

# MODEL REDUCTION EXPERIMENTS

## PART I: BASIC SYSTEMS THEORY

M. EMBREE AND D. C. SORENSEN\*

**Abstract.** This document describes a set of experiments for students in CAAM 651 at Rice University, Spring 2011, that aims to foster insight into the structure and behavior of a particular linear dynamical system. Students should complete these exercises for one course credit.

**0. Instructions.** The first section of CAAM 651 concentrated on fundamental concepts from the theory of linear systems: (i) the solution of the basic dynamical system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t),$$

and the matrix exponential; (ii) solutions to the control problem

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t);\end{aligned}$$

(iii) the notions of energy, controllability and observability; (iv) the transfer function. To develop a deeper appreciation for these concepts, we invite you to complete the following set of investigations for a particular system of your choosing. We have collected several systems of varying dimensions, drawn from test collections and research projects, mostly inspired by applications. Select a system from this set – even better, choose one from your own research if possible – and complete the tasks outlined below.

This list of tasks is intentionally open-ended. The investigations you conduct will be guided by your own instinct and insight, along with any restrictions imposed by the size and properties of the matrix you select. Be frank if you are observing a phenomenon you cannot understand; feel free to include additional experiments that might give you greater insight. If the size of the matrix you have chosen prohibits you from completing some aspects of the assignment, please adapt the problem to something more reasonable for your system. We are happy to discuss strategies with you to help facilitate large-scale computations.

**0.1. A note about your write-up.** Please submit a typed report describing the experiments you conduct, as directed by the tasks below. We recommend that you use the L<sup>A</sup>T<sub>E</sub>X typesetting system to collect and organize your results. In particular, you can download a L<sup>A</sup>T<sub>E</sub>X template, `modred_tasks1.tex`, from the class website, which you can then edit; this will give our reports a uniform appearance. (This document uses the SIAM journal macros. These macros and supporting files, along with all the graphics files used in this report, will be collected in one `.zip` file on the class website).

**1. Compute solutions to the system  $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$ .** Select a unit vector  $\mathbf{x}_0$  with random entries (use `randn` and normalize), and produce a `semilogy` plot showing  $\|\mathbf{x}(t)\|$  versus  $t$ . Choose a scale for  $t$  so that the asymptotic behavior of the system is clear. Superimpose on this plot the results obtained for several other random unit

---

\*Department of Computational and Applied Mathematics, Rice University, 6100 Main Street—MS 134, Houston, Texas 77005–1892 ([embree@rice.edu](mailto:embree@rice.edu), [sorensen@rice.edu](mailto:sorensen@rice.edu)).

vectors  $\mathbf{x}_0$ . (Further extensions to this plot are suggested in problems 2, 3, and 6 below.)

How should one best compute  $\mathbf{x}(t)$  for each  $t$ ? Here are several options.

- (a) If your matrix is of modest dimension, you can compute  $e^{t\mathbf{A}}$  using the `expm` command in MATLAB (not `exp!`), then form  $\mathbf{x}(t) = e^{t\mathbf{A}}\mathbf{x}_0$ .
- (b) Alternatively, particularly for large-scale systems, you can use your favorite ODE integrator (e.g., forward or backward Euler, Runge–Kutta, MATLAB’s `ode45`, etc.) to approximately solve  $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$ . (Be sure to choose your time-step or tolerance sufficiently small that the solution is not polluted by stability/accuracy effects.)

**2. Compute/approximate the spectrum of  $\mathbf{A}$ .** Produce a plot of the eigenvalues of the matrix  $\mathbf{A}$  in the complex plane. Ultimately we are interested in several aspects of the spectrum, which might take two or three plots to properly communicate:

- the extent of the entire spectrum;
- the scale of the imaginary versus real parts of the eigenvalues (try `axis equal` in MATLAB).
- fine aspects of the spectrum close to the imaginary axis, particularly the value of the spectral abscissa,  $\alpha(\mathbf{A}) = \max_{\lambda \in \sigma(\mathbf{A})} \operatorname{Re} \lambda$ ;

If your matrix  $\mathbf{A}$  is Hermitian, then you might also produce a plot that illustrates the density of the spectrum, e.g., `plot(eig(A))`.

