# CMDA 3606 · MATHEMATICAL MODELING II

Problem Set 1

Posted 23 January 2019. Due at 5pm on Thursday, 31 January 2019.

***Please submit your problem set either in class, or directly to the instructor's office***

***Be sure to include your* MATLAB *code for all exercises.***

Basic guidelines: Students may discuss the problems on this assignment, but each student must submit his or her individual writeup and code. (In particular, you *must write up your own individual MATLAB code.*) Students may consult the class notes and other online resources, but *the use of solutions from previous classes is forbidden and will be regarded as a violation of the Honor Code.*

1. [8 points: 2 points per part] Compute (by hand) the matrix-vector product $\mathbf{Ax}$ for

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & -2 \\ 0 & -6 & 6 \\ -2 & 4 & 0 \end{bmatrix}.$$

and the column vectors

$$\mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}.$$

2. [12 points: 3 points per part] Consider the matrices

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 3 & -2 \\ 0 & 1 \end{bmatrix}.$$

   (a) Compute (by hand) $2\mathbf{A} - 3\mathbf{B}$.

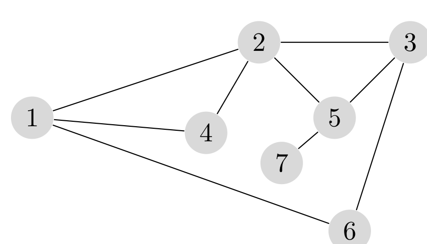
   (b) Compute (by hand) $\mathbf{AB}$, $\mathbf{BA}$, $(\mathbf{AB})^T$, $\mathbf{A}^T\mathbf{B}^T$, and $\mathbf{B}^T\mathbf{A}^T$.

   (c) From the calculations in part (b): does $\mathbf{AB} = \mathbf{BA}$?

   (d) From the calculations in part (b): does $(\mathbf{AB})^T = \mathbf{A}^T\mathbf{B}^T$ or does $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$?

3. [40 points: 10 points per part] This problem is designed to warm up your matrix creation and multiplication skills in MATLAB, and to introduce you to a fun application in network theory. The technique involves a slick way to count paths in graphs. (With this method you can analyze, for example, genetic networks or social networks.)

   A *graph* is a collection of *nodes* connected by *edges*. The graph below has 7 nodes (gray circles) connected by 9 edges (straight lines).



How "close" are two nodes in this network? One way to assess closeness is to count the number of *paths* that go from one node to another. For example, there is 1 path of length 2 from node 1 to node 5 (1–2–5) and 3 paths of length 3 (1–4–2–5, 1–2–3–5, and 1–6–3–5), but no paths of length 2 from node 1 to node 7.

As the paths get longer, there are more routing options between nodes and they get much harder to count. For example, there are already 14 paths of length 4 between nodes 2 and 6 (allowing for multiple visits to a node); it is hard to correctly count them all by hand. (Try it, if you like.) Matrices come to the rescue!

With each graph, we associate an *adjacency matrix* that encodes the edges. A graph with $n$ nodes gives an $n \times n$ adjacency matrix, $\mathbf{A}$. The entries in $\mathbf{A}$ are all zero, except that the $(j,k)$ entry of $\mathbf{A}$ is 1 if there is an edge from node $j$ to node $k$:

$$a_{j,k} = \begin{cases} 1, & \text{there is an edge connecting node } j \text{ to node } k; \\ 0, & \text{otherwise.} \end{cases}$$

For the graph above, the first row of the adjacency matrix is

$$[0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0]$$

since node 1 is connected to nodes 2, 4, and 6, but not to nodes 1, 3, 5, and 7.
(By convention, nodes are not connected to themselves, so the diagonal of $\mathbf{A}$ is all zero.)

**Neat fact:** (proved using the formula for matrix-matrix multiplication at the end of the assignment) The number of paths of length $p$ between node $j$ and node $k$ is given by the $(j,k)$ entry of the $p$th matrix power of the adjacency matrix: $\mathbf{A}^p$:

$$\mathbf{A}^p = \underbrace{\mathbf{A}\mathbf{A}\cdots\mathbf{A}}_{p \text{ times}}.$$

Use MATLAB to answer the following questions.

(a) Write down the $7 \times 7$ adjacency matrix $\mathbf{A}$ for the graph shown above and enter it in MATLAB.

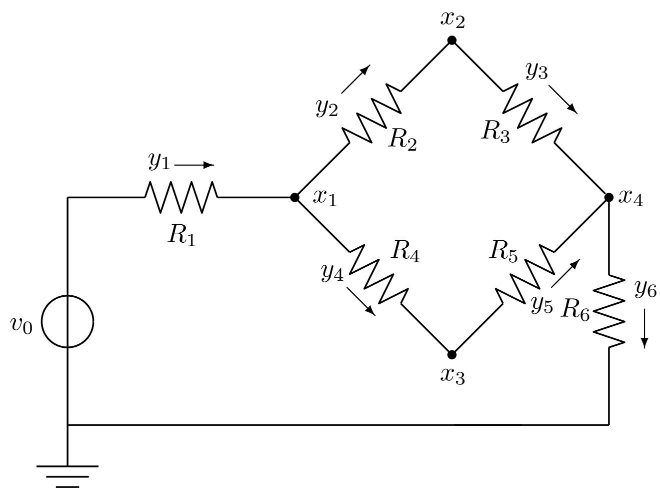
(b) Compute $\mathbf{A}^2$ and list the number of paths of length 2 between every pair of nodes.

