varying liénard system exhibiting bursting. SIAM J Appl Math 1994;54(3):814–32.
[40] Smolen P, Terman D, Rinzel J. Properties of a bursting model with two slow inhibitory variables. SIAM J Appl Math 1993;53(3):861–92.
[41] Quarteroni A, Manzoni A, Negri F. Reduced basis methods for partial differential equations. Springer; 2016.
[42] Fernández-García S, Vidal A. Symmetric coupling of multiple timescale systems with mixed-mode oscillations and synchronization. Physica D 2020;401:132129.
[43] Pernarowski M, Miura R, Kevorkian J. Perturbation techniques for models of bursting electrical activity in pancreatic β-cells. SIAM J Appl Math 1992;52(6):1627–50.
[44] Fallani FDV, Corazzol M, Sternberg JR, Wyart C, Chavez M. Hierarchy of neural organization in the embryonic spinal cord: Granger-causality graph analysis of in vivo calcium imaging data. IEEE Trans Neural Syst Rehabil Eng 2015;23(3):333–41.
[45] Barrault M, Maday Y, Nguyen NC, Patera AT. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. C R Math 2004;339(9):667–72.
[46] Grepl MA, Maday Y, Nguyen NC, Patera AT. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. ESAIM Math Model Numer Anal 2007;41(3):575–605.
[47] Shilnikov A, Kolomiets M. Methods of the qualitative theory for the Hindmarsh–Rose model: A case study–a tutorial. Int J Bifur Chaos 2008;18(08):2141–68.
[48] Bertram R, Butte MJ, Kiemel T, Sherman A. Topological and phenomenological classification of bursting oscillations. Bull Math Biol 1995;57(3):413–39.
[49] Barrio R, Ibáñez S, Pérez L. Hindmarsh–Rose model: Close and far to the singular limit. Phys Lett A 2017;381(6):597–603.
[50] Holden L, Erneux T. Slow passage through a hopf bifurcation: from oscillatory to steady state solutions. SIAM J Appl Math 1993;53(4):1045–58.
[51] Holden L, Erneux T. Understanding bursting oscillations as periodic slow passages through bifurcation and limit points. J Math Biol 1993;31:351–65.