If your matrix is large in size, you will only be able to compute a modest portion of the spectrum. Call MATLAB’s `eigs` command (based on the implicitly restarted Arnoldi algorithm) several times with various different options to get an overall impression of the spectrum.

Be sure to report the value of the spectral abscissa. For extra insight, superimpose a plot of  $e^{t\alpha(\mathbf{A})}$  versus  $t$  on your results from problem 1.

**3. Analyze the potential for transient growth.** We have seen that the initial behavior of the system  $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$  is governed by the rightmost extent of the numerical range,

$$W(\mathbf{A}) = \{\mathbf{x}^* \mathbf{A} \mathbf{x} : \|\mathbf{x}\| = 1\}.$$

If  $\mathbf{A}$  is Hermitian, this set will simply be the real line segment connecting the leftmost and rightmost eigenvalues. For non-Hermitian  $\mathbf{A}$  this set can extend far into the right half plane, even when all eigenvalues of  $\mathbf{A}$  are in the left half-plane.

Compute (or approximate) the numerical range of your system. Use your plots in Question 1 to confirm the fact that

$$\max_{\|\mathbf{x}_0\|=1} \lim_{t \downarrow 0} \frac{d}{dt} \|\mathbf{x}(t)\| = \lim_{t \downarrow 0} \frac{d}{dt} \|e^{t\mathbf{A}}\| = \omega(\mathbf{A}),$$

where the *numerical abscissa* is the rightmost extent of  $W(\mathbf{A})$ :

$$\omega(\mathbf{A}) = \max_{z \in W(\mathbf{A})} \operatorname{Re} z.$$

Report the value of  $\omega(\mathbf{A})$ . Does it differ much from  $\alpha(\mathbf{A})$ ? If so, you might superimpose a plot of  $e^{t\omega(\mathbf{A})}$  versus  $t$  on your plot from question 1.

Here are a few suggestions for ways you can compute (or approximate)  $W(\mathbf{A})$  and  $\omega(\mathbf{A})$ .

- (a) If  $\mathbf{A}$  is sufficiently small, use the `fv.m` routine from Nick Higham's Matrix Computation Toolbox to approximate the numerical range (or edit the code given below). You can control the number of  $\theta$  values that are used; each one will require the solution of a dense Hermitian eigenvalue problem.
- (b) If  $\mathbf{Q} \in \mathbb{C}^{n \times k}$  is any rectangular matrix with orthonormal columns, then  $W(\mathbf{Q}^* \mathbf{A} \mathbf{Q}) \subseteq W(\mathbf{A})$ . Taking  $\mathbf{Q} = \mathbf{V}_k$  from the Arnoldi decomposition (e.g., computed using the `ArnoldiC` code on the class website), we arrive at a 'Ritz' inner approximation:

$$W(\mathbf{V}_k^* \mathbf{A} \mathbf{V}_k) = W(\mathbf{H}_k) \subseteq W(\mathbf{A}).$$

Use techniques from (a) to compute  $W(\mathbf{H}_k)$ .

- (c) If  $\mathbf{A}$  is large and you wish to compute  $W(\mathbf{A})$  with some accuracy, you can do so by using `eigs` to compute the extreme eigenvalues of the Hermitian matrix

$$\frac{1}{2}(\mathrm{e}^{i\theta} \mathbf{A} + \mathrm{e}^{-i\theta} \mathbf{A}^*)$$

for a modest number of  $\theta$  values in  $[0, \pi)$ . For each value of  $\theta$ , we obtain two lines that sharply bound the extent of the numerical range of  $\mathbf{A}$ . (For example, we are most interested in knowing if  $0 \in W(\mathbf{A})$ ; if  $\mathbf{A}$  is real, this question can be answered from the  $\theta = 0$  computation alone.) By varying  $\theta \in [0, \pi)$ , you will trace out a bound for the numerical range; see Figure 3.1 for approximations of the numerical range of the matrix

$$(3.1) \quad \mathbf{A} = \begin{bmatrix} 0 & 3i \\ 1 & 2i \end{bmatrix}.$$

It would be interesting to compare the result of this approach to that obtained from the Ritz approximation in part (b.)

