Application of POD-DEIM Approach for Dimension Reduction of a Diffusive Predator-Prey System with Allee effect

Gabriel Dimitriu[1], Ionel M. Navon[2] and Răzvan Ştefănescu[2]

[1] The "Grigore T. Popa" University of Medicine and Pharmacy,
Department of Mathematics and Informatics, dimitriu.gabriel@gmail.com,
700115 Iaşi, Romania
[2] The Florida State University, Department of Scientific Computing,
inavon@fsu.edu, rstefanescu@fsu.edu,
Tallahassee, FL 32306, USA

Abstract. In this work we carry out an application of DEIM combined with POD to provide dimension reduction of a system of two nonlinear partial differential equations describing the spatio-temporal dynamics of a predator-prey community, where the prey *per capita* growth rate is damped by the Allee effect. DEIM improves the efficiency of the POD approximation reducing the computational complexity of the nonlinear term and regains the full model reduction expected from the POD model. Numerical results show that the dynamics of the predator-prey model in the full-order system of dimension 2048 can be captured accurately by the POD-DEIM reduced system of dimension 5 with the computational time reduced by a factor of $\mathcal{O}(10^4)$.

1 Introduction

Proper Orthogonal Decomposition (POD) is probably the mostly used and most successful model reduction technique, where the basis functions contain information from the solutions of the dynamical system at pre-specified time-instances, so-called *snapshots*. Due to a possible linear dependence or almost linear dependence, the snapshots themselves are not appropriate as a basis. Hence a singular value decomposition is carried out and the leading generalized eigenfunctions are chosen as a basis, referred to as the POD basis.

Unfortunately, for nonlinear PDEs, the efficiency in solving the reduced-order systems constructed from standard Galerkin projection with any reduced globally supported basis set, including the one from POD, is limited to the linear or bilinear part, both for finite volume or finite difference schemes since nonlinear terms still require calculation on the full dimensional model. For finite element schemes, DEIM reduces the CPU complexity of the POD models for nonlinear terms other then quadratics. In the case of quadratic nonlinearities a so-called precomputed POD technique achieves the same level of reduction as in the case of linear terms.

A considerable reduction in complexity is achieved by DEIM – a discrete variation of Empirical Interpolation Method (EIM), proposed by Barrault, Maday, Nguyen and Patera in [2]. According to this method, the evaluation of the approximate nonlinear term does not require a prolongation of the reduced state variables back to the original high dimensional state approximation required to evaluate the nonlinearity in the POD approximation.

In this study we carry out an application of DEIM combined with POD to provide dimension reduction of a system of two nonlinear partial differential equations describing the spatio-temporal dynamics of a predator-prey community, where the prey *per capita* growth rate is damped by the Allee effect. This model was introduced and analyzed in an infinite space by Petrovskii et al. ([11]), together with properties of the solution and biologically significant dependence on the parameter values.

2 The Predator-Prey Model with Allee Effect

The spatio-temporal dynamics of a predator-prey system can be described by the equations ([10]):

$$\frac{\partial U(X,T)}{\partial T} = D\frac{\partial^2 U}{\partial X^2} + f(U)U - r(U)V, \tag{1}$$

$$\frac{\partial V(X,T)}{\partial T} = D\frac{\partial^2 V}{\partial X^2} + \kappa r(U)V - g(V)V, \tag{2}$$

where U and V are the densities of prey and predator, respectively, at position X and time T. The function $f(U)$ is the *per capita* growth rate of the prey and the term $r(U)V$ stands for predation. κ is the coefficient of food utilization, and $g(V)$ is the *per capita* mortality rate of predator. Here, the first term on the right-hand side of equations (1) and (2) describes the spatial mixing caused either by self-motion of individuals ([12]) or by properties of the environment, for example, for plankton communities the mixing is attributed to turbulent diffusion ([7]). D is the diffusion coefficient, which we assume to be the same for both prey and predator.

