Exercise Sheet 11

Due Thursday, January 15 (before class/exercise)
http://www-ai.math.uni-wuppertal.de/~dfritzsche/lehre/nla_ws08

Exercise 1 (4 points)
Consider an algorithm for the problem of computing the (full) SVD of a matrix. The
data for the problem is a matrix $A$, and the solution is three matrices $U$ (unitary),
$\Sigma$ (diagonal), und $V$ (unitary) such that $A = U\Sigma V^*$. Note: We are speaking here of
explicit matrices $U$ and $V$, not implicit representations as products of reflectors.
(a) Explain what it would mean for this algorithm to be backward stable.
(b) In fact, for a simple reason, this algorithm cannot be backward stable. Explain.
(c) Fortunately, the standard algorithms for computing the SVD are stable. Ex-
plain what stability means for such an algorithm.

Exercise 2 (4 points)
The idea of this exercise is to carry out an experiment analogous to the one de-
scribed in the lecture (householder_stability), but for the SVD instead of QR
factorization.
(a) Write a MATLAB program that constructs a $50 \times 50$ matrix $A=U*S*V^*$, where
$U$ and $V$ are random orthogonal matrices (i.e. the $Q$ of a QR-decomposition of a
random matrix) and $S$ is a diagonal matrix whose diagonal entries are random
uniformly distributed numbers in $[0, 1]$, sorted into nonincreasing order. Have
your program compute $[U2, S2, V2]=\text{svd}(A)$ and the norms of $U-U2$, $V-V2$, $S-S2$
and $A-U2*S2*V2^*$. Do this for five matrices $A$ and comment on the results.
(Hint: Plots of $\text{diag}(U2'*U2)$ and $\text{diag}(V2'*V2)$ may be informative).
(b) Fix the signs in your computed SVD so that the difficulties of (a) go away.
Run the program again for five random matrices and comment on the various
norms. Do they have a connection with $\text{cond}(A)$?
(c) Replace the diagonal entries of $S$ by their sixth powers and repeat (b). Do you
see significant differences between the results of this exercise and those of the
experiment for QR factorization?

– Continues on next page –
Exercise 3 (6 points)
A matrix \( A \in \mathbb{R}^{n \times n} \) has bandwidth \( p \) if \( a_{ij} = 0 \) whenever \( j < i - p \) or \( j > i + p \).
Let \( A \in \mathbb{R}^{n \times n} \) have bandwidth \( p \) and let \( A = L \cdot U \) be computed using Gaussian Elimination without pivoting.
(a) Show that \( L \) and \( U \) have bandwidth \( p \).
(b) Show that the required number of floating point operations is \( O(p^2 n) \).
(c) What can be said about the bandwidths of \( L \) and \( U \) if column pivoting was used during the elimination process?

Exercise 4 (5 points)
Consider the matrix \( A \) from the file sheet11ex4.mat.
(a) Use the MATLAB function \( \text{spy} \) to produce a plot of the nonzero pattern of \( A \).
(b) Use MATLAB to compute a sparse LU factorization \( P_s A Q_s = L_s \cdot U_s \) of \( A \) using \([L_s,U_s,P_s,Q_s] = \text{lu}(A)\). Give the time \( t_s \) needed for this computation.
What algorithm is used by MATLAB for this sparse LU factorization?
(c) Use MATLAB to compute a full LU factorization \( P_f A = L_f \cdot U_f \) of \( A \) using \([L_f,U_f,P_f] = \text{lu}(\text{full}(A))\). Give the time \( t_f \) needed for this computation.
What algorithm is used by MATLAB for this full LU factorization?
(d) Compute the number of nonzeros in \( A, L_s, U_s, L_f, U_f \).
(e) Produce \( \text{spy} \)-plots of \( L_s \) and \( L_f \).
Hints: Use the MATLAB functions \( \text{tic} \) and \( \text{toc} \) to measure the elapsed times. Use the function \( \text{nnz} \) to compute the number of nonzeros in a matrix.
Note: For full credit you have to hand in printouts of your plots. Also write down (or print) the timings and nonzero numbers computed in this exercise. You do not need to hand in your MATLAB code.