18.335 Midterm, Fall 2011

Problem 1: (10+15 points)

Suppose $A$ is a diagonalizable matrix with eigenvectors $v_1$ and eigenvalues $\lambda_k$, in decreasing order $|\lambda_1| \geq |\lambda_2| \geq \cdots$. Recall that the power method starts with a random $x$ and repeatedly computes $x \leftarrow Ax/\|Ax\|_2$.

(a) Suppose $|\lambda_1| = |\lambda_2| > |\lambda_3|$, but $\lambda_1 \neq \lambda_2$. Explain why the power method will not in general converge.

(b) Give a simple fix to obtain $\lambda_1$ and $\lambda_2$ and $v_1$ and $v_2$ from the power method or some small modification thereof. (No fair going to some much more complicated/expensive algorithm like inverse iteration, Arnoldi, QR, or simultaneous iteration!)

Problem 2: (25 points)

Review: We described GMRES as minimizing the norm $\|r\|_2$ of the residual $r = b - Ax$ over all $x \in \mathcal{X}_n$ where $\mathcal{X}_n = \text{span}(b, Ab, \ldots, A^{n-1}b)$. This was done using Arnoldi (starting with $q_1 = b/\|b\|_2$) to build up an orthonormal basis $Q_n$ of $A$, where $AQ_n = Q_{n+1} \tilde{H}_n$ ($\tilde{H}_n$ being an $(n+1) \times n$ upper-Hessenberg matrix), in terms of which we wrote $x = Q_n y$ and solved the least-square problem $\min_{y} \| \tilde{H}_n y - b e_1 \|_2$ where $b = \| b \|_2$ and $e_1 = (1, 0, 0, \ldots)^T$ (since $b = Q_{n+1} b e_1$).

- Suppose, after $n$ steps, we want to restart GMRES. That is, we want to restart our Arnoldi process with one vector $\tilde{q}_1$ based (somehow) on the solution $x_0 = Q_n y$ from the $n$-th step, and build up a new Krylov space. What should $\tilde{q}_1$ be, and what minimal-residual problem should we solve on each step of the new GMRES iterations, to obtain improved solutions $x$ in some Krylov space?

(Note: if you’re remembering implicitly restarted Lanczos now and panicking, relax: all the complexity there was to restart with a subspace of dimension $> 1$, which doesn’t apply when we are restarting with only one vector. Think simpler.)

(Note: be sure to obtain a small least-squared problem on each step. No $m \times n$ problems! This may screw up the first thing you try. Hint: think about residuals.)

Problem 3: (15+10 points)

(a) The following two sub-parts can be solved independently (you can answer the second part even if you fail to prove the first part):

(i) Suppose $A$ is an $m \times n$ matrix with rank $n$ (i.e., independent columns). Let $B = A_{1:p}$ be the first $p$ $(1 \leq p \leq n)$ columns of $A$. Show that $\kappa(A) \geq \kappa(B)$. (Hint: recall that our first way of defining $\kappa(A)$ was by $\kappa(A) = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$.)

(ii) Suppose that we are doing least-square fitting of a bunch of data points (containing some experimental errors) to a polynomial. Does the $\kappa(A) \geq \kappa(B)$ result from the previous part tell you about what happens about the sensitivity to errors as you increase the number of data points or as you increase the degree of the polynomial, and what does it tell you?

(b) Prove that if $\kappa(A) = 1$ then $A = cQ$ where $Q^*Q = I$ and $c$ is some scalar. (The SVD definition of $\kappa$ might be easiest here: $\kappa(A) = \frac{\sigma_{\max}}{\sigma_{\min}}$ when $A$ has full column rank.)

Problem 4: (8+8+9 points)

Recall that an IEEE double-precision binary floating-point number is of the form $\pm s \times 2^e$ where the significand $s = 1.xxx\ldots$ has 53 binary digits (about 16 decimal digits, $\varepsilon_{\text{machine}} \approx 10^{-16}$) and the exponent $e$ has 11 binary digits ($e \in [-1022, 1023] \implies 10^{-308} \leq 2^e \leq 10^{308}$).

(a) Computing $\sqrt{x^2 + y^2}$ by the obvious method, $\sqrt{(x \times x) + (y \times y)}$ sometimes yields “$\infty$” (Inf) even when $x$ and $y$ are well within the representable range. Propose a solution.

(b) Explain why solving $x^2 + 2bx + 1 = 0$ for $x$ by the usual quadratic formula $x = -b \pm \sqrt{b^2 - 1}$ might be very inaccurate for some $b$, and propose a solution.

(c) How might you compute $1 - \cos x$ accurately for small $|x|$? Assume you have floating-point sin and cos functions that compute exactly rounded results, i.e. $\sin x = \text{fl}(\sin x)$ and $\cos x = \text{fl}(\cos x)$.