$\frac{80}{100}$

# Fall 2013, CSE 383, Numerical Linear Algebra, Final
## Saturday, December 14, 2013,

This is a closed-book exam. Please explain all your answers.
Questions sum to 120 points. You only need 100 points. (The last question is 20 points.)

**YOUR NAME :** Stephen Shannon

-3

1. **[10 points,** ] Let $A$ be a $m \times n$ real matrix and $A = USV^T$ be its reduced SVD. Let $r_i$ denote the rows of $A$, $c_j$ the columns of $A$, $u_k$ the columns of $U$, $v_l$ the columns of $V$, and $\sigma_\nu$ the singular values of $A$. By span$\{r_i\}$ we mean the span of $r_1, r_2, \ldots, r_m$. Similar notation is used for the other set of vectors. For each of the following statements, briefly explain if it is true or false and why.

   ✓ (a) span$\{r_i\}$ = span$\{u_k\}$ False – span$\{c_j\}$ = span$\{u_k\}$, span$\{r_i\}$ = span$\{v_k\}$

   ✓ (b) $Az = 0 \Rightarrow z \in$ span$\{r_i\}$ False – $Az = 0 \Rightarrow z \in Null(A)$

   Assuming $m > n$ ✓ (c) span$\{u_k\} = \mathbb{R}^m$ False – span$\{u_k\} \subseteq \mathbb{R}^n$ $\quad c_j \in \mathbb{R}^m$

   ✗ (d) span$\{c_j\} \subseteq \mathbb{R}^n$ True – $\{c_j\}$ spans column space of $A$, which is $\subseteq \mathbb{R}^n$

   ✓ (e) span$\{r_i\}$ = span$\{v_l\}$ True – right singular vectors span rowspace of $A$

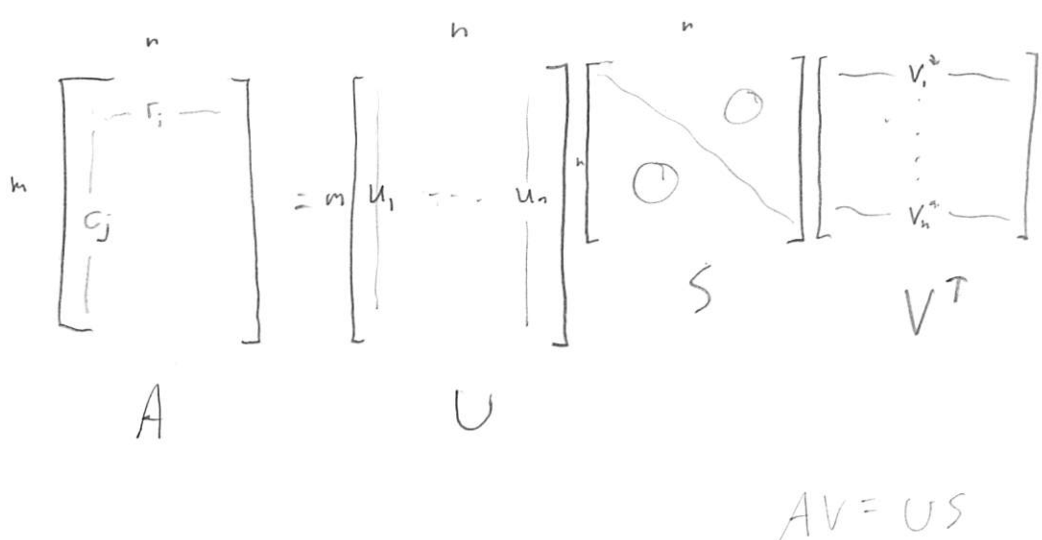
   $Ax = 0,\ x \neq 0$ ✗ (f) $\min_{x \in \mathbb{R}^n, \|x\|_2 = 1} \|Ax\|_2 = \min_\nu \sigma_\nu$ True – kind of similar to explanation of (i)

   ✓ (g) $\min_{x \in \text{span}\{v_l\}, \|x\|_2 = 1} \|Ax\|_2 = \min_\nu \sigma_\nu$ True (always true for V)

   ✗ (h) $\|A\|_\infty = \max_\nu \sigma_\nu$ False

   ✓ (i) $\|Av_2\|_2 = \sigma_2$ True – $Av_2 = US_2 = u_2\sigma_2$ $\quad \|u\|_2 = 1 \Rightarrow \|u_2\sigma_2\|_2 = \sigma_2 = \|Av_2\|_2$

