Extending Justin's Guide to MATLAB in MATH 240 - Part 4

New Commands

1. Eigenvalues can be found easily. If `A` is a matrix then:

   `>> eig(A)`

   will return the eigenvalues. Note that it will return complex eigenvalues too. So keep an i open for those.

2. If we have an eigenvalue λ for A, we can use `rref` on an augmented matrix $[A - \lambda I \mid \mathbf{0}]$ to lead us to the eigenvectors. For example if A is 4×4 and $\lambda = 3$ is an eigenvalue, then we can obtain the coefficient matrix of this system by entering `A - 3*eye(4)`.

3. Even better: MATLAB can do everything in one go. If you recall from class, *diagonalizing* a matrix A means finding a diagonal matrix D and an invertible matrix P with $A = PDP^{-1}$. The diagonal matrix D contains the eigenvalues along the diagonal and the matrix P contains eigenvectors as columns, with column j of P corresponding to the eigenvalue in column j of D.

   To do this we use the `eig` command again but demand different output. The format is:

   `>> [P,D]=eig(A)`

   which assigns `P` and `D` for A, if possible. If it's not possible MATLAB returns very strange-looking output.

4. We can compute the dot product of two vectors using the command `dot`. For example:

   `>> dot([1;2;4],[-2;1;5])`

5. We can find the length of a vector from the basic definition. If `v` is a vector then:

   `>> sqrt(dot(v,v))`

6. Or we can just use the `norm` command:

   `>> norm(v)`

7. To get the transpose of a matrix `A` we do:

   `>> transpose(A)`

   or

   `>> A'`

   Note that `A'` actually returns that *conjugate transpose* (or *adjoint*) of `A`. That is, complex conjugation is also applied to all of the entries of A^T. This doesn't make a difference if the entries of the matrix are real numbers, but it will make a difference if they are (nonreal) complex numbers

8. To find the rank of a matrix `A` we do

   `>> rank(A)`

9. When A is a matrix with linearly independent columns, the command

   `>> [Q,R]=qr(A,0)`

   will create and exhibit the matrices Q, R which give the QR factorization of A as defined in the text of Lay.

MATH 240 Fall 2019 MATLAB Project 4 – due Thursday, December 3, 11:59pm

---

Directions:

Previous guidelines on format and collaboration hold. Please review them if you forget them. For this project, do the first problem in `format rat` and do the rest in in `format short`.

As before, a question part marked with a star $\star$ indicates the answer should be typed into your output as a comment – the question isn't asking for MATLAB output.

---

1. (Use `format rat`) Recall $[\mathbf{x}]_{\mathcal{B}}$ denotes the coordinate vector of $\mathbf{x}$ with respect to a basis $\mathcal{B}$ for a vector space V. Given two bases $\mathcal{B}$ and $\mathcal{C}$ for V, $\underset{\mathcal{C}\leftarrow\mathcal{B}}{P}$ denotes the change of coordinates matrix, which has the property that

$$\underset{\mathcal{C}\leftarrow\mathcal{B}}{P}[\mathbf{x}]_{\mathcal{B}} = [\mathbf{x}]_{\mathcal{C}} \quad \text{for all } \mathbf{x} \in V.$$

It follows that

$$\underset{\mathcal{B}\leftarrow\mathcal{C}}{P} = \left(\underset{\mathcal{C}\leftarrow\mathcal{B}}{P}\right)^{-1}.$$

Also, if we have three bases $\mathcal{B}, \mathcal{C}$, and $\mathcal{D}$, then

$$\left(\underset{\mathcal{D}\leftarrow\mathcal{C}}{P}\right)\left(\underset{\mathcal{C}\leftarrow\mathcal{B}}{P}\right) = \underset{\mathcal{D}\leftarrow\mathcal{B}}{P}.$$

Each of the following three sets is a basis for the vector space $\mathbb{P}_3$:

$$\begin{aligned}
\mathcal{E} &= \{1,\ t,\ t^2,\ t^3\}\ , \\
\mathcal{B} &= \{1,\ 1+2t,\ 2-t+3t^2,\ 4-t+t^3\}\ , \qquad \text{and} \\
\mathcal{C} &= \{1+3t+t^3,\ 2+t,\ 3t-t^2+4t^3,\ 3t\}\ .
\end{aligned}$$

(a) Find and enter the matrices $P = \underset{\mathcal{E}\leftarrow\mathcal{B}}{P}$ and $Q = \underset{\mathcal{E}\leftarrow\mathcal{C}}{P}$.

(b) Use P and Q and the properties above to compute $R = \underset{\mathcal{C}\leftarrow\mathcal{B}}{P}$.

(c) Compute the $\mathcal{C}$ coordinate vector of the polynomial t^3.

(d) Suppose $p(t)$ is the polynomial for which $[p(t)]_{\mathcal{B}} = \begin{bmatrix} 0 \\ 3 \\ 2 \\ 1 \end{bmatrix}$. Compute the coordinate vector $[p(t)]_{\mathcal{C}}$.

(e) Let $p(t)$ denote the polynomial from the previous part. Express this polynomial in the form $p(t) = a_0 + a_1 t + a_2 t^2 + a_2 t^3$.

2. (Use `format short` for here onward) Let $A = \begin{bmatrix} 163 & 34 & -8 \\ -522 & -108 & 26 \\ 990 & 210 & -47 \end{bmatrix}$.

