User's Manual for the Function `matpod`. A MATlab function for the Proper Orthogonal Decomposition technique

The MatMOL Group (2009)

1 Introduction

The main inconvenience of classical techniques like the FEM or the FD is that, in general, they result into a large set of ODEs. This is specially critical when dealing with real time tasks as optimization and control. During the last decades a number of techniques for the model reduction of PDEs systems have arosen, among which, one of the most popular is the proper orthogonal decomposition (POD) (Sirovich, 1987; Berkooz et al., 1993).

In dissipative systems, the dynamics evolve to a low dimensional subspace and remain in it in the future. This property will allow us to approximate the dynamics of the system by computing only the dynamics in the low dimensional subspace. The basis functions of such subspace can be computed in many different forms. In the case of the POD, the procedure is as follows:

- Obtain a set of measurements or simulation data representative of the behaviour of the system.
- Compute the kernel $\mathcal{K}(\xi, \xi')$ of the integral equation

$$\phi_i(\xi) = \lambda_i \int_{\mathscr{V}} \mathcal{K}(\xi, \xi')\phi_i(\xi')d\xi', \tag{1}$$

  For a discrete set of data (Z_i), the kernel is computed as:

$$\mathcal{K} = \frac{1}{\ell}\sum_{i=1}^{\ell} Z_i Z_i^T. \tag{2}$$

  where ℓ number of elements in the data set.

- Solve the integral equation (1).

In the toolbox, this procedure for obtaining the basis functions will be known as *direct method*. It must be pointed out that for large values of N, solving Eqn (1) can be computationally involved. In order to avoid this problem, a useful alternative, proposed in Sirovich (1987) and known as the

method of snapshots or *strobes*, is briefly discussed. In this method, which will be known in the toolbox as *indirect method*, each eigenfunction is expressed in terms of the original data as:

$$\phi_j = \sum_{i=1}^{\ell} w_i^j Z_i, \tag{3}$$

where w_i^j are the weights to be computed. To this purpose, the following matrix is defined

$$\mathcal{M}_{ij} = \frac{1}{\ell} \langle Z_i, Z_j \rangle_{\mathcal{Y}}. \tag{4}$$

Introducing Eqns (2) and (3) in the eigenvalue problem (1), results into:

$$\mathcal{M}\mathcal{W}_j = \lambda_j \mathcal{W}_j, \tag{5}$$

where the eigenvectors $\mathcal{W}_j$ have as element the weights in equation (3) so that $\mathcal{W}_j = [w_1^j, w_2^j, ..., w_\ell^j]^T$.

A more detailed description, including references, about this technique can be obtained in the thesis (Vilas, 2008; García, 2008).

Another important point which will be employed along the examples is the relationship between the FEM matrices (see the reference manual of the `user_manual_matfem.m` function) and spatial derivatives and integrals. Such relationships are summarized in Table 1.

	Continuous	**Discrete**
1	$\int_{\mathcal{Y}} g(\boldsymbol{\xi}) f(\boldsymbol{\xi}) d\boldsymbol{\xi}$	$\mathcal{G}^T \mathcal{M} \mathcal{M} \mathcal{F}$
2	$\int_{\mathcal{Y}} g(\boldsymbol{\xi}) \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i}$ d$\boldsymbol{\xi}$	$\mathcal{G}^T \mathcal{C} \mathcal{M}_i \mathcal{F}$
3	$\int_{\mathcal{Y}} g(\boldsymbol{\xi}) \Delta f(\boldsymbol{\xi})$ d$\boldsymbol{\xi}$	$-\mathcal{G}^T \left(\mathcal{D}\mathcal{M} + \mathcal{B}\mathcal{M} \right) \mathcal{F}$
4	$\frac{\partial}{\partial \xi_i}$	$\mathcal{M}\mathcal{M}^{-1}\mathcal{C}\mathcal{M}_i$
5	$\frac{\partial^2}{\partial \xi_1^2} + \frac{\partial^2}{\partial \xi_2^2} + \frac{\partial^2}{\partial \xi_3^2}$	$-\mathcal{M}\mathcal{M}^{-1} \left(\mathcal{D}\mathcal{M} + \mathcal{B}\mathcal{M} \right)$

