18.335 Midterm, Fall 2013

Each problem has **equal weight**. You have 1 hour and 55 minutes.

Problem 1: GMRES (20 points)

From class, the GMRES algorithm iteratively builds up an orthonormal basis Q_n for the Krylov space $\mathscr{K}_n = \text{span}\langle b, Ab, \ldots, A^{n-1}b\rangle$ and then uses this basis to solve $\min_{x\in\mathscr{K}_n} \|Ax - b\|_2$.

(a) We normally assume that each iteration n gives us a linearly independent vector, i.e. that $A^n b$ is not in $\mathscr{K}_n$. What happens if this is false, i.e. $A^n b \in \mathscr{K}_n$ ("breakdown")? Show that in the (unlikely) event that this occurs, it is a *good* thing, not a bad thing, for solving $Ax = b$.

(b) Given an $m \times m$ A (which you can assume to be diagonalizable), how would you (theoretically) construct a b such that breakdown occurs after $n < m$ steps (in exact arithmetic)?

For reference, the GMRES algorithm is listed below.

$q_1 = b/\|b\|_2$
`for` $n = 1, 2, \ldots$

  $v = Aq_n$
  `for` $j = 1, 2, \ldots, n$
    $h_{jn} = q_j^* v$
    $v = v - h_{jn} q_j$
  $h_{n+1,n} = \|v\|_2$
  $q_{n+1} = v/h_{n+1,n}$
  `solve` $\min_{x\in\mathscr{K}_n} \|Ax - b\|_2 \implies \min_{y\in\mathbb{C}^n} \quad \tilde{H}_n y - e_1\|b\|_2 \;_2 \implies x_{n+1} = Q_n y$

Problem 2: Conditioning (20 points)

The following parts can be solved *independently*.

(a) Suppose that A is an $m \times n$ matrix (of rank $n < m$). In some applications, only certain elements C_{ij} of $C = (A^*A)^{-1}$ are required. If you are given a few desired i and j, outline an efficient, well-conditioned algorithm to compute those C_{ij}. (You can use as subroutines any of the algorithms described in class...you need not reproduce their details here.)

(b) Compare the condition numbers of $f(x) = Ax$ and $f(A) = Ax$ (for $A \in \mathbb{C}^{m\times n}$ and $x \in \mathbb{C}^n$), using the L_2 norm (and an L_2 induced norm for matrices).

- Recall that, for a differentiable function $g(z)$ mapping $z \in \mathbb{C}^p$ to $g(z) \in \mathbb{C}^q$, the condition number is $\kappa(z) = \frac{\|J\|}{\|g(z)\|/\|z\|}$ where $\|J\|$ is the induced norm ($\sup_{z\neq 0} \frac{\|Jz\|}{\|z\|}$) of the Jacobian matrix $J_{ij} = \frac{\partial g_i}{\partial z_j}$.

Problem 3: QR updating (20 points).

Suppose you are given the QR factorization $A = QR$ of an $m \times n$ matrix A (rank $n < m$). Describe an efficient $O(m^2 + n^2) = O(m^2)$ algorithm to compute the QR factorization of a rank-1 update to A, that is to factorize $A + uv^* = Q'R'$ for some vectors $u \in \mathbb{C}^m$ and $v \in \mathbb{C}^n$, following these steps:

(a) Show that $Q'R' = Q(R + zv^*)$ for some z that can be computed in $O(m^2)$ operations. Therefore, we just need to find a unitary matrix that (acting on the left) re-triangularizes $R + zv^*$ to get R' (and Q', which may be stored implicitly in terms of a sequence of rotations).

(b) Every column of zv^* is proportional to the same vector z. Using this fact, explain how we can apply Givens rotations (from the bottom row to the top) which rotate z into a multiple of e_1, in order to convert $R + zv^*$ into **upper-Hessenberg** form using $O(n^2)$ operations. Recall from homework that a Givens rotation is a 2×2 unitary matrix that rotates $\begin{pmatrix} a \\ b \end{pmatrix} \to \begin{pmatrix} \# \\ 0 \end{pmatrix}$.

(c) From the upper-Hessenberg form in the previous part, explain how we can unitarily convert back to upper-triangular form in $O(n^2)$ operations.