Assignment 2: Norms, Inner Products, and Linear System Solving

ACM 106a: Introductory Methods of Computational Mathematics (Fall 2013)

Due date: Thursday, October 17, 2013

1 Equivalence of norms

(a) Prove that all vector norms on $\mathbf{R}^n$ and $\mathbf{C}^n$ are equivalent: show that for any pair of norms $\|\cdot\|_a$ and $\|\cdot\|_b$ there exist constants m, M, independent of x, such that

$$m\|x\|_a \leq \|x\|_b \leq M\|x\|_a$$

for all x.

(b) Derive the constants m and M for $a, b \in \{1, 2, \infty\}$. Specifically, show that

$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$$
$$\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2$$

for all $x \in \mathbf{R}^n$.

2 Operator norm of the inverse

Recall that the matrix operator norm induced by the vector norm $\|\cdot\|$ is given by

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

Use this definition to prove the identity

$$\|A^{-1}\|^{-1} = \frac{1}{\|A^{-1}\|} = \min_{y \neq 0} \frac{\|Ay\|}{\|y\|}$$

for all nonsingular A.

3 Floating point dot and matrix products

(a) Show that, for any $x, y \in \mathbf{R}^d$, we have

$$fl(x^T y) = fl\left(\sum_{i=1}^d x_i y_i\right) = \sum_{i=1}^d x_i y_i (1 + \delta_i)$$

for some $\delta_i \in [-d\epsilon, d\epsilon]$.

(b) Use the result of part (a) to show that

$$(|fl(AB) - AB|)_{ij} \leq n\epsilon(|A||B|)_{ij}$$

for all $A, B \in \mathbf{R}^{n \times n}$. Here $|X|$ denotes the $n \times n$ matrix with (i,j) entry given by $(|X|)_{ij} = |X_{ij}|$.

4 Linear systems and rank-one error

Suppose we can solve the nonsingular system $Ax = b$ efficiently, i.e., we have a fast, accurate algorithm to do so. In this exercise, we verify that we can also solve the system $\tilde{A}x = b$ where $\tilde{A} = A + E$, for some rank-one matrix E.

(a) Prove that if $E \in \mathbf{R}^{n \times n}$ has rank equal to one, i.e., the columns of E span a one-dimensional vector space, then E must have the factorization

$$E = uv^T$$

for some $u, v \in \mathbf{R}^n$.

(b) Establish the **Sherman-Morrison formula**: show that $A + uv^T$ is nonsingular if $v^T A^{-1} u \neq -1$ and

$$(A + uv^T)^{-1} = A^{-1} - \sigma A^{-1} uv^T A^{-1}$$

for some σ in this case. Calculate σ.

Hint: first derive the formula for $I + uv^T$ and extend to general A.

(c) Write (in pseudocode) an algorithm for solving $\tilde{A}x = b$ using the Sherman-Morrison formula. You may use as a subroutine the given algorithm for solving $Ax = b$; that is, your algorithm can include operations of the form "solve $Ax = b$" without providing further detail on how $Ax = b$ is solved.

Suppose that $Ax = b$ can be solved in M flops, where M satisfies $n = O(M)$. Your algorithm should require only $O(M)$ operations; provide a calculation of the number of flops required by your algorithm to verify this property is satisfied.

(d) Suppose that A is an orthogonal matrix. Implement your algorithm from part (c) in Matlab; your function should take as input A, u, v, b.

Compare performance of your algorithm with Matlab's built-in linear system solver $A\backslash b$ (N.B. Matlab's $A \setminus b$ solutions are computed using Gaussian elimination using partial pivoting). For $n = 10, 100, 500$ do the following:

- Generate 25 random permuation matrices A and right-hand side vectors using the commands

$$A = \text{eye}(n); \quad idx = \text{randperm}(1:n); \quad A = A(idx, :); \quad b = \text{rand}(n, 1);$$

- Generate 25 random rank-one perturbations uv^T by the commands

$$u = \text{rand}(n, 1); \quad v = \text{rand}(n, 1);$$

- For each of the 25 program instances, find the solution of $\tilde{A}x = b$ using your algorithm from part (c), call this solution $\tilde{x}$, and the solution given by the command $x = A \setminus b$ in Matlab.
- Summarize your findings in a table of the form:

	$n = 10$	$n = 100$	$n = 500$
$\Vert\tilde{x} - x\Vert_2$			
Time: my_alg			
Time: $A \setminus b$			

For each desired quantity, report the mean and standard deviation across all 25 trials for each n in the form: **mean (std dev)** in scientific notation to machine precision (format long e). Explain carefully any observed differences between the two algorithms.

Instructions:

- Assignments are due at the **start** of class (1pm) on the due date.
- Write your name and ID# clearly on all pages, and underline your last name.
- **Matlab files:** please submit a single zip/rar/etc file with file name in the format **Lastname_Firstname_ID#_A2** to **homework.acm106a@gmail.com** that decompresses to a single folder of the same name containing the following:
  - A single thoroughly commented Matlab script file containing all commands used for each assigned programming/simulation problem. File names should have format **Lastname_Firstname_ID#_A2_P#** :

    e.g. **Ames_Brendan_12345678_A2_P4** for problem 4.
  - All Matlab functions used to perform any assigned programming/simulation problems with appropriate file names. Please add a comment to the beginning of each file with the format **Lastname_Firstname_ID#_A2_P#**
  - A diary file of your session for each programming/simulation problem with file name in the format **Lastname_Firstname_ID#_A2_P#_DIARY.txt**. Please also submit a hard copy of the diary and any relevant derivations, pseudocode, etc. you may want considered for partial credit with your submitted solution sets at the beginning of class.