User's Manual for the Function `matlsd`. A MATlab function for the Laplacian Spectral 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. One interesting option when dealing with homogeneous boundary conditions is the Laplacian Spectral Decomposition (LSD) technique.

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 LSD, they are obtained by the spectral decomposition of the Laplacian operator. This is:

$$\Delta\phi_i = -\lambda_i\phi_i, \quad i = 1, ..., neig \tag{1}$$

where $neig$ is the number of basis functions considered.

As shown in the `matfem` user's manual, there exists a relationship between the FEM matrices and spatial derivatives and integrals. For the sake of clarity, such relationships are rewritten here in Table 1.

Using these matrices, Eqn (1) can be approximated as follows:

$$\mathcal{M}\mathcal{M}^{-1}(\mathcal{D}\mathcal{M} + \mathcal{B}\mathcal{M})\phi_i = -\lambda_i\phi_i, \quad i = 1, ..., neig \tag{2}$$

which is the expression used in the `matlsd` function.

A more detailed description of this technique can be found in the literature (Christofides, 2001; Vilas, 2008; García, 2008).

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

	Continuous	Discrete
1	$\int_{\mathscr{V}} g(\boldsymbol{\xi}) f(\boldsymbol{\xi}) d\boldsymbol{\xi}$	$\mathcal{G}^T \mathcal{M} \mathcal{M} \mathcal{F}$
2	$\int_{\mathscr{V}} g(\boldsymbol{\xi}) \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i} \, \mathrm{d}\boldsymbol{\xi}$	$\mathcal{G}^T \mathcal{C} \mathcal{M}_i \mathcal{F}$
3	$\int_{\mathscr{V}} g(\boldsymbol{\xi}) \Delta f(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}$	$-\mathcal{G}^T \left(\mathcal{D} \mathcal{M} + \mathcal{B} \mathcal{M} \right) \mathcal{F}$
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 $f(\boldsymbol{\xi})$.

2 The Function `matlsd`

This function will compute the basis functions using the LSD technique. The way of calling the function is

```
[phi , lambda]  = matlsd(MM, DM, BM, k, v);
```

where the input and output parameters are

```
Input variables:
MM:   Mass matrix of the FEM
DM:   Diffusion matrix of the FEM
BM:   Boundary (homogeneous part) matrix of the FEM
k:    Diffusivity
v:    Velocity


Output variables:
phi:    Eigenvectors of the laplacian operator
lambda: Eigenvalues of the laplacian operator
```

If the velocity is not introduced, the function takes automatically the value $v = 1$. It is important to remark that this technique only works with homogeneous boundary conditions. When non ho-

mogeneous BC are considered, we can define a state transformation in order to obtain an equivalent system with homogeneous BC.

3 Examples

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

3.1 A simple diffusion problem

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

$$\frac{\partial z}{\partial t} = \kappa \frac{\partial^2 z}{\partial x^2}; \qquad \kappa = 0.1 \tag{3}$$

with boundary conditions

$$z(0,t) = 3, \qquad \frac{\partial z(L,t)}{\partial x} = 0. \tag{4}$$

Note that in this example non-homogeneous boundary conditions are considered. The LSD method does not allow working directly with this kind of BC, therefore a state transformation is requiered to obtain an equivalent system with homogeneous BC. In this case, the following transformation is defined:

$$\overline{z} = z - z(0,t) = z - 3.$$

Note that

$$\frac{\partial \overline{z}}{\partial t} = \frac{\partial z}{\partial t}, \qquad \frac{\partial \overline{z}}{\partial x} = \frac{\partial z}{\partial x}, \qquad \frac{\partial^2 \overline{z}}{\partial^2 x} = \frac{\partial^2 z}{\partial^2 x}.$$

Using these relations, system (3) can be rewritten as:

$$\frac{\partial \overline{z}}{\partial t} = \kappa \frac{\partial^2 \overline{z}}{\partial x^2}; \qquad \kappa = 0.1 \tag{5}$$

with boundary conditions

$$\overline{z}(0,t) = 0, \qquad \frac{\partial \overline{z}(L,t)}{\partial x} = 0. \tag{6}$$

We can, now, apply the LSD technique to this problem and recover the original field by $z = \overline{z} + 3$. 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 indicated before, the FEM matrices are employed to approximate the continuous eigenvalue problem by a discrete version. So the first step is to obtain the FEM matrices as indicated in the `matfem` user's manual. After this we can compute the basis functions by:

```
[Phi , lambda]  = matlsd(MM, DM, BM, k);
neig            = 3;    % Number of basis functions
Phi             = Phi(: , 1:neig);
lambda          = lambda(1:neig , 1:neig);
```

In this case we choose to use only 3 basis functions. The next step is to project Eqn (3) 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, the relationships of Table 1 and Eqn (2) can be rewritten in discrete form as:

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

with initial conditions:

$$m_0 = \Phi^T MM\bar{z}_0$$

So that the system of ODEs remains:

```
dm = -lambda*m;
```

Note that now we are solving 3 ODEs instead 21. The maximum error at $t = 0$ is about 1% and it decreases very fast with time, in fact, for $t > 5$ the maximum error remains below 0.004%. If we consider the same example with two basis functions, the maximum error for all $t > 5$ is lower than the 0.1%.

The complete code is included in the Matlab© scripts `Ex1_LSD.m` and `ode_Ex1_LSD.m`.

