MODEL REDUCTION BY PROPER ORTHOGONAL DECOMPOSITION FOR LAMBDA-OMEGA SYSTEMS

H. Müller* and S. Volkwein†

*University of Graz
Institute for Mathematics and Scientific Computing
Heinrichstraße 36
8010 Graz, Austria
e-mail: hannes.mueller@stud.uni-graz.at

†University of Graz
Institute for Mathematics and Scientific Computing
Heinrichstraße 36
8010 Graz, Austria
e-mail: stefan.volkwein@uni-graz.at
web page: http://www.uni-graz.at/imawww/volkwein/

Key words: Model reduction, proper orthogonal decomposition, reaction-diffusion systems, lambda-omega systems, chemical turbulence.

Abstract. Proper orthogonal decomposition (POD) is a powerful technique for model reduction of linear and non-linear systems. It is based on a Galerkin type discretization with basis elements created from the system itself. In this work POD is used to derive low-order models for a so-called lambda-omega (λ-ω) system that is a universal model to investigate two-species reaction-diffusion problems. In the case of fast reaction kinetics and small diffusion, these systems evolve to turbulent behavior. The performance of the POD model reduction is studied in dependence on the parameters of the λ-ω system. With increasing turbulence more POD modes are needed to capture the dynamics of the full system in a satisfactory way.

1 INTRODUCTION

Reaction diffusion systems describe processes that can be often found in nature. In particular, lambda-omega (λ-ω) systems can be used to study two species reaction-diffusion systems. They are used as a ‘universal model’ to study chemical reaction processes [19], to describe the dynamics of biological systems [24], to investigate the mechanism of pattern formation [7], and to analyze the occurrence of turbulent behavior [23]. In [8] a technical application based on pattern forming systems is considered. Optimal control problems for the λ-ω system are considered in [5, 6].

The amount of computing time and space needed for the solution of such a dynamical system can be very large. Due to the importance of possible applications related to
systems we apply a model reduction technique to derive a low-dimensional approximation. Here we utilize proper orthogonal decomposition (POD) that is a method to derive reduced-order models for systems of differential equations. It is based on projecting the system onto subspaces consisting of basis elements that contain characteristics of the expected solution. This is in contrast to, e.g., finite element techniques, where the elements of the subspaces are uncorrelated to the physical properties of the system that they approximate. Dependent on the parameters of the $\lambda$-$\omega$ system it shows different turbulent behavior. With increasing turbulence more POD modes are needed to capture the dynamics of the full system.

Let us finally briefly comment on further literature containing applications of POD. It is successfully used in different fields including signal analysis and pattern recognition (see, e.g., [10]), fluid dynamics and coherent structures (see, e.g., [12, 28]) and more recently in control theory (see, e.g., [2, 4, 18, 22]) and inverse problems [3]. Surprisingly good approximation properties are reported for POD based schemes in several articles, see [9, 30], for example. The relationship between POD and balancing is considered in [21, 26, 31]. Error analysis for nonlinear dynamical systems in finite dimensions are carried out in [13, 25]. In [16, 17] error estimates for POD Galerkin approximations are derived for non-linear parabolic differential equations. These results are extended to linear-quadratic optimal control problems in [11]. In [14] error estimates for POD Galerkin schemes for linear and certain semi-linear elliptic, parameter dependent systems are proved. Parameter estimation problems for coefficients in elliptic partial differential equations are solved numerically by POD approximations in [15].

The article is organized in the following manner: In Section 2 we introduce the $\lambda$-$\omega$ system and review some of its properties. The POD method and the reduced-order modeling is described in Section 3. Section 4 is devoted to present some numerical experiments including a predictor-corrector method, where we compute the POD solution by splitting the POD modes in lower and higher modes as usual done in non-linear Galerkin projections. In the last section we draw some conclusions.

2 LAMBDA-OMEGA SYSTEMS

Let $\Omega = (0, 1) \times (0, 1)$ be the spatial domain with Lipschitz-boundary $\Gamma = \partial \Omega$. By $n \in \mathbb{R}^2$ we denote the outward normal vector on $\Gamma$. For $T > 0$ we set $Q = (0, T) \times \Omega$ and $\Sigma = (0, T) \times \Omega$. For given diffusion parameter $\sigma > 0$ the functions $u, v : \overline{Q} \to \mathbb{R}$ solve

\[
\begin{pmatrix}
    u_t(t, x) \\
    v_t(t, x)
\end{pmatrix} = \begin{pmatrix}
    \lambda(s) & -\omega(s) \\
    \omega(s) & \lambda(s)
\end{pmatrix} \begin{pmatrix}
    u(t, x) \\
    v(t, x)
\end{pmatrix} + \begin{pmatrix}
    \sigma \Delta u(t, x) \\
    \sigma \Delta v(t, x)
\end{pmatrix}
\]

for all $(t, x) \in Q$, (1a)

where $s = u(t, x)^2 + v(t, x)^2$ holds and $x = (x, y)$. We supply (1a) with homogeneous Dirichlet boundary conditions

\[
u(t, s) = v(t, s) = 0 \quad \text{for all } (t, s) \in \Sigma
\]

(1b)
or homogeneous Neumann boundary conditions

\[ \sigma \frac{\partial u}{\partial n}(t,s) = \sigma \frac{\partial v}{\partial n}(t,s) = 0 \quad \text{for all } (t,s) \in \Sigma \]  

(1c)

and with the initial conditions

\[ u(0,x) = u_0(x) \quad \text{for all } x \in \Omega \quad \text{and} \quad v(0,x) = v_0(x) \quad \text{for all } x \in \Omega. \]  

(1d)

In (1d) the functions \( u_0, v_0 : \Omega \to \mathbb{R} \) are given. The initial-boundary value problem (1) is called \( \lambda\)-\( \omega \) system.

