18.335 Midterm Solutions, Fall 2011

Problem 1: (10+15 points)

(a) After many iterations of the power method, the λ_1 and λ_2 terms will dominate:

$$\mathbf{x} \approx c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2$$

for some c_1 and c_2. However, this is not an eigenvector. Multiplying this by A gives $\lambda_1 c_1 \mathbf{v}_1 + \lambda_2 c_2 \mathbf{v}_2 = \lambda_1 \left(c_1 \mathbf{v}_1 + \frac{\lambda_2}{\lambda_1} c_2 \mathbf{v_2} \right)$, which is not a multiple of $\mathbf{x}$ and hence will be a different vector after normalizing, meaning that it does not converge to any fixed vector.

(b) The key point is that if we look at the vectors $\mathbf{x} \approx c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2$ and $\mathbf{y} \approx \lambda_1 c_1 \mathbf{v}_1 + \lambda_2 c_2 \mathbf{v}_2$ from **two subsequent** iterations, then after **many iterations** these are *linearly independent* vectors that *span the two desired eigenvectors*. We can then employ e.g. a Rayleigh–Ritz procedure to find $\mathbf{v}_1$ and $\mathbf{v}_2$: use Gram–Schmidt to find an orthonormal basis $\mathbf{q}_1 = \mathbf{x}/\|\mathbf{x}\|_2$ and $\mathbf{q}_2 = (\mathbf{y} - \mathbf{q}_1 \mathbf{q}_1^* \mathbf{y})/\|\cdots\|_2$, form the matrix $Q = (\mathbf{q}_1, \mathbf{q}_2)$ and find the 2×2 matrix $A_2 = Q^* A Q$. The eigenvalues of A_2 (the Ritz values) will then converge to the eigenvalues λ_1 and λ_2 and we obtain $\mathbf{v}_1$ and $\mathbf{v}_2$ (or some multiple thereof) from the corresponding Ritz vectors. The key point is that AQ is in the span of $\mathbf{q}_1$ and $\mathbf{q}_2$ (in the limit of many iterations so that other eigenvectors disappear), so the Ritz vectors are eigenvectors.

Of course, since we don't know λ_3 then we don't know how many iterations to run, but we can do the obvious convergence tests: every few iterations, find the Ritz values from the last two iterations, and stop when these Ritz values stop changing to our desired accuracy.

Alternatively, if we form the matrix $X = (\mathbf{x}, \mathbf{y})$ from the vectors of two subsequent iterations, then we know that (after many iterations) the columns of AX are in $C(X) = \text{span}\langle \mathbf{x}, \mathbf{y} \rangle$. Therefore, the problem $AX = XS$, where S is a 2×2 matrix, has an exact solution S. If we then diagonalize $S = Z\Lambda Z^{-1}$ and multiply both sizes by Z, we obtain $AXZ = XZ\Lambda$, and hence the columns of XZ are eigenvectors of A and the eigenvalues $\text{diag}\,\Lambda$ of S are the eigenvalues λ_1 and λ_2 of A. However, this is computationally equivalent to the Rayleigh–Ritz procedure above, since to solve $AX = XS$ for S we would first do a QR factorization $X = QR$, and then solve the normal equations $X^* X S = X^* A X$ via $RS = Q^* A Q R = A_2 R$. Thus, $S = R^{-1} A_2 R$: the S and A_2 eigenproblems are similar; in exact arithmetic, the two approaches will give exactly the same eigenvalues and exactly the same Ritz vectors.

[As yet another alternative, we could write $AXZ = XZ\Lambda$ as above, and then turn it into $(X^* A X)Z = (X^* X)Z\Lambda$, which is a 2×2 *generalized* eigenvalue problem, or $(X^* X)^{-1}(X^* A X)Z = Z\Lambda$, which is an ordinary 2×2 eigenproblem.]