   ✓ (j) $\dim(\text{null}(A^T)) = \dim(\text{null}(A))$ False (if $m \neq n$) $\dim(\text{range}(A)) = \dim(\text{range}(A^T)) = k$
   Rank-nullity $\Rightarrow \dim(\text{null}(A)) + k = n \Rightarrow \dim(\text{null}(A)) = n - k$
   $\dim(\text{null}(A^T)) + k = m \Rightarrow \dim(\text{null}(A^T)) = m - k$




$AV = US$

2. [**5 points**, ] We solve $Ax = b$ using a backward stable algorithm, where $A$ is a square matrix. $\|A\|_2 = 10$ and $\|A^{-1}\|_2 = 1E6$. What is the expected relative accuracy in the computed $x$ in the presence of rounding errors in $b$ and $A$?

We want $\frac{\|\delta x\|}{\|x\|}$

Rounding errors in $A$: $(A+\delta A)(x+\delta x) = b \Rightarrow Ax + A\delta x + \delta A x + \boxed{\delta A \delta x} = b$ negligible

$A\delta x + \delta A x = 0$, $A\delta x = -\delta A x$, $\delta x = -A^{-1}\delta A x$

$\|\delta x\| \le \|A^{-1}\|\|\delta A\|\|x\| \Rightarrow \frac{\|\delta x\|}{\|x\|} \le 10^6 \|\delta A\|$

$\|A\|\|A^{-1}\| \cdot 10^7 \epsilon_M = 10^{-9}$

-2

Rounding errors in $b$: $A(x+\delta x) = b+\delta b \Rightarrow A\delta x = \delta b$, $\delta x = A^{-1}\delta b$

$\|\delta x\| \le \|A^{-1}\|\|\delta b\| \Rightarrow \frac{\|\delta x\|}{\|x\|} \le 10^6 \frac{\|\delta b\|}{\|x\|}$

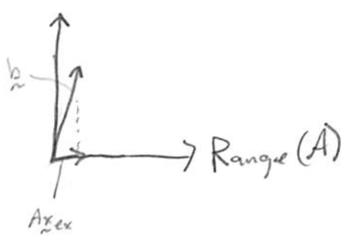
3. [**8 points**, ] We wish to solve $\min_x \|Ax - b\|_2$ using the QR factorization algorithm, where $A$ is full rank $\in \mathbb{R}^{m \times n}$, $m > n$. $\|A\|_2 = 10$, $\|A^{-1}\|_2 = 1E6$, and $\cos^{-1}(\|Ax_{ex}\|_2/\|b\|_2) = 0.01$, where $x_{ex}$ is the exact solution to the least square problem. What is the expected relative accuracy in the computed $x$ in the presence of rounding errors in $b$ and $A$?

-7

This means $b$ is nearly orthogonal to range$(A)$. I expect high relative error.

Least squares solution

$x = \underbrace{(A^*A)^{-1}A^*}_{\text{pseudoinverse of } A} b$




→ You don't want to use normal equations.

I'm sorry, I have a lot of trouble with error analysis without consulting a book. I remember $\|A\|\|A^{-1}\| = K$...

QR: $K(A)/\cos\theta + K^2(A)\tan\theta$.

4. [**12 points**, ] Answer the following questions for the LU, SVD, Cholesky, and QR factorizations:

(a) Does the factorization always exist?

(b) Is the factorization unique?

(c) Is there a backward stable algorithm for the computation of the factorization?

-2

(a) LU - yes, SVD - yes, QR - yes

(b) LU - no, SVD - yes, QR - yes, to a sign

(c) LU - yes, SVD - yes, QR - yes

No; A not invertible

Yes; if it exists; up to normalization of the diagonal.

5. [**45 points**, *Multiple choice part*] The following questions require a choice followed by a brief explanation, or a reasonably short answer (usually at most 1-2 sentences or a figure). No credit will be given for answers without a correct explanation.

(a) Let $P$ be an orthogonal projection operator, and $x$ a vector. Then: (A) $\|Px\|_2 \leq \|x\|$; (B) $\|Px\|_2 \geq \|x\|$; (C) $\|Px\|_2 = \|x\|_2$.

If $Px = x$, then $\|Px\|_2 = \|x\|_2$. If $Px \neq x$, then $x$ is "losing some components" and the Eucliden length will be shorter than that of $x$.