To model chemical turbulence the choice

\[ \lambda(s) = 1 - s \quad \text{and} \quad \omega(s) = -\beta s \quad \text{with } \beta > 0 \]  

(2)

is proposed in [20, 24, 27]. Existence of a unique solution pair \((u, v)\) to (1) together with (2) is proved in [5] by using the Leray-Schauder fixed point theorem and the Gronwall lemma.

Remark 1 Note that (1a) together with (2) can be written as

\[ z_t(t,x) = z(t,x) - (1 + i\beta)|z(t,x)|^2z(t,x) + \sigma \Delta z(t,x) \quad \text{for all } (t,x) \in Q, \]  

(3)

where \( z = u + iv : \Omega \to \mathbb{C} \) with the imaginary unit \( i = \sqrt{-1} \) holds. Equation (3) is a special form of the complex Ginzburg-Landau model [29], which has been used widely in physics as a generic amplitude equation near the onset of instabilities that lead to turbulent/chaotic behavior in fluid dynamics.

The solution pair \((u, v)\) to (1) has the form of spiral waves that persist indefinitely. In Figure 1 snapshots of the \( \lambda\)-\( \omega \) system are plotted for the solution \( u \) at the time \( t = 100 \) for homogeneous Dirichlet boundary conditions and in Figure 2 for homogeneous Neumann boundary conditions. The corresponding plots for the solution \( v \) are similar to Figure 1 respectively Figure 2. If the parameter \( \beta \) in (2) increases, the spiral waves are not stable anymore and turbulent behavior can be observed.

The goal of our work is to solve (1) by a reduced-order approach utilizing POD. This approach is described in the next section.

3 POD REDUCED-ORDER MODELING

In this section we describe the POD approximation of the \( \lambda\)-\( \omega \) system. By \((u, v)\) we denote the unique solution pair to (1). We introduce the equidistant time grid \( \{t_j\}_{j=1}^n \) by

\[ \Delta t = \frac{T}{n-1} \quad \text{and} \quad t_j = (j-1)\Delta t \quad \text{for } j = 1, \ldots, n. \]  

(4)
Figure 1: Solution $u$ of the $\lambda$-$\omega$ system at the time $t = 100$ for homogeneous Dirichlet boundary conditions for $\beta \in \{1, 2, 3\}$ in (2) and for initial conditions $u_0(x, y) = (y - 1/2)$, $v_0(x, y) = (x - 1/2)/2$.

Figure 2: Solution $u$ of the $\lambda$-$\omega$ system at the time $t = 100$ for homogeneous Neumann boundary conditions and for $\beta \in \{1, 2, 3\}$ in (2) and for initial conditions $u_0(x, y) = (y - 1/2)$, $v_0(x, y) = (x - 1/2)/2$.

To simplify the presentation we choose an equidistant mesh in (4). Of course, the approach is analogous for non-equidistant time grids; see, e.g., [17]. Let $\bar{u}, \bar{v} : \Omega \to \mathbb{R}$ be given offsets. Further, $\{(u_j, v_j)\}_{j=1}^n$ denotes the solution pairs to (1) evaluated at the time grid $\{t_j\}_{j=1}^n$, i.e.,

$$u_j(x) = u(t_j, x), \quad v_j(x) = v(t_j, x) \quad \text{for } j = 1, \ldots, n \text{ and } x \in \Omega.$$  

**Remark 2** Common choices for the offsets — in particular in the context of turbulent flows [12] — are mean values, e.g., $(\bar{u}, \bar{v}) = (u_{\text{mean}}, v_{\text{mean}})$, where

$$u_{\text{mean}} = \frac{1}{n} \sum_{j=1}^n u_j \quad \text{and} \quad v_{\text{mean}} = \frac{1}{n} \sum_{j=1}^n v_j.$$  

Since the spatial functions $u_j$ and $v_j$ satisfy either homogeneous Dirichlet or homogeneous Neumann boundary conditions, their mean values possesses the same boundary conditions.
Next we take the snapshots
\[ \hat{u}_j = u_j - \bar{u} \quad \text{and} \quad \hat{v}_j = v_j - \bar{v} \quad \text{for} \quad j = 1, \ldots, n. \]

Introducing the \( L^2 \) inner product
\[ \langle \phi, \psi \rangle = \int_{\Omega} \phi \psi \, dx \] (5)
and the associated induced norm \( \| \phi \| = \sqrt{\langle \phi, \phi \rangle} \) we determine a POD basis of rank \( \ell \in \{1, \ldots, n\} \) for the function \( u \) by solving the constrained minimization problem
\[
\min_{\psi_1, \ldots, \psi_\ell} \sum_{j=1}^{n} \alpha_j \left\| \hat{u}_j - \sum_{i=1}^{\ell} \langle \hat{u}_j, \psi_i \rangle \psi_i \right\|^2 \quad \text{s.t.} \quad \langle \psi_i, \psi_j \rangle = \delta_{ij}, \quad 1 \leq i, j \leq \ell, \quad (P_u^\ell)
\]
where the \( \alpha_j \)'s are chosen as the trapezoidal weights \( \alpha_1 = \alpha_n = \Delta t/2 \) and \( \alpha_j = \Delta t \) for \( 2 \leq j \leq n - 1 \). In \( (P_u^\ell) \) we denote by \( \delta_{ij} \) the Kronecker symbol with \( \delta_{ij} = 1 \) for \( i = j \) and \( \delta_{ij} = 0 \) otherwise.

**Remark 3** Instead of (5) one can also utilize the \( H^1 \) inner product
\[ \langle \phi, \psi \rangle_{H^1} = \int_{\Omega} \phi \psi + \nabla \phi \cdot \nabla \psi \, dx \]
with its associated induced norm \( \| \cdot \|_{H^1} \); see [14, 16, 17].