Table 1: Relationships between the continuous spatial derivatives and integrals and their discrete counterparts using the FEM matrices. Vectors $\mathcal{G}$ and $\mathcal{F}$ are the FEM discretized versions of continuous functions $g(\boldsymbol{\xi})$ and $f(\boldsymbol{\xi})$.

In the section 2 the function `matpod` will be employed for computing the basis functions which will define the low dimensional subespace.

2 The Function `matpod`

This function will compute the basis functions using the POD technique from a set of experimental or simulation data. The discretization of the data must coincide with the discretization of the finite element method. The way of calling the function is

```
[pods] = matpod(V , MM , mth , ener)
```

where the input and output parameters are

```
Input variables:
V:    Set of simulation or experimental data. Each column corresponds with
      a snapshot at a given time
MM:   Mass matrix obtained from the FEM. The FEM discretization must
      coincide with the discretization for V
mth:  Method used for computing the basis functions. 'd' is the direct
      method, 'i' is the indirect method. The indirect method results
      more efficient if the number of discretization points is much larger
      than the number of measurements.
ener: Energy captured by the PODs.

Output variables:
pods: This is a structure where pods.phi are the basis functions and
      pods.lambda the eigenvalues
```

It is important to note that the rows of the data matrix V are related to the FEM spatial discretization while the data at different times are stored in the columns.

3 Examples

In this section the same examples as in the `matfem` function manual will be employed to illustrate the use of the `matpod` function.

3.1 A simple diffusion problem

Using the POD technique, compute the numerical solution of equation:

$$\frac{\partial z}{\partial t} = \kappa \frac{\partial^2 z}{\partial x^2}; \quad \kappa = 0.1 \tag{6}$$

with boundary conditions

$$z(0,t) = 3, \quad \frac{\partial z(L,t)}{\partial x} = 0. \tag{7}$$

These are Dirichlet in the first point and Neumann homogeneous in the last point. The spatial domain is $\mathscr{V} = \{x/x \in [0,\pi]\}$. The initial conditions are chosen as $z(x,0) = -x^2 + 2\pi x + 3$

As mentioned in section 2, the first step to obtain the basis functions in the POD method is to generate a set of snapshots (measurements of the real system or simulation data) representative of the behaviour of the system. In this case, the snapshots were obtained by means of simulation data taken each 0.1 unit of time till 150 and they were saved in the file `data_simulation_Ex1.mat`. Thus, in the code we first load the snapshots:

```
load data_simulation_Ex1
```

After this, we compute the FEM matrices as indicated before and compute the basis functions by typing:

```
[pods]  = matpod(Z , MM , 'd' , 99.999);
Phi     = pods.phi;
neig    = size(Phi , 2);
```

It is easy to see, by means of the parameter `neig`, that 3 basis functions are enough to capture the 99.999% of the energy. The next step is to project Eqn (6) over the basis functions, this is:

$$\int_{\mathscr{V}} \Phi \frac{\partial z}{\partial t} d\xi = \int_{\mathscr{V}} \kappa \Phi \frac{\partial^2 z}{\partial x^2} d\xi$$

which, taking into account the boundary conditions, the orthonormality of basis functions and the relationships of Table 1 can be rewritten in discrete form as:

$$m_t = -\Phi^T MMMM^{-1}(\kappa DM + BM)\Phi m = -\Phi^T(\kappa DM + BM)\Phi m$$

with initial conditions:

$$m_0 = \Phi^T M M z_0$$

So in the Matlab© code, we can define a new matrix `Sp_oper` and a new vector `G` of the form:

```
Sp_oper = - Phi'*(k*DM + BM)*Phi;
G       = Phi'*G;
```

so that the system of ODEs remains:

```
dm = Sp_oper*m + G;
```

Note that now we are solving 3 ODEs instead 21. The maximum error at $t = 0$ is about 2.5% and it decreases very fast with time, in fact, for $t > 5$ the maximum error is always below 0.2%. If we consider the same example but capturing the 99.9% of the energy, the number of basis functions will be two. In this case the maximum error for all $t > 5$ is lower than the 1%.