For different species, functions f, r and g can represent different functional responses (logistic, Gompertz, Holling, etc.). We assume that the prey dynamics is subjected to the Allee effect ([1], [4], [9]), so that its *per capita* growth rate is not a monotonically decreasing function of the prey density, but possesses a local maximum. In this model, the standard parametrization ([8]) is defined by

$$f(U) = \alpha(U - U_0)(K - U),$$

where K denotes the prey carrying capacity and U_0 is a certain measure of the Allee effect. Regarding the *per capita* predator mortality, one assumes that it is described by the following function:

$$g(v) = M + d_0 V^2$$

where M and d_0 are positive parameters. Function $g(V)$ gives the so-called *closure term* because it is supposed not only to describe the process taking place inside the predator population (such as natural mortality, competition, possibly cannibalism, etc.) but also, virtually to take into account the impact of higher predators that are not included into the model explicitly. We assume that the predator shows a linear response to prey according to the classical Lotka-Volterra model, that is, $r(U) = \mu U$. Then, equations (1)-(2) take the form

$$\frac{\partial U(X,T)}{\partial T} = D\frac{\partial^2 U}{\partial X^2} + \alpha U(U - U_0)(K - U) - \mu UV, \tag{3}$$

$$\frac{\partial V(X,T)}{\partial T} = D\frac{\partial^2 V}{\partial X^2} + \kappa\mu UV - MV - d_0 V^3. \tag{4}$$

A common procedure for solving the system of equations (3)-(4) is to first nondimensionalize the system, and then obtaining the numerical solution by employing a discretization scheme. Define the nondimensional variables and parameters to be:

$$u = \frac{U}{K}, \quad v = \frac{\eta V}{\alpha K^2}, \quad x = X\sqrt{\frac{\alpha K^2}{D}}, \quad t = T\alpha K^2.$$

From equations (3) and (4) one obtains

$$u_t = u_{xx} - \beta u + (\beta + 1)u^2 - u^3 - uv, \tag{5}$$

$$v_t = v_{xx} + kuv - mv - \delta v^3, \tag{6}$$

where $\beta = U_0 K^{-1}$, $k = \kappa\eta(\alpha K)^{-1}$, $m = M(\alpha K^2)^{-1}$ and $\delta = d_0\alpha K^2\eta^{-2}$ are positive dimensionless parameters, subscripts x and t stand for the partial derivatives with respect to dimensionless space and time, respectively. Here we consider equations (5) and (6) in a bounded domain Ω with homogeneous Dirichlet boundary conditions. The initial conditions given by $u(x, 0) = u_0(x)$ and $v(x, 0) = v_0(x)$ will be specified in Section 4.

3 The POD and POD-DEIM Reduced Order System

In this section we provide some details for constructing the reduced-order system of the full-order system (5)–(6) applying Proper Orthogonal Decomposition (POD) and Discrete Empirical Interpolation Method (DEIM).

POD is an efficient method for extracting orthonormal basis elements that contain characteristics of the space of expected solutions which is defined as the span of the snapshots ([5], [6]). In this framework, *snapshots* are the sampled (numerical) solutions at particular time steps or at particular parameter values. POD gives an optimal set of basis vectors minimizing the mean square error from approximating these snapshots. In this finite dimensional setting, POD is in fact just the singular value decomposition (SVD).

The projected nonlinearities in equations (5)–(6) are approximated by DEIM in the form that enables precomputation, so that evaluating the approximate nonlinear terms using DEIM does not require a prolongation of the reduced state variables back to the original high dimensional state approximation, as it is required for nonlinearity evaluation in the original POD approximation. Only a few entries of the original nonlinear term, corresponding to the specially selected interpolation indices from DEIM must be evaluated at each time step ([2], [3], [13]). We give formally the DEIM approximation in Definition 1, and the procedure for selecting DEIM indices is shown in Algorithm DEIM. Each DEIM index is selected to limit growth of a global error bound using a greedy technique relating the DEIM approximation to the full optimal POD approximation ([3]).

Definition 1. Let $\mathbf{f} : \mathcal{D} \mapsto I\!R^n$ be a nonlinear vector-valued function with $\mathcal{D} \subset I\!R^d$, for some positive integer d. Let $\{\mathbf{u}\}_{\ell=1}^m \subset I\!R^n$ be a linearly independent set, for $m = 1, \ldots, n$. For $\tau \in \mathcal{D}$, the DEIM approximation of order m for $\mathbf{f}(\tau)$ in the space spanned by $\{\mathbf{u}\}_{\ell=1}^m$ is given by

$$\widehat{\mathbf{f}}(\tau) := \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(\tau), \tag{7}$$

where basis $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m] \in I\!R^{n\times m}$ can be constructed effectively by applying the POD method on the nonlinear snapshots $f(\tau_i)$, $\tau_i \in \mathcal{D}$ and $\mathbf{P} = [\mathbf{e}_{\varrho_1}, \ldots, \mathbf{e}_{\varrho_m}] \in I\!R^{n\times m}$ with $\{\varrho_1, \ldots, \varrho_m\}$ being the output from Algorithm DEIM with the input basis $\{\mathbf{u}_i\}_{i=1}^m$.