Analogously, a POD basis of rank \( \ell \in \{1, \ldots, n\} \) for the function \( v \) solves
\[
\min_{\phi_1, \ldots, \phi_\ell} \sum_{j=1}^{n} \alpha_j \left\| \hat{v}_j - \sum_{i=1}^{\ell} \langle \hat{v}_j, \phi_i \rangle \phi_i \right\|^2 \quad \text{s.t.} \quad \langle \phi_i, \phi_j \rangle = \delta_{ij}, \quad 1 \leq i, j \leq \ell. \quad (P_v^\ell)
\]

The solutions to \( (P_u^\ell) \) and \( (P_v^\ell) \) are given by the solution to the symmetric eigenvalue problems [12, 30]
\[
R_u \psi_i = \lambda_i \psi_i \quad \text{and} \quad R_v \phi_i = \mu_i \phi_i \quad \text{for} \quad i = 1, \ldots, \ell, \quad (6)
\]
respectively, where
\[
R_u \psi = \sum_{j=1}^{n} \alpha_j \langle \hat{u}_j, \psi \rangle \hat{u}_j, \quad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_\ell > 0,
\]
\[
R_v \phi = \sum_{j=1}^{n} \alpha_j \langle \hat{v}_j, \phi \rangle \hat{v}_j, \quad \mu_1 \geq \mu_2 \geq \ldots \geq \mu_\ell > 0.
\]
Remark 4 (Methods of snapshot [28]) Let us supply \( \mathbb{R}^n \) with the weighted inner product
\[
\langle \varphi, \chi \rangle_{\mathbb{R}^n} = \sum_{i=1}^{n} \alpha_i \varphi^i \chi^i \quad \text{for} \quad \varphi = (\varphi^1, \ldots, \varphi^n)^T, \chi = (\chi^1, \ldots, \chi^n)^T \in \mathbb{R}^n.
\]
We define the symmetric non-negative matrix \( K_u \in \mathbb{R}^{n \times n} \) with the elements \( \langle \hat{u}_i, \hat{u}_j \rangle \), \( 1 \leq i, j \leq n \), and consider the eigenvalue problem
\[
K_u \varphi_i = \lambda_i \varphi_i, \quad 1 \leq i \leq \ell \quad \text{and} \quad \langle \varphi_i, \varphi_j \rangle_{\mathbb{R}^n} = \delta_{ij}, \quad 1 \leq i, j \leq \ell. \tag{7}
\]
Often the matrix \( K_u \) is called a correlation matrix for the snapshots ensemble \( \{\hat{u}_j\}_{j=1}^n \).

From singular value decomposition it follows that \( K_u \) has the same eigenvalues \( \{\lambda_i\}_{i=1}^\ell \) as the operator \( R_u \). Furthermore, the POD basis functions are given by the formula
\[
\psi_i = \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^{n} \alpha_j (\varphi_i)^j \hat{u}_j \quad \text{for} \quad i = 1, \ldots, \ell,
\]
where \( (\varphi_i)^j \) denotes the \( j \)-th component of the eigenvector \( \varphi_i \in \mathbb{R}^n \). This approach is called methods of snapshots. Analogously, the POD basis \( \{\phi_i\}_{i=1}^\ell \) can be computed by defining the symmetric non-negative matrix \( K_v \in \mathbb{R}^{n \times n} \) with the elements \( \langle \hat{v}_i, \hat{v}_j \rangle \), \( 1 \leq i, j \leq n \), solving the eigenvalue problem
\[
K_v \chi_i = \mu_i \chi_i, \quad 1 \leq i \leq \ell, \quad \text{and} \quad \langle \chi_i, \chi_j \rangle_{\mathbb{R}^n} = \delta_{ij}, \quad 1 \leq i, j \leq \ell. \tag{8}
\]
and setting
\[
\phi_i = \frac{1}{\sqrt{\mu_i}} \sum_{j=1}^{n} \alpha_j (\chi_i)^j \hat{v}_j \quad \text{for} \quad i = 1, \ldots, \ell.
\]
Note that (7) and (8) are eigenvalue problems in \( \mathbb{R}^n \), whereas (6) are eigenvalue problems in \( L^2(\Omega) \) that have to be discretized for numerical realization. In particular, for two- or three-dimensional domains \( \Omega \) it turns out that the POD basis should be computed by solving (7) instead the discretized version of (6). In our numerical test examples (see Section 4) we compute the POD basis by utilizing the methods of snapshots.

Remark 5 Note that \( (P_u^\ell) \) can be interpreted as the trapezoidal approximation of
\[
\min_{\psi_1, \ldots, \psi_\ell} \int_0^T \left\| (u(t, \cdot) - \bar{u}) - \sum_{i=1}^{\ell} \langle u(t, \cdot) - \bar{u}, \psi_i \rangle \psi_i \right\|^2 dt \quad \text{s.t.} \quad \langle \psi_i, \psi_j \rangle = \delta_{ij}, \quad 1 \leq i, j \leq \ell.
\]
This relationship is used in [17] to study asymptotic convergence properties of the eigenfunctions \( \{\psi_i\}_{i=1}^\ell \) and the corresponding eigenvalues \( \{\lambda_i\}_{i=1}^\ell \) as the mesh size \( \Delta t \) tends to zero or, equivalently, \( n \) tends to \( \infty \).
When the POD bases \( \{\psi_i\}_{i=1}^\ell \) and \( \{\phi_i\}_{i=1}^\ell \) have been computed, we make the ansatz

\[
  u_\ell(t, x) = \bar{u}(x) + \sum_{j=1}^\ell u_\ell^j(t)\psi_j(x) \quad \text{and} \quad v_\ell(t, x) = \bar{v}(x) + \sum_{j=1}^\ell v_\ell^j(t)\phi_j(x)
\]

for the POD Galerkin approximations of \( u \) and \( v \), respectively. Recall that \( u_j \) and \( v_j \) satisfy homogeneous Dirichlet or Neumann boundary conditions on \( \Gamma \). Due to Remark 2 both POD approximations \( u_\ell \) and \( v_\ell \) satisfy homogeneous boundary condition on \( \Gamma \) for the choices \((\bar{u}, \bar{v}) = (u_{\text{mean}}, v_{\text{mean}}) \) and \((\bar{u}, \bar{v}) = (0, 0) \). In this case (1b) or (1c) hold for the POD approximations \( u_\ell \) and \( v_\ell \) so that the boundary conditions for \( u_\ell \) and \( v_\ell \) are satisfied.