An example is shown in Figure 3.1. To produce plots like these, you could run a loop over the following code with various values of `th`  $\in [0, 2\pi)$ . For large, sparse problems, you will want to replace the call to `eig` with calls to `eigs` to compute the leftmost and rightmost eigenvalues of the sparse, Hermitian matrix  $\mathbf{H}$ .

```
H = .5*(exp(1i*th)*A+exp(-1i*th)*A');
ew = eig(H);
l1 = exp(-1i*th)*(min(ew)+1i*[-scl scl]);
l2 = exp(-1i*th)*(max(ew)+1i*[-scl scl]);
plot(l1,'k--','linewidth',1.25), hold on
plot(l2,'k--','linewidth',1.25)
```

(The scaling factor `scl` is problem-dependent; take, e.g., `scl=norm(A)`.)

- (d) Download the MATLAB package EigTool from:

<http://www.comlab.ox.ac.uk/pseudospectra/eigtool/>

You can use EigTool to compute both the numerical range  $W(\mathbf{A})$  and the  $\varepsilon$ -pseudospectra,

$$\sigma_\varepsilon(\mathbf{A}) = \{z \in \mathbb{C} : z \in \sigma(\mathbf{A} + \mathbf{E}) \text{ for some } \mathbf{E} \in \mathbb{C}^{n \times n} \text{ with } \|\mathbf{E}\| < \varepsilon\}.$$

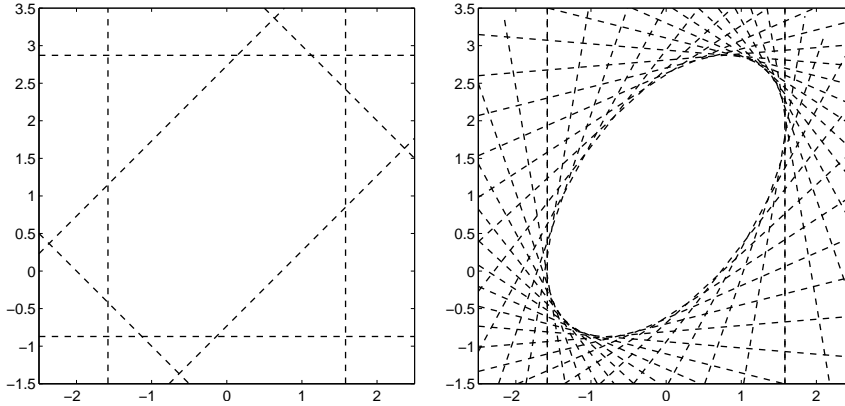


FIG. 3.1. Bounds on the numerical range for the matrix (3.1) using  $\theta = 0, \pi/4, \pi/2$  (left) and twenty values of  $\theta \in [0, \pi]$  (right).

**4. Controllability.** Suppose that your system has one input,  $m = 1$ , in which case we write  $\mathbf{B} = \mathbf{b} \in \mathbb{C}^n$ . Recall that the pair  $(\mathbf{A}, \mathbf{b})$  is *controllable* if the matrix

$$\mathbf{K} = [\mathbf{b} \ \mathbf{A}\mathbf{b} \ \cdots \ \mathbf{A}^{n-1}\mathbf{b}]$$

has full rank,  $\text{rank}(\mathbf{K}) = n$ . This condition, as stated with the matrix  $\mathbf{K}$ , is difficult to check in practice.

To investigate controllability, consider the following two experiments. Please conduct each experiment twice: once for a column  $\mathbf{b} \in \mathbb{C}^n$  of the matrix  $\mathbf{B} \in \mathbb{C}^{n \times m}$  in your system, and a once for an artificial  $\mathbf{b} \in \mathbb{C}^n$  constructed as some linear combination of twenty eigenvectors of  $\mathbf{A}$ . This choice of  $\mathbf{B}$  will make  $(\mathbf{A}, \mathbf{b})$  uncontrollable. Can you detect this numerically? Pursue the two following approaches.

