APM120, Homework #0
Applied Linear Algebra and Big Data
Last updated: Thursday 19th January, 2023, 20:10
Assigned Jan 24, 2023, due Jan 31, 1:00 pm, via gradescope, **in pdf $\leq$ 20Mb, $\leq$ 30 pages.**

This homework reviews the basic linear algebra needed for this course, please see the corresponding review examples in appendix A of the course notes. **This HW assignment is long... the following ones will be shorter.**

Show all steps in all calculations explicitly. Attach code used, well documented, and relevant plots and Matlab/python output, attaching code and figures *immediately following* the relevant question solution. A code printout is not a substitute for complete solutions, your solution should stand alone without the Matlab/python code or output. See needed python preliminaries at end of this HW[1]. You may use Matlab/python only for questions indicated in orange (in future HW the default would be that Matlab/python are allowed unless it is indicated explicitly that you need to use hand-calculations).

1. **Scalars, vectors and matrix operations:** In this and the following questions, denote

$$\mathsf{A} = \begin{pmatrix} 3 & 4 & 4 \\ -0.6 & -4.8 & -7.8 \\ 1.5 & 0.4 & -10.8 \end{pmatrix}, \ \mathsf{B} = \begin{pmatrix} 11 & -9 & 4 \\ 10 & -8 & 4 \\ -14 & 12 & -4 \end{pmatrix}, \ \mathsf{N} = \begin{pmatrix} 1 & 3 & 2 & 7 \\ 3 & 1 & 4 & 2 \\ -2 & 2 & -2 & 5 \end{pmatrix},$$

$$\mathsf{P} = \begin{pmatrix} 4 & 2 \\ 1 & 5 \end{pmatrix}, \ \mathbf{a} = \begin{pmatrix} 1-3i \\ 1 \\ -2 \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} -2 \\ 4 \\ 1 \end{pmatrix}, \ \mathbf{c} = \begin{pmatrix} -2 \\ 4 \\ 1 \end{pmatrix}, \quad \begin{aligned} z_1 &= 2+5i, \\ z_2 &= 5-2i. \end{aligned}$$

Calculate and write **(i)** the matrix product $\mathsf{B}^T\mathsf{B}$, **(ii)** product of a raw vector and a matrix $\mathbf{b}^T\mathsf{B}$, **(iii)** scalar products of two vectors $\mathbf{a}^T\mathbf{c}$ and $\mathbf{a}^\dagger\mathbf{a}$, **(iv)** the vector norm $|\mathbf{c}|$. Here, $(\cdot)^T$ and $(\cdot)^\dagger$ represent the transpose and conjugate transpose, correspondingly.

2. **Linear equations, Gaussian elimination & back substitution:** solve $\mathsf{A}\mathbf{x} = \mathbf{b}$ using Gaussian elimination to bring the equation to an upper triangular form and then using back substitution to find all elements of $\mathbf{x}$.

3. **Determinants, linear independence of vectors:** **(i)** Calculate $\det(\mathsf{A})$ by reducing to row-echelon form and multiplying the diagonal terms. **(ii)** Show that $\det(\mathsf{B}) = 0$ using Matlab: `det(B)`, or python: `np.linalg.det(B)`. **(iii)** Conclude that the columns of B are linearly dependent; write the 1st column as a linear combination of the other two. **(iv)** *****Optional extra credit:** Calculate $\det(\mathsf{A})$ using the cofactors of A.

4. **Matrix inversion, invertible vs singular matrices:** **(i)** calculate A^{-1} by performing row operations in parallel on A and on the identity matrix. **(ii)** Using Matlab/python, show explicitly that $\mathsf{A}^{-1}\mathsf{A} = \mathsf{I}$. Does B^{-1} exist? Why?

5. **Complex numbers:** calculate the following quantities,

(a) $z_1^\dagger z_2$,	(b) $\lvert z_1/z_2\rvert$,
(c) $3(e^{2z_1})^\dagger$,	(d) Write $z_1 = re^{i\theta}$ and find r, θ.

6. **Homogeneous linear equations:** **(i)** What conditions does a general matrix G need to satisfy for there to be a nonzero vector $\mathbf{x}$ for which $\mathsf{G}\mathbf{x} = \mathbf{0}$? **(ii)** *****Optional extra credit:** Find the general solution $\mathbf{x}$ to $\mathsf{B}\mathbf{x} = \mathbf{0}$ for the matrix defined above.