Inserting (9) into (1a) we find

\[
  \sum_{j=1}^\ell \dot{u}_\ell^j(t)\psi_j = \lambda(s_\ell)\bar{u} - \omega(s_\ell)\bar{v} + \sigma\Delta\bar{u} + \sum_{j=1}^\ell u_\ell^j(t)s\Delta\psi_j + \lambda(s_\ell)\sum_{j=1}^\ell u_\ell^j(t)\psi_j - \omega(s_\ell)\sum_{j=1}^\ell v_\ell^j(t)\phi_j,
\]

\[
  \sum_{j=1}^\ell \dot{v}_\ell^j(t)\phi_j = \omega(s_\ell)\bar{u} + \lambda(s_\ell)\bar{v} + \sigma\Delta\bar{v} + \sum_{j=1}^\ell v_\ell^j(t)s\Delta\phi_j + \omega(s_\ell)\sum_{j=1}^\ell u_\ell^j(t)\phi_j + \lambda(s_\ell)\sum_{j=1}^\ell v_\ell^j(t)\phi_j
\]

with \( s_\ell = u_\ell(t, \cdot)^2 + v_\ell(t, \cdot)^2 \) and \( t \in (0, T] \). Next we multiply (10) by \( \psi_i \), \( i = 1, \ldots, \ell \), (11) by \( \phi_i \), \( i = 1, \ldots, \ell \), and integrate over \( \Omega \). From \( \int_\Omega \psi_j \psi_i \, dx = \int_\Omega \phi_j \phi_i \, dx = \delta_{ij} \) we obtain

\[
  \begin{pmatrix}
    \dot{u}_\ell(t) \\
    \dot{v}_\ell(t)
  \end{pmatrix} = \sigma
  \begin{pmatrix}
    A_u & B_u(n(t), \psi_\ell(t)) \\
    A_v & B_v(n(t), \psi_\ell(t))
  \end{pmatrix}
  \begin{pmatrix}
    u_\ell(t) \\
    v_\ell(t)
  \end{pmatrix} + \begin{pmatrix}
    f_u \\
    f_v
  \end{pmatrix}
\]

for \( t \in (0, T] \), (12a)

where the vectors \( u_\ell, v_\ell, f_u, f_v \) satisfy

\[
  u_\ell = \begin{pmatrix}
    u_\ell^1 \\
    \vdots \\
    u_\ell^\ell
  \end{pmatrix}, \quad v_\ell = \begin{pmatrix}
    v_\ell^1 \\
    \vdots \\
    v_\ell^\ell
  \end{pmatrix}, \quad f_u = \begin{pmatrix}
    \int_\Omega \sigma\Delta\bar{u}\psi_1 \, dx \\
    \vdots \\
    \int_\Omega \sigma\Delta\bar{u}\psi_\ell \, dx
  \end{pmatrix}, \quad f_v = \begin{pmatrix}
    \int_\Omega \sigma\Delta\bar{v}\phi_1 \, dx \\
    \vdots \\
    \int_\Omega \sigma\Delta\bar{v}\phi_\ell \, dx
  \end{pmatrix},
\]

the matrices \( A_u, A_v \) are given by

\[
  A_u = \left( (a_u^{ij}) \right) \in \mathbb{R}^{\ell \times \ell} \quad \text{with} \quad a_u^{ij} = \int_\Omega \Delta\psi_j \psi_i \, dx,
\]

\[
  A_v = \left( (a_v^{ij}) \right) \in \mathbb{R}^{\ell \times \ell} \quad \text{with} \quad a_v^{ij} = \int_\Omega \Delta\phi_j \phi_i \, dx,
\]
and the non-linear terms
\[
F_u(u_\ell(t), v_\ell(t)) = \left( \begin{array}{c} F_u^1(u_\ell(t), v_\ell(t)) \\ \vdots \\ F_u^\ell(u_\ell(t), v_\ell(t)) \end{array} \right), \quad F_v(u_\ell(t), v_\ell(t)) = \left( \begin{array}{c} F_v^1(u_\ell(t), v_\ell(t)) \\ \vdots \\ F_v^\ell(u_\ell(t), v_\ell(t)) \end{array} \right)
\]
are defined as
\[
F_u^i(u_\ell(t), v_\ell(t)) = \int_\Omega (\lambda(s_\ell)u_\ell(t, \cdot) - \omega(s_\ell)v_\ell(t, \cdot))\psi_i \, dx,
\]
\[
F_v^i(u_\ell(t), v_\ell(t)) = \int_\Omega (\omega(s_\ell)u_\ell(t, \cdot) + \lambda(s_\ell)v_\ell(t, \cdot))\phi_i \, dx
\]
for \(1 \leq i \leq \ell\). From (1d) we derive the initial condition
\[
\left( \begin{array}{c} u_\ell(0) \\ v_\ell(0) \end{array} \right) = \left( \begin{array}{c} u_\circ \\ v_\circ \end{array} \right) - \left( \begin{array}{c} \bar{u} \\ \bar{v} \end{array} \right)
\]
(12b)
with
\[
u_\circ = \left( \begin{array}{c} \int_\Omega u_\circ \psi_1 \, dx \\ \vdots \\ \int_\Omega u_\circ \psi_\ell \, dx \end{array} \right), \quad v_\circ = \left( \begin{array}{c} \int_\Omega v_\circ \phi_1 \, dx \\ \vdots \\ \int_\Omega v_\circ \phi_\ell \, dx \end{array} \right),
\]
\[
\bar{u} = \left( \begin{array}{c} \int_\Omega \bar{u} \psi_1 \, dx \\ \vdots \\ \int_\Omega \bar{u} \psi_\ell \, dx \end{array} \right), \quad \bar{v} = \left( \begin{array}{c} \int_\Omega \bar{v} \phi_1 \, dx \\ \vdots \\ \int_\Omega \bar{v} \phi_\ell \, dx \end{array} \right).
\]
Summarizing the POD reduced-order model is given by (12) that is solved numerically in Section 4 for different values for the parameter \(\beta\) in (2) and number \(\ell\) of POD ansatz functions.