Algorithm DEIM:
INPUT: $\{\mathbf{u}\}_{\ell=1}^m \subset I\!R^n$ linearly independent
OUTPUT: $\boldsymbol{\varrho} = [\varrho_1, \ldots, \varrho_m]^T \in I\!R^m$

1. $[|\rho|\ \varrho_1] = \max\{|\mathbf{u}_1|\}$
2. $\mathbf{U} = [\mathbf{u}_1],\ \mathbf{P} = [\mathbf{e}_{\varrho_1}],\ \boldsymbol{\varrho} = [\varrho_1]$
3. **for** $\ell = 2$ to m **do**
4. Solve $(\mathbf{P}^T\mathbf{U})\mathbf{c} = \mathbf{P}^T\mathbf{u}_\ell$ for $\mathbf{c}$
5. $\mathbf{r} = \mathbf{u}_\ell - \mathbf{U}\mathbf{c}$
6. $[|\rho|\ \varrho_\ell] = \max\{|\mathbf{r}|\}$
7. $\mathbf{U} \leftarrow [\mathbf{U}\ \mathbf{u}_\ell]$, $\mathbf{P} \leftarrow [\mathbf{P}\ \mathbf{e}_{\varrho_\ell}]$, $\boldsymbol{\varrho} \leftarrow \begin{bmatrix} \boldsymbol{\varrho} \\ \varrho_\ell \end{bmatrix}$
8. **end for**

The notation *max* in Algorithm DEIM is the same as the function *max* in Matlab. Thus, $[|\rho|\ \varrho_\ell] = \max\{|\mathbf{r}|\}$ implies $|\rho| = |r_{\varrho_\ell}| = \max_{i=1,\ldots,n}\{|r_i|\}$, with the smallest index taken in case of a tie. According to this algorithm, the DEIM procedure generates a set of indices inductively on the input basis in such a way that, at each iteration, the current selected index captures the maximum variation of the input basis vectors. The order of the input basis $\{\mathbf{u}\}_{\ell=1}^m$ according to the dominant singular values is important and an error analysis indicated that the POD basis is a suitable choice for $\mathbf{U}$ ([3]).

4 Numerical Results

We shall present three numerical experiments. The system (5)-(6) was solved numerically using a finite difference discretization. Let $0 = x_0 < x_1 < \cdots < x_n < x_{n+1} = 1$ be equally spaced points on x-axis for generating the grid points on the dimensionless domain $\Omega = [0, 1]$, and take time domain $[0, T] = [0, 1]$. The corresponding spatial finite difference discretized system of (5)-(6) becomes a system of nonlinear ODEs. The semi-implicit Euler scheme was used to solve the discretized system of full dimension, as well as POD and POD-DEIM reduced order systems.

Case 1. The parameters used here are $m = \beta = 4$, $\kappa = 15$, and $\delta = 0.25$. The initial conditions were set to $u_0(x) = \sin x \sin(\pi x) \exp(x)$, and $v_0(x) = x(1-x)^3$. The number of spatial inner grid points on the x-axis – which defines the dimension of the full-order system – was successively taken as 16, 32, 64, 128, ..., 2048. In Figure 1, the solutions for state variables (u and v) from POD and POD-DEIM reduced systems, with dimPOD=10 and dimDEIM=5, are depicted with the corresponding ones from the full-order system, as well as the corresponding average relative errors at the grid points. It shows that POD-DEIM reduces more than 400 times in dimension and reduces the computational time by a factor of $\mathcal{O}(10^4)$ as shown in Table 1. From Table 1, the CPU time used in computing POD reduced system clearly reflects the dependency on the dimension of the original full-order system. Table 1 also shows a significant improvement in computational time of the POD-DEIM reduced system from both POD reduced system and full-order system.

