Take-Home Final Exam

Due Thursday May 7, at 5pm,
to me or in my Chapman 101 (DMS office) box.

115 points total

Rules. You may not talk or communicate about this exam with any person other than me (Ed Bueler). You may use any reference, print or electronic, as long as it is clearly cited, but you may not search out complete solutions to these particular problems. References to the textbook may be omitted, in general, though you should add them as needed for clarity.

F1. Consider a full rank matrix $A \in \mathbb{C}^{m\times n}$, where $m < n$.[1] Because the system $Ax = b$, for $b \in \mathbb{C}^m$, is underdetermined and has many solutions one must give further properties to determine a single solution. In this case, we decide that the desired solution is $\tilde{x} \in \mathbb{C}^n$ where $A\tilde{x} = b$ and $\|\tilde{x}\|_2 = \min\left\{\|x\|_2 \big| Ax = b\right\}$.

(a) (*5 points*) Carefully and mathematically explain why the system $Ax = b$ is "underdetermined" and why it has many solutions. Compare the definition of the desired solution $\tilde{x}$ to the corresponding definition for overdetermined systems (Lecture 11). (Imagine, if needed, that you are writing a new lecture in the textbook on underdetermined systems ...)

(b) (*5 points*) Recalling Figure 11.3 of the text, draw the analogous picture for an underdetermined system. Include null(A) and range(A^*) in your picture. (*Hint*: Your picture should be a picture of $\mathbb{C}^n$, the input space of A, rather than the output space $\mathbb{C}^m$ as in Figure 11.3.)

(c) (*5 points*) Use the SVD of A to describe an algorithm to find $\tilde{x}$.

F2. **(a)** (*5 points*) Fix $m = 1000$. Generate a random $m \times m$ matrix `A=randn(m,m)` and a random column vector `b=rand(m,1)`. How accurately can the system $Ax = b$ be solved? (*Hint: I want a number, and its meaning. See lectures 12 and 15.*) Repeat this experiment a few times, and comment.

(b) (*5 points*) Now ignor accuracy and focus on speed of solving systems of this size. Set up an experiment comparing M|O's backslash (which is Gauss elimination, namely LU factorization with partial pivoting)[2] versus the method on page 54 of the textbook, using M|O's built-in `qr`, for solving $Ax = b$. (*Note: In the latter algorithm you* may *use* M|O*'s backslash to solve a triangular system.*) Use `tic` and `toc` to time the two algorithms, skipping any actual display of the answer x or any information other than for timing. Make sure to repeat experiments enough to get some convincing sense of execution speed. As demonstrated by these execution times, by what factor is the second algorithm slower than the first here?

(c) (*5 points*) Now read the appropriate lectures in the textbook, including Lecture 20, to give the theoretical answer to "By what factor is the second algorithm slower than the first?" in part (b). Note that for the LU algorithm described in Lecture 20, adding partial pivoting adds only lower order terms to the work estimate. Comment on why practice might differ from theory.

---

[1]The result of this problem should be a clear mental picture of underdetermined systems, and a bit of their numerical analysis.

[2]For OCTAVE I track it to this fortan code: `http://www.netlib.org/lapack/double/dgetrf.f`, but you don't need to know that to answer this question ...

F3. Go to this Cleve Moler page and read the whole thing (interesting, I hope, and not terribly long):

`http://www.mathworks.com/company/newsletters/news_notes/clevescorner/oct02_cleve.html`

Let's compute PageRank on a "small world" web of ten pages. I will use the notation of this Cleve Moler page, so I'll write $n = 10$ and start by generating randomly the connectivity matrix G. Let me assume that G does not have ones on the diagonal, because they would represent links from a page back to itself. Here goes:

```
>> n = 10;
>> G = (rand(n,n)<.3); G = G & (eye(n,n)==0);
>> G
G =
     0     0     0     1     0     0     1     1     1     1
     1     0     0     0     0     0     1     1     0     0
     1     0     0     1     0     1     0     1     1     0
     0     0     0     0     0     0     0     1     0     0
     0     0     1     0     0     0     1     0     1     1
     0     1     0     0     0     0     0     0     0     1
     1     0     0     0     0     0     0     0     0     0
     0     0     0     1     1     0     0     0     0     0
     0     0     0     1     1     0     0     0     0     0
     1     0     1     1     0     0     1     0     0     0
```

(a) (*5 points*) Show what web I have generated by drawing a diagram of the ten web pages, as nodes labeled $W_1, \dots, W_{10}$, and their links as arrows. A "directed graph". What is the role of ".3" in the above? What is "`G = G & (eye(n,n)==0)`" doing to modify G?