4 NUMERICAL EXPERIMENTS

This section is devoted to present two numerical test examples. All coding is done in MATLAB. The programs are executed on a standard Pentium D, 3 Ghz desktop PC.

We choose the initial conditions
\[
u_\circ(x, y) = y - \frac{1}{2} \quad \text{and} \quad v_\circ(x, y) = \frac{x}{2} - \frac{1}{4}.
\]
The snapshots \(\{(u_j, v_j)\}_{j=1}^n\) are computed by applying a finite difference approximation for the spatial domain (with a five-point discretization for the Laplace operator). In (4) we choose \(T = 50, n = 501\) and the spatial equidistant mesh size \(h > 0\) and the corresponding grid points \(x_{ij} = (x_i, y_j) \in \Omega\) are given by
\[
h = \frac{1}{N+1}, \quad x_i = i h \text{ for } 0 \leq i \leq N+1, \quad y_j = j h \text{ for } 0 \leq j \leq N+1.
\]
After spatial discretization we arrive at an initial-value problem for \(2N^2\) coupled ordinary differential equations in case of boundary condition (1b) and for \(2(N+2)^2\) coupled ordinary differential equations in case of (1c). These systems are solved by the MATLAB routine `ode15s` to obtain an approximation \((u_h, v_h)\) to the solution of (1). As snapshots we take the ensemble \(\{(u_h(t_j, \cdot), v_h(t_j, \cdot))\}_{j=1}^{501}\).

### 4.1 Dirichlet boundary conditions

Let us investigate the \(\lambda - \omega\) system with homogeneous Dirichlet boundary conditions. We analyze numerically the dependence of the accuracy of the reduced-order model on the parameter \(\beta\) in (2). First we compute the \(n = 501\) snapshots, solve the eigenvalue problems (7), (8) and determine the POD bases \(\{\psi_i\}_{i=1}^{\ell}\) and \(\{\chi_i\}_{i=1}^{\ell}\). The routine `ode15s` needs 2644.09 seconds and the computation of the reduced-order model requires about 4.5 seconds for \(\ell = 50\). The decay of the first 50 eigenvalues is shown in Figure 3. We observe from Figure 3 that the eigenvalues decay slower for increasing parameter \(\beta\). For small \(\beta\) only a few modes are needed to reflect the evolution of the whole system, but with increasing \(\beta\) the model becomes turbulent. Therefore, the eigenvalues decay not as fast as for smaller parameter \(\beta\).

Next we take \(\bar{u} = \bar{v} = 0\) (no offsets in the POD Galerkin projection) and solve the reduced-order model by applying the MATLAB solver `ode15s`. In Figure 4 we compare the relative errors

\[
t \mapsto \frac{\|u_{\ell}(t) - u_h(t)\|^2}{\|u_h(t)\|^2}
\]

for \(\beta = 1.5\) (left plot) and \(\beta = 2\) (right plot). It turns out that for \(\beta = 1.5\) (left plot in Figure 4) even \(\ell = 10\) POD ansatz functions are sufficient to obtain a reliable reduced-order model. On the other hand, for \(\beta = 2\) the POD solution based on \(\ell = 10\) is completely...
different from the finite difference solution. Also for $\ell = 15$ and $\ell = 25$ we observe that the relative errors increase drastically for $t > 25$ and $t > 30$, respectively. Using $\ell = 50$ POD basis functions yields a reliable POD solution that is close to the finite difference solution $u^h$. In Table 1 the relative errors

$$E_{\text{rel}}(u) = \frac{\sum_{j=1}^{n} \alpha_j \| u_\ell(t_j) - u_h(t_j) \|^2}{\sum_{j=1}^{n} \alpha_j \| u_h(t_j) \|^2}$$

are presented. We observe that the errors decrease if $\ell$ increase. Moreover, for larger values of $\beta$ the errors increases for fixed number $\ell$ of POD ansatz functions. In particular, for $\beta = 3$ the error for $\ell = 50$ is quite high so that we should include more POD basis functions in our reduced-order approach. The relative errors $E_{\text{rel}}(v)$ for the $v$ are plotted in Table 2. The results for the POD solution $v_\ell$ are similar than the results for $u_\ell$.