Case 2. The numerical results obtained in this case (see Table 2 and Figure 2) were generated with parameters: $m = \beta = 1.1$, $\kappa = 5$, and $\delta = 1$. We used the following initial conditions: $u_0(x) = 10x(1-x)(1 + 0.8 \sin(30x) \cos(10x))$, $v_0(x) = 10x(1-x)(1 + 0.8 \sin(10x) \cos(30x))$. In comparison with Case 1, here the densities of the species present initially large fluctuations along the whole space domain, damped very fast by the Allee effect.

Case 3. In this experiment we use the same initial conditions and values of the parameters as those indicated in Case 1. Here we performed the computations with dimPOD=45 and dimPOD-DEIM=90. The numerical results are contained in Table 3. We note that the POD-DEIM relative errors for both state variables, u and v, are 10 times smaller than those obtained in Case 1.

5 Conclusions

The model reduction technique combining POD with DEIM has been shown to be efficient for capturing the spatio-temporal dynamics of a diffusive predator-prey model with substantial reduction in both dimension and computational time. The failure to decrease complexity with the standard POD technique was clearly demonstrated by the comparative computational times shown in Tables 1–3. DEIM was shown to be very effective in overcoming the deficiencies of POD with respect to quadratic and cubic nonlinearities in the model under study. The strong Allee effect for prey leads to a very rich dynamics ([11]), travelling fronts of invasive species and sensitivity to parameter variations ([11], [14]).

In order to increase the efficiency of the POD-DEIM approximation, a possible extension is to incorporate the POD-DEIM approach with higher-order FD schemes to improve the overall accuracy.

Table 1. CPU time of full-order system, POD and POD-DEIM reduced systems with the corresponding average relative errors for u and v – Case 1.

Internal Nodes N	CPU Time Full Dim	CPU Time POD	CPU Time POD–DEIM	$Error^{rel}$ POD – u	$Error^{rel}$ POD – DEIM – u	$Error^{rel}$ POD – v	$Error^{rel}$ POD – DEIM – v
16	3.632462e-001	7.122489e-001	1.811141e-002	1.170196e-005	1.857876e-004	2.103251e-004	1.222462e-002
32	4.169362e-001	7.154559e-001	1.860040e-002	1.169715e-005	1.463095e-004	2.099999e-004	1.230968e-002
64	6.164471e-001	7.516708e-001	2.826825e-002	1.169587e-005	9.926064e-005	2.099146e-004	1.128253e-002
128	6.529374e-001	8.020902e-001	1.812896e-002	1.169554e-005	1.560804e-004	2.098927e-004	1.165304e-002
256	1.631008e+000	8.673314e-001	1.819947e-002	1.169545e-005	1.481253e-004	2.098871e-004	1.168835e-002
512	6.377997e+000	1.012015e+000	1.823390e-002	1.169543e-005	1.323507e-004	2.098857e-004	1.166019e-002
1024	2.924355e+001	1.291486e+000	1.827065e-002	1.169542e-005	1.330641e-004	2.098853e-004	1.172391e-002
2048	1.675980e+002	2.788567e+000	1.825973e-002	1.169542e-005	1.340120e-004	2.098852e-004	1.171443e-002

Table 2. CPU time of full-order system, POD and POD-DEIM reduced systems with the corresponding average relative errors for u and v – Case 2.

Internal Nodes N	CPU Time Full Dim	CPU Time POD	CPU Time POD–DEIM	$Error^{rel}$ POD – u	$Error^{rel}$ POD – DEIM – u	$Error^{rel}$ POD – v	$Error^{rel}$ POD – DEIM – v
16	3.553413e-001	6.865113e-001	1.793760e-002	2.356905e-005	1.580829e-005	3.047587e-004	4.552821e-004
32	4.284408e-001	6.980146e-001	1.802235e-002	2.360341e-005	2.358619e-005	3.067127e-004	2.008601e-004
64	4.867406e-001	7.455845e-001	1.855140e-002	2.360549e-005	2.207394e-005	3.070870e-004	7.456335e-004
128	6.408845e-001	7.897221e-001	1.834978e-002	2.360580e-005	3.493150e-005	3.071769e-004	4.430065e-004
256	1.098676e+000	8.647706e-001	1.822993e-002	2.360586e-005	3.485313e-005	3.071994e-004	5.934588e-004
512	3.885919e+000	1.006938e+000	1.858914e-002	2.360588e-005	3.497038e-005	3.072050e-004	6.145148e-004
1024	1.917511e+001	1.256279e+000	1.878149e-002	2.360588e-005	3.495507e-005	3.072064e-004	6.035787e-004
2048	1.148451e+002	1.883425e+000	1.870667e-002	2.360588e-005	3.507343e-005	3.072068e-004	5.992118e-004

Table 3. CPU time of full-order system, POD and POD-DEIM reduced systems with the corresponding average relative errors for u and v – Case 3.