(a) Execute the command `[P,D] = eig(A)` to diagonalize A.

(b) Use MATLAB to verify that $A = PDP^{-1}$.

(c) $\star$ Use the previous results to give the eigenvalues of A, and give an eigenvector for each eigenvalue.

3. Let $A = \begin{bmatrix} -23 & -32 & -10 \\ 11 & 15 & 5 \\ 18 & 26 & 7 \end{bmatrix}$.

(a) Use MATLAB to compute A^n for $n = 2, 3, 4, 5, 6, 7, 8$. Do you notice a pattern?

(b) Have MATLAB produce an invertible P and a diagonal D such that $A = PDP^{-1}$. Notice that complex numbers get involved.

(c) $\star$ To understand A^n, it suffices to understand D^n because $A^n = PD^nP^{-1}$. Describe the pattern that emerges when we consider powers of D: D, D^2, D^3, D^4,etc.

(d) $\star$ Without doing a computation in MATLAB, determine $A^{20000001}$.

4. Let $A = \begin{bmatrix} 3 & 1 \\ 0 & 3 \end{bmatrix}$.

   (a) Execute the command `[P,D]=eig(A)`. Something strange should occur in the output (take a close look at P).

   (b) Use MATLAB to try see if $A = PDP^{-1}$.

   (c) Find a basis for the eigenspace of A corresponding to the eigenvalue $\lambda = 3$.

   (d) $\star$ Is there a basis for $\mathbb{R}^2$ consisting of eigenvectors for A? Does this explain why something went wrong in part (b)? (There is a relevant theorem in §5.3.)

5. Let W be the subspace of $\mathbb{R}^5$ given by

$$W = \operatorname{Span}\left\{ \begin{bmatrix} 9 \\ 14 \\ -11 \\ 3 \\ 0 \end{bmatrix}, \begin{bmatrix} -14 \\ -4 \\ -10 \\ 9 \\ -5 \end{bmatrix}, \begin{bmatrix} 1 \\ -10 \\ 4 \\ -7 \\ 5 \end{bmatrix}, \begin{bmatrix} 6 \\ 8 \\ -1 \\ -12 \\ -8 \end{bmatrix} \right\}$$

   (a) Enter the four vectors into MATLAB as `v1`, `v2`, `v3` and `v4` respectively.

   (b) Let `A = [v1 v2 v3 v4]`. Compute the rank of this matrix. Explain briefly in a comment why this shows that this set of four vectors is a basis for W.

   (c) We shall apply the Gram-Schmidt Process to produce an orthogonal basis $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4\}$ for W. To begin, let `w1 = v1` and `w2 = v2 - (dot(w1,v2)/dot(w1,w1))*w1`.

   (d) Continue the Gram-Schmidt Process and compute `w3` and `w4`.

   (e) Rescale each vector of the orthogonal basis $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4\}$ to produce an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4\}$ for W. (Recall there is a command in MATLAB to compute the norm of a vector.)

   (f) Enter the vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$ into the columns of a matrix Q. Verify that the columns of Q are orthonormal with a single matrix multiplication.

   (g) Compute $R = Q^TA$. Verify that R is upper triangular with positive diagonal entries and that $A = QR$.

   (h) MATLAB can compute a QR factorization in a single command. Enter `[Q1, R1] = qr(A,0)`. Notice that there is a small discrepancy between your `Q`, `R` and the `Q1`, `R1` produced by MATLAB. This is because QR factorizations are not unique. They are only unique if we insist that the diagonal entries of R are all postive. Nonetheless, the columns of `Q1` still form an orthonormal basis for W.

6. Let W be the subspace of $\mathbb{R}^6$ given by

$$W = \operatorname{Span}\left\{ \begin{bmatrix} 5 \\ -5 \\ 0 \\ -4 \\ -3 \\ 4 \end{bmatrix}, \begin{bmatrix} 9 \\ -12 \\ 2 \\ -3 \\ -7 \\ 2 \end{bmatrix}, \begin{bmatrix} -2 \\ -4 \\ 4 \\ 10 \\ -2 \\ -12 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \\ -2 \\ -4 \\ -2 \\ 7 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \\ 2 \\ 5 \\ -1 \\ -6 \end{bmatrix} \right\}$$

   (a) Enter those five vectors as the columns of a matrix A and compute its rank.

   (b) The previous computation shows that the five vectors are linearly dependent, hence they do not form a basis for W. Find a basis for W. (Hint: Treat W as $\operatorname{Col} A$)

(c) Enter your basis for W as the columns of a matrix B. Compute the factorization $B = QR$ with a single command. Give an orthonormal basis for W.

(d) Let $E = QQ^T$. As a linear transformation on $\mathbb{R}^6$, E is the orthogonal projection onto the subspace W. Use E to compute the orthogonal projection of the vector $\mathbf{v} = (1, 1, 1, 1, 1, 1)^T$ onto W.

(e) Find a basis for $W^\perp$ as follows. Note that $W = \operatorname{Col} A = \operatorname{Col} B$. From a theorem in class, we have

$$W^\perp = (\operatorname{Col} B)^\perp = \operatorname{Nul}(B^T),$$

so it suffices to find a basis for $\operatorname{Nul}(B^T)$.

(f) As you did for W, find an orthonormal basis for $W^\perp$.

(g) Compute the matrix F for the orthogonal projection onto $W^\perp$. What is the sum $E + F$?