Problem 2: (25 points)

The obvious, but **wrong** thing to do here is to now minimize $\|\mathbf{b} - A\mathbf{x}\|_2$ for $\mathbf{x} \in \mathcal{K}_n$ where $\mathcal{K}_n = \text{span}\langle \mathbf{x}_0, A\mathbf{x}_0, \ldots, A^{n-1}\mathbf{x}_0 \rangle$. This looks superficially like a good approach, because $\mathbf{x}_0 \in \mathcal{K}_n$ and hence the solutions $\mathbf{x}$ will be at least as good as $\mathbf{x}_0$. The problem is that the least-square problem that you end up solving that way is rather inconvenient. Suppose that we again build up a basis Q_n for $\mathcal{K}_n$ by Arnoldi, starting with $\tilde{q}_1 = \mathbf{x}_0/\|\mathbf{x}_0\|_2$. In that case, we can write $\mathbf{x} = Q_n \mathbf{y}$ as before and minimize $\|AQ_n \mathbf{y} - \mathbf{b}\|_2 = \|Q_{n+1}\tilde{H}_n \mathbf{y} - \mathbf{b}\|_2$. In the original GMRES, our next step was to write $\mathbf{b}$ in the Q_{n+1} basis and then factor out the Q_{n+1} to obtain a small $(n+1) \times n$ least-squared problem. Now, however $\mathbf{b} \notin \mathcal{K}_{n+1}$ in general (unless the exact solution is in the Krylov space!), so we cannot do this, and we are left with a huge $m \times n$ least-squares problem. We don't want to solve this numerically, as that would be quite expensive! [In principle, it might not be prohibitive—the cost would be $\theta(mn^2)$, proportional to the cost of n Arnoldi steps, but we can do much better.]

Instead, the key is to minimize $\|\mathbf{b} - A(\mathbf{x}_0 + \mathbf{d})\|_2$ over the *change* $\mathbf{d}$ (in some Krylov space) compared to $\mathbf{x}_0$. Which Krylov space? $\|\mathbf{b} - A(\mathbf{x}_0 + \mathbf{d})\|_2 = \|\mathbf{r}_0 - A\mathbf{d}\|_2$ where $\mathbf{r}_0$ is the initial residual $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, and

from above we need $\mathbf{r}_0$ to be $\in \tilde{\mathscr{K}}_n$ to simplify our least-squares problem, so we want $\boxed{\tilde{\mathscr{K}}_n = \text{span}\langle \mathbf{r}_0, A\mathbf{r}_0, \ldots, A^{n-1}\mathbf{r}_0 \rangle}$.

That is, we start our Arnoldi process with $\boxed{\tilde{q}_1 = \mathbf{r}_0 / \|\mathbf{r}_0\|_2}$ to build up our orthonormal basis Q_n for our new $\tilde{\mathscr{K}}_n$, and then write $\mathbf{d} = Q_n \mathbf{y}$. Now, since $\mathbf{r}_0 = Q_{n+1} r_0 \mathbf{e}_1$ where $r_0 = \|\mathbf{r}_0\|$, we obtain a *small* least-squares problem $\boxed{\min_{\mathbf{y}} \|\tilde{H}_n \mathbf{y} - r_0 \mathbf{e}_1\|_2}$ that we can solve for $\mathbf{y}$ and hence obtain solutions $\boxed{\mathbf{x} = \mathbf{x}_0 + Q_n \mathbf{y}}$. Since these again contain $\mathbf{x}_0$, they clearly must be improvements on the solution when we minimize the residual over all $\mathbf{y}$.