Internal Nodes N	CPU Time Full Dim	CPU Time POD	CPU Time POD–DEIM	$Error^{rel}$ POD – u	$Error^{rel}$ POD – DEIM – u	$Error^{rel}$ POD – v	$Error^{rel}$ POD – DEIM – v
128	5.741809e-001	8.289299e-001	5.605313e-002	1.169554e-005	1.106775e-005	2.098927e-004	5.454479e-003
256	1.144514e+000	9.969025e-001	5.958563e-002	1.169545e-005	1.055126e-005	2.098871e-004	2.347417e-003
512	3.807256e+000	1.154277e+000	6.668969e-002	1.169543e-005	1.486580e-005	2.098857e-004	5.295462e-003
1024	1.886075e+001	1.411974e+000	6.400274e-002	1.169542e-005	1.264476e-005	2.098853e-004	5.813729e-003
2048	1.098164e+002	2.144956e+000	6.639338e-002	1.169542e-005	1.548229e-005	2.098852e-004	7.346586e-003

Acknowledgement. The authors acknowledge the generous help of Professor Danny C. Sorensen Noah Harding Professor of Computational and Applied Mathematics at Rice University, that provided us with a 1-D Matlab code of POD DEIM of the Burgers equation. Prof. G. Dimitriu acknowledges the support of the grant of the Romanian National Authority for Scientific Research, CNCS - UEFISCDI, project number PN-II-ID-PCE-2011-3-0563, contract no. 343/5.10.2011 “Models from medicine and biology: mathematical and numerical insights”. Prof. I.M. Navon and Dr. R. Ştefănescu acknowledge the partial support of NOAA grant NA10NES4400008.

References

1. Allee, W.C.: The social life of animals. New York: Norton and Co. (1938).
2. Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An “empirical interpolation” method: application to efficient reduced-basis discretization of partial differential equations. C.R. Math. Acad. Sci. Paris 339, 667–672 (2004).
3. Chaturantabut, S., Sorensen, D.C.: Discrete Empirical Interpolation for Nonlinear Model Reduction. TR09–05, CAAM, Rice University (2009).
4. Dennis, B.: Allee effects: population growth, critical density, and the chance of extinction. Nat. Res. Model. 3, 481–538 (1989).
5. Dimitriu, G., Apreutesei, N.: Comparative study with data assimilation experiments using proper orthogonal decomposition method. Lecture Notes in Computer Science 4818, 393–400 (2008).
6. Dimitriu, G., Apreutesei, N., Stefanescu, R.: Numerical simulations with data assimilation using an adaptive POD procedure. Lecture Notes in Computer Science 5910, 165–172 (2010).
7. Dubois, D.: A model of patchiness for prey-predator plankton populations. Ecol. Model. 1, 67–80 (1975).
8. Lewis, M.A., Kareiva, P.: Allee dynamics and the spread of invading organisms. Theor. Popul. Biol. 43, 141–158 (1993).
9. Morozov, A.Y., Petrovskii, S.V., Li, B.-L.: Bifurcations and chaos in a predator-prey system with the Allee effect. Proc. R. Soc. B 271, 1407–1414 (2004).
10. Murray, J.D.: Mathematical biology, 2nd edn. Berlin, Springer (1993).
11. Petrovskii, S.V., Malchow, H., Li, B.-L.: An exact solution of a diffusive predator-prey system. Proc. R. Soc. A 461, 1029–1053 (2005).
12. Skellam, J.G.: Random dispersal in theoretical populations. Biometrika 38, 196–218 (1951).
13. Ştefănescu, R., Navon, I.M.: POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model. Journal of Computational Physics 237, 95–114 (2013).
14. Volpert, V., Petrovskii, S.: Reaction-diffusion waves in biology. Physics of Live Reviews 6, 267–310 (2009).