7. **Eigenvalues and eigenvectors:** **(i)** Write the characteristic equation of P, $\det(\mathsf{P} - \lambda\mathsf{I}) = 0$, and solve it for the eigenvalues λ_i. Calculate the eigenvectors $\mathbf{e}_i$ (by hand) by solving $\mathsf{P}\mathbf{e}_i = \lambda_i\mathbf{e}_i$ and requiring that their norm is equal to one, $|\mathbf{e}_i|^2 = 1$. **(ii)** Find the eigenvalues and eigenvectors of B in Matlab using `[V,D]=eig(B)` or in python using `D,V=np.linalg.eig(B)`. **(iii)** Verify that the eigenvector $\mathbf{e}$ corresponding to the *largest absolute value* eigenvalue λ indeed satisfies $\mathsf{B}\mathbf{e} = \lambda\mathbf{e}$. If the ith eigenvalue returned by Matlab ($1 \le i \le 3$) or python ($0 \le i \le 2$), is the largest in absolute value, then the corresponding eigenvector is the ith column of V, `V(:,i)` in Matlab, `V[:,i]` in python, and the eigenvalue is ith element of `diag(D)` in Matlab, or of `D` in python. **(iv)** Solve for the eigenvector of B corresponding to the eigenvalue of largest absolute value.

8. **Matrix diagonalization:** Let U be a matrix whose columns are the above-calculated eigenvectors of P and consider $\mathsf{H} = \mathsf{U}^{-1}\mathsf{P}\mathsf{U}$. **(i)** Calculate H_{11} and H_{12}. **(ii)** Use Matlab/python to calculate all elements of H and show it is diagonal and that the values along the diagonal are the eigenvalues of P which you also calculated above.

9. **Gram-Schmidt orthogonalization:** starting from the three columns of A taken as vectors $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, apply the Gram-Schmidt orthogonalization process to find three orthonormal vectors $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$ satisfying $\mathbf{y}_i^T\mathbf{y}_j = \delta_{ij}$ where δ_{ij} is the Kronecker delta.

10. **Null space of a matrix:** calculate the null space of the matrix N.

11. **Gradient:** Given a column vector $\mathbf{x} = (x, y)^T$, **(i)** write the explicit expression for $J(x, y) = \mathbf{x}^T(\mathsf{P}^T\mathsf{P})\mathbf{x}$ in terms of (x, y), evaluating numerically the coefficients that depend on P. **(ii)** Write the gradient of this function, $\nabla J(x, y)$ as a function of (x, y). Evaluate the gradient at the point $\mathbf{x}_0 = (x, y) = (-2, 1)$. **(iii)** What do the direction and magnitude of the gradient represent in terms of the geometric properties of $J(x, y)$ as a function of two variables? **(iv)** Calculate the two numbers $J(\mathbf{x}) \pm \delta\mathbf{x} \cdot \nabla J$, where $\delta\mathbf{x}$ is a small vector in the direction of ∇J, $\delta\mathbf{x} \equiv 0.001\nabla J$, evaluated at $\mathbf{x} = \mathbf{x}_0$; explain the results. **(v)** *****Optional extra credit:** Use Matlab/python to plot contours of J and a vector pointing in the direction of $\nabla J(x, y)$, originating at $\mathbf{x}_0$.

12. **Running a Matlab/python program:** Modify the sample problem code `HW_00.m`/`HW_00.py` from the HW-00 homework folder to **(i)** plot $2e^{-x/2}\sin(2x)$ vs x for $(2\pi \le x \le 3\pi)$, **(ii)** solve the linear equations $\mathsf{A}\mathbf{x} = \mathbf{b}$ with the above matrix A by calculating its inverse using Matlab (`inv(A)`) or python (`np.linalg.inv(A)`) and compare to the solution you found above using Gaussian elimination.

* [What's the point of *****optional extra credit** challenge problems: apart from the fun of doing them, they may bring the total score of this HW assignment up to 110%, making up for problems you may have missed in this or other HW assignments. . .]

Python preliminaries & notes

1 python commands within the HW assume you have first used the followings: `import numpy as np;` `from numpy import linalg; import scipy as scipy; from scipy import linalg;` `import matplotlib.pyplot as plt; import matplotlib;`
Input a matrix A, column vector **b** and row vector **c** into python in the form `A=np.array([[a11,a12,a13],[a21,a22,a23],[a31,a32,a33]]; b=np.array([[b1],[b2],[b3]]);` `c=np.array([[c1,c2,c3]]);` or convert Matlab arrays given in HW directly to python arrays using, e.g., `A=np.array(np.matrixlib.defmatrix.matrix('[1 2 3; 4 5 6]'));`