(c) How many paths of length 10 are there between nodes 1 and 5 ?

You can associate the "importance" of node $j$ in the graph by looking at the magnitude of the $(j,j)$ entry of $\mathbf{A}^p$. Note that this reveals the number of paths of length $p$ from node $j$ returning back to itself. For example, the $(1,1)$ entry of $\mathbf{A}^2$ is 3, since there are 3 paths of length 2 that go from node 1 back to node 1. (These are 1–2–1, 1–4–1, and 1–6–1.)

(d) Look at the diagonal entries of $\mathbf{A}^p$ for various values of $p$. Based on this data, which node do you think is most important? (Show some data, but do not report all the $\mathbf{A}^p$ matrices that you compute. Be judicious.) Explain why the "importance" of a node might reasonably be evaluated in this way.

4. [40 points: 10 points per part]
Consider the following circuit with six resistors.




(a) Work through the first three steps of our circuit modeling procedure:
(1) compute the drops in potential, $\mathbf{e} = \mathbf{v} - \mathbf{A}\mathbf{x}$;
(2) compute the current through each resistor, $\mathbf{y} = \mathbf{K}\mathbf{e}$;
(3) apply Kirchhoff's current law, $\mathbf{0} = \mathbf{A}^T\mathbf{y}$.
At each step, write out the individual scalar equations (e.g., $e_1 = v_0 - x_1$), and then the matrix–vector form that collects those scalar equations together. In particular, specify the entries of $\mathbf{x}$, $\mathbf{y}$, $\mathbf{v}$, $\mathbf{e}$, and $\mathbf{A}$.

(b) Now assume $R_1 = R_2 = R_3 = R_4 = R_5 = R_6 = 1\,\Omega^\dagger$. Work out the entries of the matrix $\mathbf{S} = \mathbf{A}^T\mathbf{K}\mathbf{A}$.

(c) Under the conditions of part (b), use Gaussian elimination (by hand, not MATLAB) to solve the system $(\mathbf{A}^T\mathbf{K}\mathbf{A})\mathbf{x} = \mathbf{A}^T\mathbf{K}\mathbf{v}$ for the unknown potential vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}.$$

(Since the vector $\mathbf{v}$ contains the reference voltage $v_0$, your answer $\mathbf{x}$ should also contain the quantity $v_0$, as in equation (2.12) in the course notes. Here is a quick check on your answer: notice that the structure of the circuit and the fact that all resistances are equal ensures that $x_2 = x_3$. Does your answer agree?)

The circuit described here is a *Wheatstone bridge*, which finds use in instrumentation and measurement applications where one has a sensor that responds via a change in electrical resistance to a change in a particular feature in its environment; a variety of sensors measure key quantities in this way, such as salinity, temperature (e.g., via *thermisters*) or curvature (e.g., via *strain gauges*). A Wheatstone bridge allows one to convert small changes in resistance into easily measurable voltage differences.

(d) In the above circuit, suppose that $R_1 = R_2 = R_4 = R_5 = R_6 = 1\ \Omega$ as before but now the value of $R_3$ is variable and dependent on a sensor whose nominal state produces a resistance of $R_3 = 1\ \Omega$, but when perturbed it will produce resistance values $R_3 = s\ \Omega$ for $s \in [0.2, 5]$. We noted above

$^\dagger$One "Ohm" ($\Omega$) is the unit of resistance that would allow one Ampere ("Amp", A) of current to flow when there is a one Volt (V) drop in potential across the resistor.

that when $R_3 = 1\ \Omega$ then $x_2 = x_3$, and so the voltage difference $x_2 - x_3$ might be usable as a measure of how far the sensor has been perturbed from its nominal state $s = 1\ \Omega$: Assuming a reference voltage $v_0 = 1$ V, plot the voltage difference $x_2 - x_3$ as a function of $R_3 = s\Omega$ for $s \in [0.2, 5]$. Observe that $x_2 - x_3$ is a *monotone increasing* (and so, invertible) function of $s$ and $x_2 - x_3 = 0$ if and only if the sensor is in its nominal state, $s = 1\Omega$. Thus, a measured voltage drop $x_2 - x_3$ determines uniquely the state of the sensor, $s$. Remarkably, the voltage level at $x_3$ itself is independent of $s$ ! Check this. (For the resistance values given here, you should be finding voltage differences $x_2 - x_3$ roughly in the range of $\pm 100$ mV.)

