18.335 Midterm

You have two hours. **All problems have equal weight**, but some problems may be harder than others, so try not to spend too much time on one problem.

Problem 1: Schur, backsubstitution, complexity (20 points)

You are given matrices A ($m \times m$), B ($n \times n$), and C ($m \times n$), and want to solve for an unknown *matrix* X ($m \times n$) solving:

$$AX - XB = C.$$

We will do this using the Schur decompositions of A and B. (Recall that any square matrix S can be factorized in the Schur form $S = QUQ^*$ for some unitary matrix Q and some upper-triangular matrix U.)

(a) *Given* the Schur decompositions $A = Q_A U_A Q_A^*$ and $B = Q_B U_B Q_B^*$, show how to transform $AX - XB = C$ into new equations $A'X' - X'B' = C'$, where A' and B' are upper triangular and X' is the new $m \times n$ matrix of unknowns. That is, express, A', B', and C' in terms of A, B, and C (or their Schur factors), and show how to get X back from X'.

(b) Given the upper-triangular system $A'X' - X'B' = C'$ from (a), give an algorithm to find the *last row* of X'. (Hint: look at the last row of the equation.)

(c) Suppose you have computed all rows $> j$ of X', give an algorithm to compute the j-th row.

(d) The combination of the previous three parts yields a backsubstitution-like algorithm to solve for X. What is the complexity of this algorithm (*not including* the cost of computing the Schur factors)? That is, give the leading terms in m, n for the operation count; don't worry about constant factors [for example, $O(m^7)$ would be fine; you don't need to specify that it is $\sim \frac{15}{9}m^7$].

Problem 2: Stability (20 points)

If a method for solving $Ax = b$ is backwards stable, what does that imply about the size of the computed (floating-point) residual $\tilde{r} = \widetilde{b - A\tilde{x}}$ from the computed $\tilde{x}$? i.e. relate $\|\tilde{r}\|$ to $O(\varepsilon_{\text{machine}})$ and any norms or condition numbers of A and/or b that you might need.

Problem 3: Conjugate gradient (20 points)

The conjugate-gradient (CG) algorithm to solve a positive-definite Hermitian system of equations $Ax = b$ is given below.

```
$x_0 = 0$, $r_0 = b$, $d_0 = r_0$
for $n = 1, 2, \ldots$
    $\alpha_n = \frac{r_{n-1}^* r_{n-1}}{d_{n-1}^* A d_{n-1}}$
    $x_n = x_{n-1} + \alpha_n d_{n-1}$
    $r_n = r_{n-1} - \alpha_n A d_{n-1}$
    $d_n = r_n + d_{n-1} \frac{r_n^* r_n}{r_{n-1}^* r_{n-1}}$
```

Now, suppose instead of A being positive-definite, it is only positive-semidefinite—the eigenvalues λ are nonnegative, but some are zero. A (non-unique) solution to $Ax = b$ still exists if b is in the column space of A, that is if b is in the span of the positive-λ eigenvectors. The following questions concern the application CG to this situation.

(a) Let the eigenvalues and (orthonormal) eigenvectors of A be denoted by $Aq_j = \lambda_j q_j$ ($j = 1, \ldots, m$), and suppose that only the first k eigenvalues $\lambda_{1,\ldots,k}$ are zero. If we expand x_n in the basis of these eigenvectors q_j at each step, what does each step of CG do to the coefficients of these first k eigenvectors? i.e. do these zero-λ eigenvector components, grow, shrink, or stay the same in magnitude when we go from x_{n-1} to x_n?

(b) From the previous part, conclude whether the CG algorithm above will converge. If not, explain what the residual $\|r_n\|$ does (how fast it grows, if it grows). If it converges, explain what it converges to and what determines the rate of convergence (it can't be $\kappa(A)$ since that is infinite!). (Hint: relate CG to CG on the linear operator A restricted to the subspace of nonzero-λ eigenvectors).

(c) Does it matter if we choose the initial guess $x_0 = 0$? i.e. what happens if we choose some random initial guess x_0 (and correspondingly set $r_0 = b - Ax_0$)?

(d) Suggest an iterative algorithm to find a vector in the null space of A. (Hint: think about $b = 0$.)

Problem 4: Rayleigh quotients (20 points)

Recall that, for a Hermitian matrix A, the Rayleigh quotient is the function $r(x) = x^*Ax/x^*x$. Furthermore, we showed that, for any x, $r(x)$ is between the minimum and maximum eigenvalues of A.

Let $A = \begin{pmatrix} B & b \\ b^* & \gamma \end{pmatrix}$ be a Hermitian matrix (for some Hermitian matrix B, some vector b, and some real number γ). Show that the smallest eigenvalue of B is $\geq$ smallest eigenvalue of A.

Problem 5: Norms and SVDs (20 points)

Let $A = \begin{pmatrix} I & B \\ B^* & I \end{pmatrix}$ with $\|B\|_2 < 1$. Show that $\kappa(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{1+\|B\|_2}{1-\|B\|_2}$.

Hint: Given the SVD $B = U\Sigma V^*$ of B, construct the SVD of A. Recall that the induced norm $\|\cdot\|_2$ of a matrix is simply the largest singular value, and hence the condition number is the ratio of the largest to smallest singular values. (It might help you to think about the case where B is just a number, to get you started—what are the eigenvectors and eigenvalues of A in that case? Then for the general case, make eigenvectors of A out of U and V.)

Problem 6: Least-squares problems (20 points)

If W is positive-definite $n \times n$, and we define the norm $\|x\|_W = \sqrt{x^*Wx}$ for $x \in \mathbb{C}^n$, propose an accurate (no stability proof required), reasonably efficient algorithm to solve the *weighted-least-squares* problem: find $x \in \mathbb{C}^n$ to minimize $\|Ax - b\|_W$ for any $m \times n$ matrix A with full column rank.