3.2 Reaction-Diffusion-Convection problem

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

$$\frac{\partial z}{\partial t} = \kappa \frac{\partial^2 z}{\partial x^2} - v\frac{\partial z}{\partial x} + f(z); \qquad f(z) = z - z^3 \tag{7}$$

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

$$\vec{\mathbf{n}} \cdot \kappa \vec{\boldsymbol{\nabla}} z\Big|_{x=0} = v(z_{in} - z|_{x=0}) \iff \kappa \left.\frac{\partial z}{\partial x}\right|_{x=0} = v(z|_{x=0} - z_{in}) \tag{8}$$

$$\vec{\mathbf{n}} \cdot \vec{\boldsymbol{\nabla}} z\Big|_{x=L} = 0, \tag{9}$$

with z_{in} being the concentration in the inlet stream ($z_{in} = 100\sin(7t) + 50$). 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 example the boundary conditions are not homogeneous in the first point so a state transformation is required. Defining a new state variable:

$$\overline{z} = z - z_{in} = z - (100\sin(7t) + 50)$$

and taking into account

$$\frac{\partial \overline{z}}{\partial t} = \frac{\partial z}{\partial t} - \frac{\partial z_{in}}{\partial t}, \quad \frac{\partial \overline{z}}{\partial x} = \frac{\partial z}{\partial x}, \quad \frac{\partial^2 \overline{z}}{\partial^2 x} = \frac{\partial^2 z}{\partial^2 x},$$

Eqn (7) can be rewritten as

$$\frac{\partial \overline{z}}{\partial t} = \kappa \frac{\partial^2 \overline{z}}{\partial x^2} - v\frac{\partial \overline{z}}{\partial x} + f(\overline{z}) - \frac{\partial z_{in}}{\partial t}; \quad f(\overline{z}) = (\overline{z} + z_{in}) - (\overline{z} + z_{in})^3, \tag{10}$$

with homogeneous boundary conditions:

$$\kappa \left.\frac{\partial \overline{z}}{\partial x}\right|_{x=0} = v\ \overline{z}|_{x=0}\,; \quad \left.\overrightarrow{\mathbf{n}} \cdot \overrightarrow{\boldsymbol{\nabla}} \overline{z}\right|_{x=L} = 0, \tag{11}$$

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

```
[Phi , lambda]  = matlsd(MM, DM, BM, k, v);
neig            = 7;    % Number of basis functions
Phi             = Phi(: , 1:neig);
lambda          = lambda(1:neig , 1:neig);
```

In this case we choose 7 basis functions. The next step is to project the spatial operator and the initial conditions.

```
Sp_oper = - (lambda + v*Phi'*CM*Phi);
proj_op = Phi'*MM;


% Initial conditions
z0              =  (-xe.^2 + 2*pi*xe + k*2*pi/v + 50)/150;
z_transform     = zeros(steps+1 , 1);
z_transform(1)  = 100*sin(7*tlist(1)) + 50;
z0_bar          = z0 - z_transform(1);
m0              = proj_op*z0_bar;
```

In the script where the ODEs are defined we need to project the nonlinear term and the term associated with the state transformation:

```
% Recovery the transformed field
z_bar   = Phi*m;

% Recovery the original field
z_trans = 100*sin(7*t) + 50;
z       = z_bar + z_trans;

% Derivative of the z transform with respect the time
dztr_dt = 700*cos(7*t);

% Nonlinear term
f = z - z.^3 - dztr_dt;
f = proj_op*f;

% ODE construction
dm = Sp_oper*m + f;
```

With 8 basis functions the maximum error is always lower than 1%. If we use 7 basis functions for the projection, the maximum error will increase to 1.7%.

The complete code is included in the Matlab© scripts `Ex2_LSD.m` and `ode_Ex2_LSD.m`.

3.3 The Burgers equation

The well known Burgers equation presents the following form:

$$\frac{\partial z}{\partial t} = \frac{\partial(-0.5z^2)}{\partial x} + \mu\frac{\partial^2 z}{\partial x^2} = -z\frac{\partial z}{\partial x} + \mu\frac{\partial^2 z}{\partial x^2} \tag{12}$$

The spatial domain is $\mathscr{V} = \{x/x \in [0,1]\}$ and Dirichlet boundary conditions are considered at both sides of the domain. Again, we need to transform the original problem with non homogeneous BC into an equivalent one with homogeneous BC. In the case of the FEM and POD, the values for the boundary conditions were computed through relatively complex expressions. If we apply the same BC now, the transformation will require to compute the time and spatial derivatives of such expressions. In order to avoid this we should note that, for the considered value of μ, the BC can be approximated as follows:

$$z(0,t) = 1; \quad z(L,t) = 0.1.$$

Thus, defining the state transformation

$$\overline{z} = z - (-0.9x + 1);$$

we obtain homogeneous boundary conditions.

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

```
[Phi , lambda]  = matlsd(MM, DM, BM, mu);
neig            = 87;     % Number of basis functions
Phi             = Phi(: , 1:neig);
lambda          = lambda(1:neig , 1:neig);
proj_op         = Phi'*MM;
```

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
```

As explained in the `matpod` user's manual, this problem dissipates energy at a very low rate. This fact translates into a large number of basis functions to be employed for a good representation. Using the same number of basis functions than in the POD case leads to a maximum error of 34%[1]. Augmenting the number of basis functions to 150, the maximum error decreases to 3%, but the dimensionallity of the problem is almost the same as in the FEM, so the reduction is quite poor.

References

Christofides, P. D. (2001). *Nonlinear and Robust Control of PDE Systems: Methods and Applications to Transport-Reaction Processes*. Birkhäuser, Boston.

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.

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.

[1] It should be remarked that this maximum error only appears in very specific regions of the spatial and time domains, in the other regions the error remains below the 5%