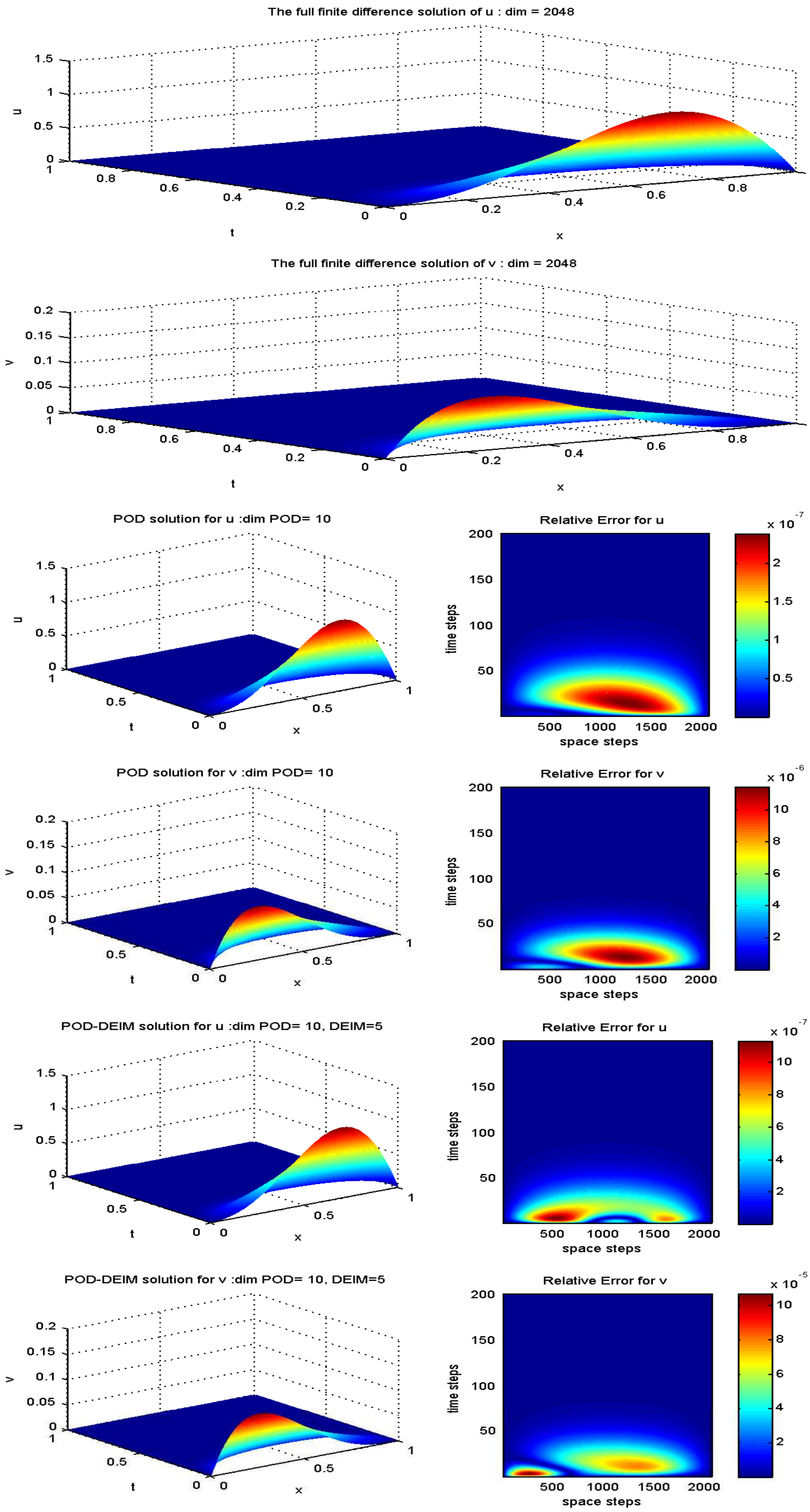

Fig. 1. Solution plots of the model from the full-order system of dimension 2048, from POD reduced system (dimPOD=10), and from POD-DEIM reduced system (dimPOD=10, dimDEIM=5), with the corresponding average relative errors at the inner grid points – Case 1.

