Homework 2

In this assignment you will assemble a model for a heat conducting bar exactly as the example illustrated in class. You will implement various model order reduction techniques, and you will simulate the resulting compact dynamical model comparing the performance of the different techniques.

You can assume the standard (normalized, 1D) heat diffusion equation:

$$\frac{\partial T(x,t)}{\partial t} = \frac{\partial^2 T(x,t)}{\partial x^2} + h(x)\, u(t), \qquad x \in [0,1]$$

where $T(x, t)$ is the temperature at location x along the bar at time t. A heat lamp $u(t)$ is "illuminating" the bar resulting in a localized special heat source distribution $h(x)$ along the bar. You can assume zero initial conditions at $t = 0$. As boundary conditions, assume that the temperature is vanishing at the left edge $T(0, t) = 0$, and that the right edge is insulated from the environment $\frac{\partial T(x,t)}{\partial x}$=0 at $x = 1$.

Part a. Write a matlab function script that assembles a dynamical model of the form:

$$E\frac{dz(t)}{dt} = Az(t) + Bu(t)$$

Hint: you can choose to first generate an intermediate "equivalent RC circuit" representation and then impose conservation and constitutive equations combined in a nodal analysis framework. Alternatively, you can use a finite difference (or finite element) direct discretization of the given heat diffusion equation.

For your method of choice:

- Specify the matrices *E,A,B* explaining <u>very</u> briefly the meaning of the entries;
- Specify the physical dimensions of the entries.
- Give a brief physical interpretation for the components in your state vector *z(t).*

The arguments of the matlab function should be the number *N* of discretization points. Construct also an auxiliary function `heat_source(x)` that returns the spatial distribution of the heat source at any arbitrary location *x*, and use it in the construction of the matrices *E,A,B*.

Part b. Assume the output of your dynamical model can be expressed as a linear combination of the state components $y(t) = C^T z(t)$. Explain how you would pick the matrices (or vectors) *B* and *C,* in order to model the following different scenarios:

1. *Input:* heat only at left end of bar, *Output:* temperature at right end of the bar;
2. *Input:* heat only at left end of bar, *Output:* average temperature of the bar;
3. *Input:* uniform heating on the bar, *Output:* temperature at right end of the bar;
4. *Input:* uniform heating on the bar, *Output:* average temperature of the bar.

Part c. Plot the outputs to the following inputs for each of the scenarios in part (b) using N=500 and your *q*:

$$u_1(t) = \begin{cases} 1 & t \ge 0 \\ 0 & t < 0 \end{cases}$$

Use Backward Euler for time integration. Be sure to pick an appropriate time step and *explain your choice.* Simulate your model until it reaches steady state (t=t_{stop}).

Part d. Write a matlab routine `PlotFreqResp(w,E,A,B,C)` which accepts *ω,E,A,B,C* as input arguments, and plots the frequency response of your dynamical model. Here *ω* refers

to a vector of frequencies in [rad/sec]. Inside your function use $s = j\,\omega$. Assume your state $z(t)$ has N=500 components. Plot the frequency response for all of the cases in part (b). Your code should generate one plot with both real and imaginary parts of the response (plotted in different colors).
Hint: to plot the frequency response, try also the command `semilogx(x,y)`. You can generate a frequency vector using the command: ω =`logspace(-8,4,500)`

Part e. Reduce your dynamical system from part (a) using the <u>eigen-mode truncation method</u> for all of the cases in part (b). Use N = 500 as original order and pick a '*reasonably small*' number of states q for your reduced model. Clearly and briefly explain how you picked q. What factors did you look at?
Hint: you may use $q = \{1, 5, 10, 50, 100, ...\}$ to see some kind of trend. Plot the frequency responses of the original system and of the reduced system on the same plot for comparison.

Part f. Repeat part (c) for the reduced dynamical mode with different values of q, and compare the results to the full dynamical system with N=500. How much speed-up do you observe between the time domain simulations of your reduced dynamical model compared to the original dynamical model?
Repeat for the following input $u_2(t) = \sin(0.01t)$. Simulate both the reduced and original models up to t_{stop}=10000 seconds, measure the speedup. Explain your results.

Part g. Repeat part (e) and (f) using <u>balanced truncation</u>. You can use matlab's built-in functions such as `reduce` or `balancmr` to do the reduction (i.e. you are not required to implement your own balanced truncation). Note that once you have the A,B,C matrices, you can form the linear dynamical system in state space by using the command `H=ss(A,B,C',0)`.

Part g. Implement a function to perform reduction using <u>moment matching</u>. (you can use any technique: e.g. single point expansion, or multipoint expansion). Repeat part (e) and (f) using your moment matching routine.

Part h. Compare the three techniques. A table with few columns would suffice.