(b) (*5 points*) Now enter the particular G displayed above, and fix[3] it as the connectivity matrix for the rest of the problem (except part (e)). Confirm that the terms "indegree" and "outdegree" make sense to you, and compute those (and show them as row vectors, for compactness). Now choose $p = 0.85$ and compute A as described at Moler's page. Note that the small constant $\delta = (1 - p)/n$ contributes to every entry of A. (*I.e., the formula for A has "$+\delta$" not "$+\delta_{ij}$".*) Actually display A and show that the sums of the columns are one.

(c) (*5 points*) Matrix A gives transition probabilities for a "Markov chain". This means that if $x_k \in \mathbb{R}^n$ is a vector with positive entries which sum to one, so that the entries $(x_k)_j$ can be thought of as probabilities of the random walk being on web page j at step k of the random walk, then the next state of the random walk is given by a new vector of probabilities,

$$x_{k+1} = Ax_k.$$

Show that if you find probabilities which don't change as the random walk goes on, then you have an eigenvector of A with all entries summing to one and with eigenvalue equal to one. Show, conversely. Also, correctly and formally state the Perron-Frobenius theorem (from any source).

(d) (*5 points*) Use the `eig` command on A to compute PageRank for the small-world-wide-web described by G. (*Hint.* Be careful to check that your probabilities sum to one.)

(e) (*2 points*) Use Google to find out what value n has for the actual world wide web in April 2009. Does Google use the algorithm you used in (d) to compute PageRank?

[3]This way I can grade your answer based on the work I've done! You can find G at `http://www.dms.uaf.edu/~bueler/Gpagerank.m`.

F4. **(a)** (*5 points*) Consider the following

Definition. A square matrix $A \in \mathbb{C}^{m\times m}$ is *normal* if $A^*A = AA^*$.

Note the obvious fact that if A is hermitian ($A^* = A$) then A is normal. Show that if A is unitary, then A is normal. Use the definition directly on

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

describe all normal 2×2 matrices with real entries. Also, state one concrete 2×2 matrix with real entries which is not normal.

The definition of "normal" is not very illuminating if we think directly about matrix entries. The rest of the problem shows that if we think in terms of eigenvalue and Schur decompositions of A, then the definition of "normal" is quite natural.

(b) (*5 points*) Show that if there is a orthonormal basis for $\mathbb{C}^m$ formed from eigenvectors of A then A is normal.

(*Hint*: Can you write a matrix decomposition for A?)

(*Informally, one might say*: "If the eigenvectors of A form an orthonormal basis then A is normal." But we don't actually mean all of the eigenvectors, right?)

(c) (*5 points*) Show that an upper triangular matrix $T \in \mathbb{C}^{m\times m}$ is normal if and only if it is diagonal.

(d) (*5 points*) Show that if A is normal then there is a basis for $\mathbb{C}^m$ formed from eigenvectors of A.

(*Hint*: Every square matrix A has a Schur decomposition. Start there and use **(c)**.)

(*Where we stand:* Parts **(b)** and **(c)** are called the *spectral theorem* for matrices. These ideas generalize to infinite dimensions, a fact essential to understanding quantum mechanics, among other fields of analysis.)

(e) (*3 points*) How common are normal matrices? Just thinking of 8×8 matrices, write a program which generates 1000 matrices from `randn(8,8)` and test them for normality. What fraction are normal? Explain, in terms of the Schur decomposition. (Weak pun: *Are normally-distributed matrices normal?*)

F5. (*10 points*) **Exercise 24.3 in** Tref & Bau.

F6. (*10 points*) **Exercise 24.4 in** Tref & Bau. (In each part, make sure to prove both the "if" and "only if" directions of "$\iff$"!)

F7. (*10 points*) **Exercise 2.5 in** Tref & Bau.

F8. (*10 points*) Choose **one**:

- **Exercise 4.2 in** Tref & Bau.
- **Exercise 4.3 in** Tref & Bau.