- (a) For  $k = 1, \dots, n$ , construct the matrix

$$\mathbf{K}_k = [\mathbf{b} \ \mathbf{A}\mathbf{b} \ \cdots \ \mathbf{A}^{k-1}\mathbf{b}] \in \mathbb{C}^{n \times k}$$

and compute the condition number  $\text{cond}(\mathbf{K}_k)$ . (This quantity, the ratio of the largest to smallest singular values of  $\mathbf{K}_k$ , indicates the distance to rank-deficiency: in principle, it should be infinite when  $\mathbf{K}_k$  is rank-deficient, and large when it is nearly so. You can compute this quantity using the `cond` command in MATLAB.)

Produce a `semilogy` plot of  $k$  versus  $\text{cond}(\mathbf{K}_k)$ .

- (b) Use the `ArnoldiC` code from the class website to compute an  $n$ -step Arnoldi factorization of  $\mathbf{A}$  with starting vector  $\mathbf{b}$ . Produce a semilogy plot showing the magnitude of the subdiagonal entries  $h_{k+1,k}$  of the upper Hessenberg matrix  $\mathbf{H}_n \in \mathbb{C}^{n \times n}$  as a function of the index  $k$ . If  $|h_{k+1,k}|$  is zero (very small) for any value of  $k$ , that will indicate that  $\mathbf{b}$  is (nearly) contained in a  $k$ -dimensional invariant subspace, and hence the pair  $(\mathbf{A}, \mathbf{b})$  is (nearly) uncontrollable.

Which (if any) of these allows you to identify that the second  $\mathbf{b}$  pair is uncontrollable? Using that method, do you think that the original  $(\mathbf{A}, \mathbf{b})$  from the application is uncontrollable? Use the preferred method to test whether  $(\mathbf{A}, \mathbf{c})$  is *observable*, i.e., whether  $\mathbf{c}^*$  is contained in an invariant subspace of  $\mathbf{A}^*$  of dimension less than  $n$ .

**5. Bode diagram.** The transfer function for our system takes the form

$$\mathcal{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}.$$

If your system has no  $\mathbf{D}$  matrix, please set  $\mathbf{D} = \mathbf{0}$ .

- (a) Produce a `loglog` plot showing  $\|\mathcal{H}(i\omega)\|$  versus  $\omega$  for a suitable (problem dependent) range of  $\omega$  values, e.g., `omega = logspace(-3,3,100)` to get logarithmically sampled values  $\omega \in [10^{-3}, 10^3]$ . (Alternatively, you can use the `Mysigma_log` routine linked from the class website, or any tools from MATLAB's Control Toolbox.)
- (b) From your plot in part (a), estimate the  $H_\infty$  norm of your system, i.e., the largest value of  $\|\mathcal{H}(i\omega)\|$  for  $\omega \in \mathbb{R}$ .

**6. Greatest Total Energy [Optional].** Recall that the energy generated by the solution  $\mathbf{x}(t)$  of the system  $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$  with  $\mathbf{x}(0) = \mathbf{x}_0$  is given by

$$\mathcal{E}(\mathbf{x}_0) = \int_0^\infty \|\mathbf{x}(t)\|^2 dt = \mathbf{x}_0^* \mathbf{E} \mathbf{x}_0,$$

where  $\mathbf{E}$  solves the Lyapunov equation

$$\mathbf{A}^* \mathbf{E} + \mathbf{E} \mathbf{A} = -\mathbf{I}.$$

Solve this Lyapunov equation for  $\mathbf{E}$  and compute the greatest total energy, given by  $\|\mathbf{E}\|$ . (You can solve for  $\mathbf{E}$  using the `lyap` command in MATLAB's Control Toolbox, though we will discuss better approaches later in the semester.)

The choice of  $\mathbf{x}_0$  that maximizes the total energy  $\mathcal{E}(\mathbf{x}_0)$  over all unit vectors is simply the eigenvector of  $\mathbf{E}$  corresponding to the largest eigenvalue. Augment your plot from Question 1 with an illustration of  $\|\mathbf{x}(t)\|$  for this distinguished choice of  $\mathbf{x}_0$ .