4.2 Neumann boundary conditions

Now we consider the $\lambda-\omega$ system with the homogeneous Neumann boundary conditions (1c). It turns out that the dependence of the quality of the reduced-order model depends

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>$\beta = 1$</th>
<th>$\beta = 1.5$</th>
<th>$\beta = 2$</th>
<th>$\beta = 2.5$</th>
<th>$\beta = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0002742</td>
<td>0.0592466</td>
<td>2.6147003</td>
<td>2.0664552</td>
<td>1.9121706</td>
</tr>
<tr>
<td>15</td>
<td>0.0000399</td>
<td>0.0006219</td>
<td>0.7012245</td>
<td>1.6072066</td>
<td>2.0462711</td>
</tr>
<tr>
<td>25</td>
<td>0.0000003</td>
<td>0.0000108</td>
<td>0.5548754</td>
<td>1.5939146</td>
<td>1.5950328</td>
</tr>
<tr>
<td>50</td>
<td>0.0000001</td>
<td>0.0000039</td>
<td>0.0000519</td>
<td>0.2994985</td>
<td>0.9057551</td>
</tr>
</tbody>
</table>

Table 1: The relative error $E_{\text{rel}}(u)$ for different numbers $\ell$ of POD ansatz functions and for different values of the parameter $\beta$ in (1) in case of homogeneous Dirichlet boundary conditions.
The relative error $E_{\text{rel}}(v)$ for different numbers $\ell$ of POD ansatz functions and for different values of the parameter $\beta$ in (1) in case of homogeneous Dirichlet boundary conditions.

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>$\beta = 1$</th>
<th>$\beta = 1.5$</th>
<th>$\beta = 2$</th>
<th>$\beta = 2.5$</th>
<th>$\beta = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0002699</td>
<td>0.0588345</td>
<td>2.6087718</td>
<td>2.0671781</td>
<td>1.9116376</td>
</tr>
<tr>
<td>15</td>
<td>0.0000038</td>
<td>0.00006968</td>
<td>0.6998566</td>
<td>1.5974137</td>
<td>2.0528978</td>
</tr>
<tr>
<td>25</td>
<td>0.0000026</td>
<td>0.00000105</td>
<td>0.5590867</td>
<td>1.5945250</td>
<td>1.5956316</td>
</tr>
<tr>
<td>50</td>
<td>0.0000001</td>
<td>0.00000039</td>
<td>0.0000512</td>
<td>0.2970237</td>
<td>0.9108994</td>
</tr>
</tbody>
</table>

Table 2: The relative error $E_{\text{rel}}(v)$ for different numbers $\ell$ of POD ansatz functions and for different values of the parameter $\beta$ in (1) in case of homogeneous Dirichlet boundary conditions.

on the size of the parameter $\beta$. The computation of the reduced-order model for (1) is done analogously as described in Section 4.1. The CPU time to compute the $n = 501$ snapshots is 3176.03 seconds and the calculation of the POD bases requires about 4.5 seconds for $\ell = 50$ ansatz functions. The decay of the first 50 eigenvalues is shown in Figure 5. Thus,

the decay rate of the eigenvalues has the same dependence on the parameter $\beta$ as in the case of homogeneous Dirichlet boundary conditions.

Let us compare the quality of the reduced-order solution with respect to the choice of the offset $\bar{u}$ and $\bar{v}$ in the POD Galerkin projection (compare Remark 2). In Table 3 we plot the relative errors $E_{\text{rel}}(u)$ and the CPU times for the solution of the reduced-order model (12) for $\beta = 1.5$. We observe that there is no big difference in the errors. Thus, for small values of $\beta$ we can take $\bar{u} = 0$ to get a reliable reduced-order model. between the two methods. Note that the POD solve is — depending on $\ell$ — much faster than the finite difference solve. In Table 4 we compare the errors for the two different offsets in case of $\beta = 2$, where the $\lambda$-$\omega$ systems has already turbulent features. For $\beta = 2$ with $\ell < 40$ we get no usefull results, so we compare the errors for the choice $\ell = 40, 45, 50$. We observe that the POD model reduction performs better provided we take $\bar{u} = u_{\text{mean}}$. Note that for $\bar{u} = 0$ the POD solutions differ significantly from the finite difference solution in case of $\ell = 40$ and 45.

Figure 5: Decay of the first eigenvalues for different values of $\beta$ in (1) and homogeneous Neumann boundary conditions.
Table 3: The relative error $E_{\text{rel}}(u)$ for different numbers $\ell$ of POD ansatz functions and for different types of POD calculation and for $\beta = 1.5$ in (2) in case of homogeneous Neumann boundary conditions (1b) and the CPU times.

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>$\bar{u} = 0$</th>
<th>$\bar{u} = u_{\text{mean}}$</th>
<th>CPU-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$0.0058895$</td>
<td>$0.0059449$</td>
<td>17.0 seconds</td>
</tr>
<tr>
<td>15</td>
<td>$0.0003499$</td>
<td>$0.0003351$</td>
<td>35.9 seconds</td>
</tr>
<tr>
<td>25</td>
<td>$0.0000305$</td>
<td>$0.0000257$</td>
<td>54.3 seconds</td>
</tr>
<tr>
<td>50</td>
<td>$0.0000093$</td>
<td>$0.0000092$</td>
<td>139.2 seconds</td>
</tr>
</tbody>
</table>

Table 4: The relative error $E_{\text{rel}}(u)$ for different numbers $\ell$ of POD ansatz functions and for different types of POD calculation for $\beta = 2$ in (2) in case of homogeneous Neumann boundary conditions.