Problem 3: (15+10 points)

(a) Solutions:

(i) Consider $\kappa(B) = \left[\max_{\mathbf{x} \neq 0} \frac{\|B\mathbf{x}\|}{\|\mathbf{x}\|} \right] \cdot \left[\max_{\mathbf{x} \neq 0} \frac{\|\mathbf{x}\|}{\|B\mathbf{x}\|} \right]$. Since B is the first p columns of A, for any $\mathbf{x}$ we can write $B\mathbf{x} = A\mathbf{x}_0$ where $\mathbf{x}_0 = \begin{pmatrix} \mathbf{x} \\ 0 \\ \vdots \end{pmatrix}$ just has extra zero components to multiply the columns $> p$ of A by zero. Note that $\|\mathbf{x}\| = \|\mathbf{x}_0\|$ in the L_2 norm that we usually use for κ (and for that matter would be true in most typical norms). Hence

$$\max_{\mathbf{x} \neq 0} \frac{\|B\mathbf{x}\|}{\|\mathbf{x}\|} = \max_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}_0\|}{\|\mathbf{x}_0\|} \leq \max_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|}$$

since the vectors $\mathbf{x}_0$ are a subset of all vectors in $\mathbb{C}^n$. Similarly for $\frac{\|\mathbf{x}\|}{\|B\mathbf{x}\|}$. Therefore $\kappa(B) \leq \kappa(A)$ as desired.

(ii) In a data-fitting problem, adding columns to the matrix corresponds to increasing the number of degrees of freedom, which in this case means **increasing the degree of the polynomial**. Hence, increasing the degree of the polynomial **increases the sensitivity of the fit coefficients to experimental errors** (recall that condition numbers bound the ratio of relative errors in the output to relative errors in the input).

(b) If $\kappa(A)$ is finite then A has full column rank. Write the reduced SVD $A = \hat{U}\hat{\Sigma}\hat{V}^*$ where $\hat{U}$ is $m \times n$, $\hat{\Sigma}$ is the $n \times n$ diagonal matrix of the n singular values, and $\hat{V}$ is $n \times n$. $\kappa(A) = 1$ implies $\sigma_{\min} = \sigma_{\max}$, which means that all the singular values are equal to some value σ and $\hat{\Sigma} = \sigma I$ is a multiple of the $n \times n$ identity matrix. Hence $A = \sigma \hat{U}\hat{V}^* = \sigma Q$ where $Q = \hat{U}\hat{V}^*$ satisfies $Q^*Q = \hat{V}\hat{U}^*\hat{U}\hat{V}^* = \hat{V}\hat{V}^* = I$ (noting that $\hat{V}$ is square hence unitary). Therefore $A = cQ$ with $\boxed{c = \sigma}$ and $\boxed{Q = \hat{U}\hat{V}^*}$.

Problem 4: (8+8+9 points)

(a) The problem is that large x or large y (magnitude $\gtrsim 10^{154}$) can cause x^2 or y^2 to overflow and give ∞, even though the result $\sqrt{x^2 + y^2}$ should be in a representable range. This fix is simple: let $z = \max(|x|, |y|)$ (which can be computed without multiplications or danger of overflow), and write

$$\sqrt{x^2 + y^2} = z\sqrt{(x/z)^2 + (y/z)^2} \approx z \otimes \sqrt{(x \oslash z) \otimes (x \oslash z) \oplus (y \oslash z) \otimes (y \oslash z)}.$$

Since we are now multiplying rescaled numbers with magnitudes ≤ 1, there is no danger of overflow and we can expect an accurate result for all representable numbers.

(b) The problem with $x = -b \pm \sqrt{b^2 - 1}$ is that for very large b, if $b^2 > 1/\varepsilon_{\text{machine}}$ then $b^2 \ominus 1$ will give b^2, and one of the two roots will have extremely large relative error. Say $b > 0$, then $\widetilde{-b + \sqrt{b^2 - 1}} \approx -b \oplus \sqrt{b \otimes b}$ will be *all* roundoff errors.