The complete code is included in the Matlab© scripts `Ex1_POD.m` and `ode_Ex1_POD.m`.

3.2 Reaction-Diffusion-Convection problem

In this example reaction and convection terms are considered. The model equations are

$$\frac{\partial z}{\partial t} = \kappa \frac{\partial^2 z}{\partial x^2} - v\frac{\partial z}{\partial x} + f(z); \qquad f(z) = z - z^3 \tag{8}$$

where $\kappa = 0.5$ and $v = 0.01$, with the usual Danckwerts boundary conditions

$$\vec{\mathbf{n}} \cdot \kappa \vec{\boldsymbol{\nabla}} z\Big|_{x=0} = v(z_{in} - z|_{x=0}) \iff \kappa \left.\frac{\partial z}{\partial x}\right|_{x=0} = v(z|_{x=0} - z_{in}) \tag{9}$$

$$\vec{\mathbf{n}} \cdot \vec{\boldsymbol{\nabla}} z\Big|_{x=L} = 0, \tag{10}$$

with z_{in} being the concentration in the inlet stream ($z_{in} = 100sin(7t) + 50$). The spatial domain is $\mathscr{V} = \{x/x \in [0, \pi]\}$. The initial conditions are $z(x, 0) = (-x^2 + 2\pi x + 2\pi k/v + 50)/150$.

As in the previous case the first step is to obtain the basis functions from the snapshots, which were previously generated each 0.001 units of time till $t = 2$ and saved in the file `data_simulation_Ex2`:

```
load data_simulation_Ex2
```

Now we define the spatial domain and the discretization and compute the FEM matrices. After these steps, the basis functions are computed as follows:

```
[pods]   = matpod(Z , MM , 'd' , 99.99);
Phi      = pods.phi;
neig     = size(Phi , 2);
proj_op = Phi'*MM;          % Projection operator
```

In this case a 99.99% of the energy implies using 3 basis functions. The next step is to project the spatial operator and the initial conditions.

```
Sp_oper = - Phi'*(k*DM + v*BM + v*CM)*Phi;


% Initial conditions
z0       = (-xe.^2 + 2*pi*xe + 2*pi*k/v + 50)/150;
m0       = proj_op*z0;
```

In the script where the ODEs are defined we need to project the boundary conditions and the nonlinear term[1]:

[1] Note that $\Phi^T MMM^{-1}G = \Phi^T G$

```
 % Recovery the original field
z     = Phi*m;

% Boundary conditions
zin   = 100*sin(7*t)+50;     % Inlet concentration
G(1)  = v*zin;               % Boundary condition in the first point
GG    = Phi'*G;              % Projection of the BC

% Nonlinear term
f     = z - z.^3;
f     = proj_op*f;           % Projection of the nonlinear function
```

With 3 basis functions the maximum error is always lower than 3% and for $t > 0.1$ decreases to 1.6%. In order to capture the 99.999% of the energy four basis functions are required. In this case the maximum error is always lower than 0.7% and for $t > 0.1$ decreases to 0.3%.

An interesting exercise here is to check what happens if the snapshots are selected in other way. For instance, instead using a $\Delta t = 0.001$ between two consecutive measurements try with $\Delta t = 0.2$ or with a final time of $t = 0.3$ instead $t = 2$.

The complete code is included in the Matlab© scripts `Ex2_POD.m` and `ode_Ex2_POD.m`.