(b) Let $N$ be the number of floating numbers in $[16, 32)$ and $M$ the number of floating numbers in $[128, 192)$. Then (A) $N < M$; (B) $N > M$; (C) $N = M$.

There are the same amount of floating point numbers in every interval $[2^E, 2^{E+1})$. $[16,32)$ represents a full interval while $[128,192)$ represents a half interval. $(N = 2M)$

(c) Suppose two numbers $x$ and $y$ can be represented exactly in the IEEE-764 standard. Then, using a IEEE-764 compatible machine their difference $x - y$: (A) is guaranteed to be computed without rounding error; (B) has error proportional to machine epsilon; (C) has relative error proportional to machine epsilon. (Ignore underflow and overflow.)

$$x \circledcirc y = (x \triangledown y)(1+\delta) \text{ where } \delta \leq \varepsilon_M$$

floating point numbers

floating point op

relative error

$$\Rightarrow \quad \Rightarrow \left| \frac{x \circledcirc y - x \triangle y}{x \circledcirc y} \right| = O(\delta) = O(\varepsilon_M).$$

(d) A Householder transformation can be used to: (A) normalize a vector; (B) project a vector onto a plane; (C) align a vector with another vector.

A Householder transformation does not project, but rather reflects ("projects to the reflection plane and keeps going twice the distance"). In practice, we use Householder transformations to align a vector with another vector.

(e) $A$ is dense, symmetric and positive definite and we wish to solve $Ax = b$. Which method will be faster? (A) LU; (B) QR; (C) Cholesky.

Cholesky takes advantage of symmetry and is twice as fast as LU, which is faster than QR.

(f) A unitary transformation of a vector is: (A) well conditioned; (B) ill-conditioned; (C) it depends on the unitary transformation.

Inner products, for example, are ill-conditioned when the 2 vectors are close to orthogonal. If the unitary transformation involves inner products, it may be ill-conditioned.

-3

But they are not unitary!

(g) Backward stability of an algorithm: (A) is possible only if a problem is well-conditioned; (B) guarantees that the numerical result using this algorithm has small relative error; (C) guarantees that the numerical result using this algorithm has a relative error that is proportional to the conditioning of the underlying problem.

2 Stability describes the algorithm, not the underlying problem.

-3 ✗ (h) Suppose $A$ is dense and square but you don't know if it is singular or not. Which method you would use to solve $Ax = b$? (A) Pivoted QR; (B) CG; (C) QR; (D) Pivoted LU factorization.

We don't need to pivot if it's dense

QR may fail if A is singular. Only pivoted QR will work

(i) A similarity transformation of a matrix preserves: (A) its range space; (B) its eigenvalues; (C) its singular values.

✓ Say $A = X\Lambda X^{-1}$ is eigenvalue decomp. of $A$

Say $\underbrace{B = YAY^{-1}}_{\text{similarity transformation}} = YX\Lambda X^{-1}Y^{-1} = \underbrace{(YX)\Lambda(YX)^{-1}}_{\text{e-value decomp of } B}$ – same e-values as $A$

(j) If a complex matrix $A$ of size $m \times m$ is diagonalizable then the following is true: (A) $A$ is normal; (B) $A$ is symmetric; (C) $A$'s eigevectors are orthogonal; (D) $A$'s eigenvectors form a basis for $\mathbb{C}^m$.

✓ A and B are diagonalizable, but a matrix need not be normal to be diagonalizable. And C is only true if B is true.

(k) Suppose you know a matrix $A$ is defective. Which factorization will reveal its eigenvalues? (A) SVD; (B) Schur; (C) the eigenvalue decomposition; (D) QR.

✓ Schur factorizations can be performed on any square matrix.

(l) Let $A$ be a real symmetric matrix. Which method would you use for computing an eigenvalue (with multiplicity one) of A and the corresponding eigenvector, given a reasonably accurate initial guess for the value of the target eigenvalue? (A) Rayleigh quotient iteration; (B) Power iteration; (C) the inverse iteration.

✓ Power iteration only solves for eigenvector corresponding to largest eigenvalue. Inverse iteration calculates eigenvalue. The Rayleigh Quotient iteration combines the two.

-2

(m) $A$ is sparse, square, real, symmetric, well-conditioned, and positive definite. Which method would you use to solve $Ax = b$? (A) LU; (B) Cholesky; (C) Conjugate Gradients (CG).