One way to fix this is to check for the case of large $|b|$ and to Taylor-expand the second root in that case. $\sqrt{1+z} = 1 + \frac{1}{2}z - \frac{1}{8}z^2 + O(z^3)$, so the problematic root can be approximated by $-b + b\sqrt{1 - \frac{1}{b^2}} = -b + b\left(1 - \frac{1}{2b^2} - \frac{1}{8b^4} + O(b^{-6})\right) \approx -\frac{1}{2b} - \frac{1}{8b^3}$. This will give us a much more accurate answer. The next-order term in the Taylor expansion would be $-bz^3/16 = -1/16b^5$, or a relative error $\approx \frac{1/16b^5}{1/2b} = \frac{1}{8b^4}$, which is less that $\varepsilon_{\text{machine}}$ in double precision for $|b| \gtrsim 5000$. So, we can use the two-term Taylor expansion to get extremely accurate answers as soon as $|b| > 5000$, and use the exact expression otherwise.

Another, even nicer solution, is to use the fact that the product of the two roots is 1, so we can instead write the two roots (for $b > 0$) as $-b - \sqrt{b^2 - 1}$ and $\frac{1}{-b - \sqrt{b^2 - 1}}$ (switching to $-b + \sqrt{\cdots}$ for $b < 0$), which are exact expressions that suffer no cancellation problems for large $|b|$.

A *separate* but less dramatic problem occurs for $b = \pm 1 + \delta$ where $|\delta|$ is very small (as can be seen by taking the derivative with b, the condition numbers of *both* roots diverge as $\delta \to 0$). Suppose $|\delta| \sim \varepsilon_{\text{machine}}$ and $b = 1 + \delta$. Then $b \otimes b \ominus 1 \approx 2\delta + \varepsilon_4 + O(\varepsilon_{\text{machine}}^2)$ where $|\varepsilon_4| \le 4\varepsilon_{\text{machine}}$ (where the 4 comes from combining the various ε factors for different operations in $b^2 - 1$). Then $-b + \sqrt{b^2 - 1}$ is computed as $-1 - \delta + \sqrt{2\delta + \varepsilon_4} + \varepsilon + O(\varepsilon_{\text{machine}}^2)$ where $|\varepsilon| \le 4\varepsilon_{\text{machine}}$, and the error (both absolute and relative) is $\sqrt{2\delta + \varepsilon_4} - \sqrt{2\delta} + O(\varepsilon_{\text{machine}}) = O(\sqrt{\varepsilon_{\text{machine}}})$. So, although the error is not large in absolute terms, it converges more slowly than would be required by the strict stability criteria. Essentially, once the user computes $1 + \eta$ we are sunk since we have thrown out most or all of the digits of η for small $|\eta|$. The only way to fix this is to have the user specify something like $\eta = b - 1$ instead of b, in which case we can compute $1 + \eta \pm \sqrt{2\eta + \eta^2}$ with $O(\varepsilon_{\text{machine}})$ errors even for very small $|\eta|$. However, the large-$|b|$ failure is much more dramatic and problematic.

(c) The problem is that for small $|x|$, $\cos x \approx 1$ and so $1 - \cos x$ will suffer a large cancellation error. One approach is to check whether $|x|$ is small and to switch to a Taylor expansion in this case, as in the previous problem, and that is a perfectly valid solution. Another, somewhat more elegant approach, is to use the trigonometric identity $1 - \cos x = 2\sin^2(x/2)$, which involves no subtractive cancellation and will give us nearly machine precision for a wide range of x. [There are various other trig identities you could use instead, e.g. $1 - \cos^2 x = \sin^2 x = (1 - \cos x)(1 + \cos x)$ and hence $1 - \cos x = \sin^2 x/(1 + \cos x)$, which is accurate for small $|x|$, but this is more expensive than the half-angle formula.]

(Note that there is a very different problem for very *large* x. For very large x, evaluating $\cos x$ requires the computation of the *remainder* after dividing x by π. So, for example, if $x = 10^{100}$ then one would need to compute x/π to ~ 115 decimal digits in order to evaluate $\cos x$ to machine precision. [There is a famous story in which Paul Olum stumped Richard Feynman by posing a similar question.] Because of this, in in practice numerical libraries only compute sin and cos accurately if you give them angles not too much bigger than π.)

3