3.3 The Burgers equation

The well known Burgers equation presents the following form:

$$\frac{\partial z}{\partial t} = \frac{\partial(-0.5z^2)}{\partial x} + \mu\frac{\partial^2 z}{\partial x^2} = -z\frac{\partial z}{\partial x} + \mu\frac{\partial^2 z}{\partial x^2} \tag{11}$$

The spatial domain is $\mathscr{V} = \{x/x \in [0,1]\}$ and Dirichlet boundary conditions are considered at both sides of the domain. The value for the boundary and initial conditions is obtained through the analytical solution given by:

$$z = \frac{0.1ea + 0.5eb + ec}{ea + eb + ec},$$

with:

$$ea = \exp\left(\frac{0.05}{\mu}(x - 0.5 + 4.95t)\right); \quad eb = \exp\left(\frac{0.25}{\mu}(x - 0.5 + 0.75 * t)\right).$$

$$ec = \exp\left(\frac{0.5}{\mu}(x - 0.375)\right).$$

These equations are were written into the `burgers_exact.m` function which is included in the toolbox.

The procedure is the same than in the above examples. First we obtain the basis functions from the snapshots taken by means of a FEM simulation. In this case the time interval between two consecutive measurements was 0.001 units of time and the simulation data were save in a Matlab© file `data_simulation_burgers.mat`.

```
load data_simulation_burgers
```

After defining the spatial grid, we compute the FEM matrices and the basis functions.

```
%... call to the matfem function for computing the FEM matrices
[MM, DM, CM, BM]   = matfem(x, 'dir', 'dir', 'MM', 'DM', 'CM', 'BM');
inv_MM             = inv(MM);

% Computation of the basis functions
[pods]  = matpod(Z , MM , 'd' , 99.9999);
Phi     = pods.phi;
neig    = size(Phi , 2);
proj_op = Phi'*MM;          % Projection operator
```

In this problem, the only difference with respect to the others is the convection term. The projection of this term is carried out in the same manner as if it was a nonlinear term:

```
fx     = Grad_op*(0.5*z.^2);
pfx    = proj_op*fx;        % Projection of the convection term
```

It must be remarked that due to the special form of the convection term and the boundary conditions, this problem dissipates energy at a very low rate. This fact can be seen by comparing the relation between eigenvalues:

$$\frac{\lambda_1}{\lambda_2} \approx 20, \quad \frac{\lambda_2}{\lambda_3} \approx 3.5, \quad \frac{\lambda_3}{\lambda_4} \approx 2.5,$$

while in other problems which dissipate much more energy, the relation is much bigger. For instance in the Reaction-Diffusion-Convection problem presented in this manual, the relation is:

$$\frac{\lambda_1}{\lambda_2} \approx 410, \quad \frac{\lambda_2}{\lambda_3} \approx 9, \quad \frac{\lambda_3}{\lambda_4} \approx 24,$$

This fact translates into a large number of basis functions to be employed for a good representation. Capturing the 99.9999% of the energy implies using 73 basis functions and the maximum error is around 7.8%. Augmenting the number of basis functions to 87, the energy captures is 99.99999% and the maximum error decreases to 2.8%.

References

Berkooz, G., Holmes, P., and Lumley, L. (1993). The Proper Orthogonal Decomposition in the analysis of turbulent flows. *Ann. Rev. Fluid Mech.*, 25:539–575.

García, M. R. (2008). *Identification and Real Time Optimisation in the Food Processing and Biotechnology Industries.* PhD thesis, University of Vigo, Spain. Available online at http://digital.csic.es/handle/10261/4662.

Sirovich, L. (1987). Turbulence and the dynamics of coherent structures. Part I: Coherent structures. *Quaterly of Appl. Math.*, 45(3):561–571.

Vilas, C. (2008). *Modelling, Simulation and Robust Control of Distributed Processes: Application to Chemical and Biological Systems*. PhD thesis, University of Vigo, Spain. Available online at http://digital.csic.es/handle/10261/4236.