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>$\bar{u} = 0$</th>
<th>$\bar{u} = u_{\text{mean}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>$0.5774415$</td>
<td>$0.4601877$</td>
</tr>
<tr>
<td>45</td>
<td>$0.8986130$</td>
<td>$0.2976193$</td>
</tr>
<tr>
<td>50</td>
<td>$0.0710348$</td>
<td>$0.0017736$</td>
</tr>
</tbody>
</table>

4.3 Predictor corrector method

From the numerical experiments carried out in Sections 4.1 and 4.2 we observe that for $\beta \geq 2$ the number of POD basis functions used for the model reduction should be larger than 25. On the other hand, the more POD ansatz functions in the POD are used, the more time is needed by the POD solver of the $\lambda$-$\omega$ system. Since the most energy is contained in the lower modes, we split the ansatz for $u$ and $v$ as follows:

$$u_\ell(t, x) = \bar{u}(x) + \sum_{j=1}^{\ell_1} u_j^\ell(t) \psi_j(x) + \sum_{j=\ell_1+1}^{\ell} u_j^\ell(t) \psi_j(x) = \bar{u}(x) + u_{\text{low}}(t, x) + u_{\text{high}}(t, x),$$

$$v_\ell(t, x) = \bar{v}(x) + \sum_{j=1}^{\ell_1} v_j^\ell(t) \phi_j(x) + \sum_{j=\ell_1+1}^{\ell} v_j^\ell(t) \phi_j(x) = \bar{v}(x) + v_{\text{low}}(t, x) + v_{\text{high}}(t, x),$$

where

$$u_{\text{low}}(t, x) = \sum_{j=1}^{\ell_1} u_j^\ell(t) \psi_j(x),$$

$$u_{\text{high}}(t, x) = \sum_{j=\ell_1+1}^{\ell} u_j^\ell(t) \psi_j(x),$$

$$v_{\text{low}}(t, x) = \sum_{j=1}^{\ell_1} v_j^\ell(t) \phi_j(x),$$

$$v_{\text{high}}(t, x) = \sum_{j=\ell_1+1}^{\ell} v_j^\ell(t) \phi_j(x).$$

Algorithm 2 (Predictor-Corrector method)

0) Let $\ell = \ell_1 + \ell_2$, $\Delta \tau \in (0, T]$; put $k = 1$. 

12
1) Set
\[ u_{\text{high}}^k = \sum_{j=\ell_1+1}^{\ell} \langle u_0, \psi_j \rangle \psi_j(x), \quad v_{\text{high}}^k = \sum_{j=\ell_1+1}^{\ell} \langle v_0, \phi_j \rangle \phi_j(x), \]

\[
\begin{aligned}
&u_{\text{high}}^k = \left( \begin{array}{c} \langle u_{\text{high}}^k, \psi_{\ell_1+1} \rangle \\ \vdots \\ \langle u_{\text{high}}^k, \psi_{\ell} \rangle \end{array} \right), \\
v_{\text{high}}^k = \left( \begin{array}{c} \langle v_{\text{high}}^k, \phi_{\ell_1+1} \rangle \\ \vdots \\ \langle v_{\text{high}}^k, \phi_{\ell} \rangle \end{array} \right)
\end{aligned}
\]

and solve the \( \ell_1 \)-dimensional problem
\[
\begin{bmatrix}
\dot{u}_{\ell_1}(t) \\
\dot{v}_{\ell_1}(t)
\end{bmatrix} = \sigma \begin{bmatrix} A_u u_{\ell_1}(t) \\ A_v v_{\ell_1}(t) \end{bmatrix} + \begin{bmatrix} F_u^k(u_{\ell_1}(t), v_{\ell_1}(t)) \\ F_v^k(u_{\ell_1}(t), v_{\ell_1}(t)) \end{bmatrix} + \begin{bmatrix} f_a \\ f_{\ell_1} \end{bmatrix}
\]
on the interval \([ (k-1)\Delta \tau, k\Delta \tau ] = [0, \Delta \tau] \), where
\[
\begin{aligned}
(F_u^k)(u_{\ell_1}(t), v_{\ell_1}(t)) &= \int_{\Omega} \left( \lambda(s_{\ell_1})(u_{\ell_1}(t, \cdot) + u_{\text{high}}^k) - \omega(s_{\ell_1})(v_{\ell_1}(t, \cdot) + v_{\text{high}}^k) \right) \psi_{\ell_1} \, dx, \\
(F_v^k)(u_{\ell_1}(t), v_{\ell_1}(t)) &= \int_{\Omega} \left( \omega(s_{\ell_1})(u_{\ell_1}(t, \cdot) + u_{\text{high}}^k) + \lambda(s_{\ell_1})(v_{\ell_1}(t, \cdot) + v_{\text{high}}^k) \right) \phi_{\ell_1} \, dx
\end{aligned}
\]
for \( i = 1, \ldots, \ell_1 \) by the explicit Euler method.

2) Compute \((u_{\text{high}}^{k+1}, v_{\text{high}}^{k+1}) \in \mathbb{R}^{\ell_2} \times \mathbb{R}^{\ell_2}\) from
\[
\begin{bmatrix}
\frac{1}{\Delta \tau}(u_{\text{high}}^{k+1} - u_{\text{high}}^k) \\
v_{\text{high}}^{k+1} - v_{\text{high}}^k
\end{bmatrix} = \sigma \begin{bmatrix} A_u u_{\text{high}}^{k+1} \\ A_v v_{\text{high}}^{k+1} \end{bmatrix} + \begin{bmatrix} F_u^k(u_{\ell_1}(t), v_{\ell_1}(t)) \\ F_v^k(u_{\ell_1}(t), v_{\ell_1}(t)) \end{bmatrix} + \begin{bmatrix} f_{\text{high}} \end{bmatrix}
\]
Set \( k = k + 1 \) and define
\[
\begin{aligned}
u_{\text{high}}^k &= \sum_{j=\ell_1+1}^{\ell} \langle u_{\text{high}}^k, \psi_j \rangle \psi_j(x), \\
v_{\text{high}}^k &= \sum_{j=\ell_1+1}^{\ell} \langle v_{\text{high}}^k, \phi_j \rangle \phi_j(x).
\end{aligned}
\]

3) Solve the \( \ell_1 \)-dimensional problem
\[
\begin{bmatrix}
\dot{u}_{\ell_1}(t) \\
\dot{v}_{\ell_1}(t)
\end{bmatrix} = \sigma \begin{bmatrix} A_u u_{\ell_1}(t) \\ A_v v_{\ell_1}(t) \end{bmatrix} + \begin{bmatrix} F_u^k(u_{\ell_1}(t), v_{\ell_1}(t)) \\ F_v^k(u_{\ell_1}(t), v_{\ell_1}(t)) \end{bmatrix} + \begin{bmatrix} f_a \\ f_{\ell_1} \end{bmatrix}
\]
on the interval \([ (k-1)\Delta \tau, k\Delta \tau ] = [\Delta \tau, 2\Delta \tau] \).