Cholesky factorization can take advantage of sparsity (at least in MATLAB) to reduce computational costs even more.

C: CG is the fastest

(n) We solve $Ax = b$ with CG. $A$ is SPD, ill-conditioned has only four distinct eigenvalues. The number of CG iterations will be (A) 4; (B) $\leq 4$; (C) greater than 4.

CG converges in no more iterations than # of distinct eigenvalues.

(o) We solve $Ax = b$ with the steepest descent method (SD). $A$ is as above. The number of SD iterations will be (A) 4; (B) $\leq 4$; (C) greater than 4.

Ill-conditioned problems tend to "bounce back and forth" with the steepest descent method.

Conjugate Gradient

(p) In the absense of rounding errors, the residual between iteration $k$ and $k+1$ (A) reduces; (B) increases; (C) remains the same; (D) none of the above.

Unlike the steepest descent method, the A-orthogonality condition avoids that the residual increases on any given iteration (we saw this in HW6 pr. 4)

; $\|e_k\|_A$ reduces. But can't say anything about residual.

---

6. **[10 points, ]** Let

$$A = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

e-vectors of e-value = 2

$$\left\{\begin{pmatrix}0\\1\\0\\0\end{pmatrix}\right\} \left\{\begin{pmatrix}0\\0\\1\\0\end{pmatrix}\right\}$$

(A) What is the characteristic polynomial of $A$? (B) State the algebraic and geometric multiplicity of each eigenvalue of $A$.

(A) $p_A = \det(A - I\lambda) = (-1-\lambda)(2-\lambda)(2-\lambda)(2-\lambda)$

(B) We have 2 distinct eigenvalues: $\lambda_1 = \lambda_2 = \lambda_3 = 2$, $\lambda_4 = -1$

$\lambda_1$: algebraic multiplicity of 3 (appears 3 times)
geometric multiplicity of 2 (only 2 corresponding LI e-vectors)

$\lambda_2$: algebraic multiplicity of 1 (appears once)
geometric multiplicity of 1 (one LI eigenvector)

7. [**5 points**, ] Let $A$ be a real symmetric positive definite matrix. Give an algorithm for computing the matrix function $\sqrt{A}$.

- Use your method of choice to diagonalize $A$ (QR iteration, for example)
- Now we have $A = Q\Lambda Q^*$ — take sqrt of elements of $\Lambda$ to get $\Lambda^{\frac{1}{2}}$

  $\uparrow$ unitary $\quad \uparrow$ diagonal matrix
- We're done; $A^{\frac{1}{2}} = Q\Lambda^{\frac{1}{2}}Q^*$

  (To check: $(A^{\frac{1}{2}})^2 = (Q\Lambda^{\frac{1}{2}}Q^*)(Q\Lambda^{\frac{1}{2}}Q^*) = Q\Lambda Q = A$

8. [**5 points**, ] Let

$$A = \begin{pmatrix} \frac{81}{10} & 10 & \frac{44}{5} & \frac{17}{2} \\ 0 & \frac{101}{10} & \frac{24}{5} & \frac{17}{10} \\ 0 & 0 & \frac{36}{5} & \frac{19}{5} \\ 0 & 0 & 0 & \frac{9}{5} \end{pmatrix}$$

Compute its eigenvalues.

$p_A(\lambda) = 0 \Rightarrow \lambda$ is eigenvalue. $p_A(\lambda) = (\frac{81}{10} - \lambda)(\frac{101}{10} - \lambda)(\frac{36}{5} - \lambda)(\frac{9}{5} - \lambda)$

(I could have just said that ~~upper~~ triangular matrices have their eigenvalues on the diagonal)

$$\lambda_1 = \frac{101}{10}, \quad \lambda_2 = \frac{81}{10}, \quad \lambda_3 = \frac{36}{5}, \quad \lambda_4 = \frac{9}{5}$$

9. [**20 points**, *Hard*] We wish to solve a least squares system $\min_x \|Ax - b\|_2$, where $A \in \mathbb{R}^{m \times n}$, $m < n$, and rank $A \leq m$. $A$ has $\mathcal{O}(n)$ non-zero entries. Assuming you only have $\mathcal{O}(n)$ computer memory, (A) suggest an algorithm for approximating the solution to the least squares problem. (B) What is the expected number of iterations and the overall computational cost of your algorithm? (C) What is the expected error? (You will need to provide an appropriate definition of error for this problem.)