Universität Konstanz *WS 11/12*
Fachbereich Mathematik und Statistik
S. Volkwein, O. Lass

Übungen zu **Modellreduktion mit Proper Orthogonal Decomposition**

http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/

Submission: 13.01.2012, 10:00 o'clock
Codes by E-Mail and Reports together with the Homework

Program 3 (6 Points)

We consider Problem (1) of Program 1 and the POD basis computed in Program 2. Write a code solving the heat equation by generating a reduced order model with the POD basis. Further, we introduce the relative error as

$$\epsilon_{rel} = \int_0^T \frac{\|y(t) - y^\ell(t)\|_W}{\|y(t)\|_W}\,\mathrm{d}t. \tag{1}$$

In our case y corresponds to the finite difference solution and y^ℓ to the solution using the reduced order model. We consider the following experiments:

1) For $c = 0.05$ we compute the POD basis of order $\ell = 1, \ldots, 15$ and solve the reduced order model. How does the relative error behave for increasing values of ℓ?

2) We compute the POD basis of order $\ell = 10$ for $c = 0.05$. Then we solve the reduced order model for $c_{test} = \{0.01, 0.02, \ldots, 0.09, 0.1\}$. How does the relative error behave for the different values of c_{test}?

Structure your code as follows:

`main_ell` ... Main script file to realize experiment one. All parameters are set in this file and the desired plots are generated.

`main_c` ... Main script file to realize experiment two. All parameters are set in this file and the desired plots are generated.

`[Y,time] = solve_heat_pod(Psi,W,c,A,h,tstep,Y0,method)` ... Solves the heat equation (1) of Program 1 using the POD basis `Psi` and the weight matrix `W` for the inner products. The other variables are the same as for the function `solve_heat_fdm` in Program 1. The return values are a matrix `Y` with columns containing the initial condition and solution to (1) of Program 1 in the inner points and `time` a vector containing the timesteps.

For the numerical experiments discretize the relative error appropriately. The parameters are set to

`n = 100`, `tsteps = 100`, `method = 'RS'`, `wtype = 'L2'` and `pod = 'eig'`.

As initial condition y_0 we choose

$$y_0(x) = \begin{cases} 1, & \text{on } (0.25, 0.75) \times (0.25, 0.75), \\ 0, & \text{otherwise.} \end{cases}$$

All needed functions from the previous Programs should be used. In the written report give details to your implementation and your observation. How does the FDM code compare to the POD code in performance? What speedup factor do you observe? Interpret all the obtained results. When plotting the relative error use the semi-log scale. Submit your code as a ZIP or TAR/ZIP archive containing one folder named `your_lastname_prog03`.