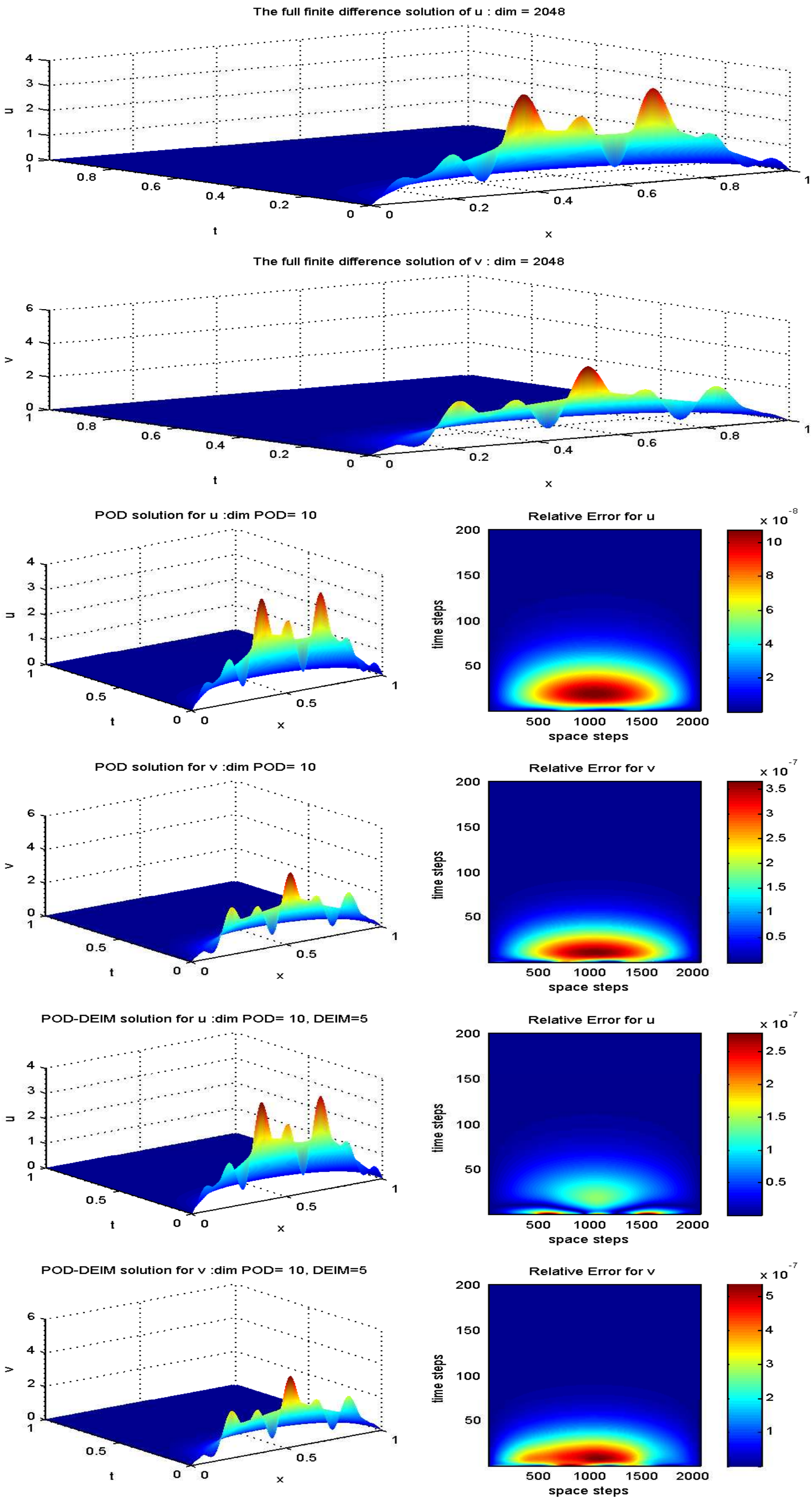

Fig. 2. Solution plots of the model from the full-order system of dimension 2048, from POD reduced system (dimPOD=10), and from POD-DEIM reduced system (dimPOD=10, dimDEIM=5), with the corresponding average relative errors at the inner grid points – Case 2.