4) Go back to step 2) unless the terminal time \( T \) is reached.
Note that we use the first $\ell_1$ POD basis functions for the (linear) POD Galerkin approximation and utilize the estimates $(u^k_{\text{high}}, v^k_{\text{high}})$ as an approximation for the higher ($j > \ell_1$) POD modes. This estimates are taken to correct the estimated POD solutions $(u_{\ell_1}, v_{\ell_1})$.

Let $\delta t = 0.001$ be the time step for the explicit method and $\Delta \tau = l \delta t$. First we compare the predictor-corrector method to the linear Galerkin algorithm for $\beta = 2$ and homogeneous Neumann boundary conditions on the time interval $[0, 40]$. For every combination of $\ell_1$ and $\ell_2$ two runs with $l = 10$ and $l = 100$ are made. Because of the turbulent behavior of the system $t > 20$, we take the same large values of $\ell$ as in Table 4 and also the mean $\bar{u} = u_{\text{mean}}$ is used in both methods. The Galerkin method with $\ell = 50$ has the lowest relative error, because all modes are updated in every step. Choosing $l = 100$ the predictor-corrector method with $\ell_1$ lower than 30 results in a large relative error, for $\ell_1 = 30$ the results are slightly better than the solution of (12) in case of $\ell = 45$. Reducing $l$ to 10 really improves the predictor-corrector method, because the higher modes are updated more often.

In a second test with the same $\beta$, but only on the time interval $[0, 20]$ we compare the methods for a smaller number of modes. Now for the smaller $l$ the relative error of the predictor-corrector approach is even a little smaller than for the Galerkin with $\ell = \ell_1 + \ell_2$, but this has no significance. The Galerkin method with $\ell = 15$ is much worse than the predictor-corrector method. After setting $l = 100$ the number of lower modes has to be increased to 15, otherwise the error is to large.

When the cost for the solution of the reduced system depends strongly on the number of modes, then the proposed predictor-corrector method gives the chance to get accurate results in a shorter time, because the higher modes are not updated that often.
4.4 Snapshot archetypes by Adrover

If the calculation of the POD basis from all snapshots is very time-consuming, then it is useful to reduce the number of snapshots. On one hand, this could be done by increasing the time differences $\Delta t$ between the snapshots, on the other hand a selection rule like the one proposed in [1] can be applied.

Algorithm 1 (Choice of snapshots [1])

0) Fix the number $k \in \{2, \ldots, n\}$ of snapshots.

1) Set $\tilde{u}_1 = u_o$, $t_{i_1} = t_1 = 0$ and $k = 1$.

2) Determine

$$t_{k+1} = \text{argmax} \left\{ \min_{1 \leq l \leq k} \left\| u(t_j, \cdot) - u(t_{i_l}, \cdot) \right\| \mid t_j \in \{t_1, \ldots, t_n\} \setminus \{t_{i_1}, \ldots, t_{i_k}\} \right\}$$

3) Set $k = k + 1$ and $\tilde{u}_k(x) = u(t_k, x)$ for all $x \in \Omega$. If $k < k$, go back to step 2).

We choose the parameter $\beta = 1.5$, the terminal time $T = 50$ and homogeneous Neumann boundary conditions (1c). The goal is to perform the POD reduced-order modeling by taking only $k = 100$ instead of $n = 501$ snapshots. We compute the snapshots $\{\tilde{u}_j\}_{j=1}^{100}$ by Algorithm 1 with the ensemble $\{\bar{u}_j\}_{j=1}^{100}$ with $\bar{u}_j = u((j - 1)5\Delta t, \cdot)$ (i.e., compared to (4) we take a 5 times coarser grid). In Table 7 we compare the relative error $E_{\text{rel}}(u)$ for different numbers $\ell$ of POD ansatz functions, where the POD basis is determined by utilizing the snapshots $\{\bar{u}_j\}$ (coarse time grid), $\{\tilde{u}_j\}_{j=1}^{100}$ (Algorithm 1) and $\{\hat{u}_j\}_{j=1}^{501}$ (fine time grid). Let us mention that we take zero offsets in (9). We observe that the results for the coarse time grid are as good as the corresponding ones for the fine time grid, but the errors for the POD Galerkin solution computed via the snapshots determined by Algorithm 1 are higher.
5 CONCLUSIONS

In this article reduced-order approximations for $\lambda$-$\omega$ systems are considered by applying POD Galerkin projection. It turns out that the quality of the POD Galerkin approximation depends on a parameter $\beta$ in the $\lambda$-$\Omega$ system. The higher the parameter $\beta$ is chosen the more turbulent behavior the solution has. Numerical experiments for the Dirichlet- and Neumann boundary conditions are presented. We compare the linear Galerkin scheme with a predictor-corrector method. Motivated by our observations the extension of our predictor-corrector method to non-linear Galerkin approximations can be a focus of future research.

REFERENCES


See http://www.uni-graz.at/imawww/reports/archive-2006/IMA06-06.pdf.

See http://www.uni-graz.at/imawww/reports/archive-2006/IMA14-06.pdf.


See http://www.uni-graz.at/imawww/reports/archive-2006/IMA10-06.pdf.