You can fill out the following skeletal MATLAB commands for this problem, if you like.

```
svec = linspace(0.2, 5, 50);  % create vector (length 50) of values between 0.2 and 5
vdiff = zeros(size(svec));    % create a vector to store voltage differences at x2 and x3
for j=1:length(svec)
   % create the matrix AtKA = A'*K*A, which depends on svec(j)
   % solve for x
   vdiff(j) = x(2)-x(3);
end
figure
plot(svec, vdiff, '-','linewidth',2)
xlabel('label your axes! (include units)')
ylabel('label your axes! (include units)')
```

5. [Optional challenge problem: 6 bonus points: 2 points for (a); 4 points for (b)]

(a) Perform the modeling steps for the branched circuit with 16 resistors in Figure 2.4 of the class notes, a primitive model of a branched neuron. Implement the model in MATLAB (with $R_j = 1\ \Omega$ for all $j$ and $v_0 = 1$ V). Show the values you obtain for $x_1, \ldots, x_8$.

(b) Neurons have a more sophisticated branching structure than the model in part (a). (See `https://en.wikipedia.org/wiki/Neuron` for illustrations.)

Develop a MATLAB code that will implement a model like the example in Figure 2.4, but now with these more general features.

- Let the left "trunk" of the circuit (before the branch) contain $N$ "compartments" (assemblies of one horizontal and one vertical resistor). (Figure 2.4 has 2 compartments in this trunk; Figure 2.1 has three compartments.) This unit will define nodes $x_1, \ldots, x_N$, and models the neuron's long *axon*.
- At $x_N$, fork into $b > 1$ branches, modeling the neuron's *synapses*. (Figure 2.4 has $b = 2$ branches.)
- Let each branch have $M$ compartments. (Figure 2.4 has $M = 2$ compartments per branch.) Notice the additional vertical resistors at the start of each branch (resistors $R_6$ and $R_{12}$ in Figure 2.4.) These branches will define nodes

$$\begin{array}{ll} \text{branch 1:} & x_{N+1}, \ldots, x_{N+M+1} \\ \vdots & \vdots \\ \text{branch } b\text{:} & x_{N+b(M+1)-M}, \ldots, x_{N+b(M+1)}. \end{array}$$

Test your model with $N = 32$, $b = 4$, $M = 16$ (with $v_0 = 1$ V and $R_j = 1\ \Omega$).

- Display the nonzero pattern of $\mathbf{S} = \mathbf{A}^T\mathbf{K}\mathbf{A}$ by producing a "spy plot" using the `spy(S)` command in MATLAB.
- Produce plots showing $x_1, \ldots x_N$ (for the axon) and $x_{N+1}, \ldots x_{N+M+1}$ (for the first branch).

**Appendix.** Proof that $(\mathbf{A}^p)_{j,k}$ counts the paths of length $p$ between nodes $j$ and $k$ in a graph. (See Problem 3).

Let $\mathbf{A}$ denote the adjacency matrix for a graph with $n$ nodes, so that

$$a_{j,k} = \begin{cases} 1, & \text{there is an edge connecting node } j \text{ to node } k; \\ 0, & \text{otherwise.} \end{cases}$$

The proof will use induction. Notice that $a_{j,k}$ counts the number of paths of length 1 between nodes $j$ and $k$. Thus the base case ($p = 1$) for the induction holds.

Now make the inductive assumption that $(\mathbf{A}^p)_{j,k}$ counts the paths of length $p$ between nodes $j$ and $k$. We will prove that $(\mathbf{A}^{p+1})$ counts the paths of length $p+1$ between nodes $j$ and $k$.

Any path of length $p+1$ from node $j$ to node $k$ can be broken into two subpaths:

$$\text{path of length } p+1 \text{ from } j \text{ to } k = (\text{any path of length 1 from } j \text{ to } \ell) \cup (\text{any path of length } p \text{ from } \ell \text{ to } k)$$

for some node $\ell \in \{1, \ldots, n\}$. To count all paths of length $p+1$, sum up over all the possible intermediate nodes $\ell$, and using the facts that:

- $(\mathbf{A})_{j,\ell}$ counts the paths of length 1 from $j$ to $\ell$ (base case);
- $(\mathbf{A}^p)_{\ell,k}$ counts the paths of length $p$ from $\ell$ to $k$ (inductive assumption).

Thus, we can calculate

$$\begin{aligned} &\#\text{ of paths of length } p+1 \text{ from } j \text{ to } k \\ &\qquad = \sum_{\ell=1}^{n} (\#\text{ of paths of length 1 from } j \text{ to } \ell) \times (\#\text{ of paths of length } p \text{ from } \ell \text{ to } k) \\ &\qquad = \sum_{\ell=1}^{n} (\mathbf{A})_{j,\ell} (\mathbf{A}^p)_{\ell,k}, \end{aligned}$$

but this last formula is simply the matrix-matrix multiplication formula for the $(j,k)$ entry of $\mathbf{A}\mathbf{A}^p = \mathbf{A}^{p+1}$: the number of paths of length $p+1$ from node $j$ to node $k$ is $(\mathbf{A}^{p+1})_{j,k}$.

Thus, by induction, we have proved that the number of paths of length $p$ between nodes $j$ and $k$ is $(\mathbf{A}^p)_{j,k}$.