

CS 621: Take-Home Midterm

Due: Monday, October 25, 2004 (In Lecture or 4130 Upson by 4pm)

The only discussion allowed is with cv@cs.cornell.edu. Submit a hardcopy of each (fully commented) solution function in 4130 Upson by 4pm on Friday October 22. Include "derivation notes" that may be useful, especially if you couldn't fully solve the problem. Electronic submissions of the four required MATLAB functions are also required. Do this by sending a single email to carlam@cam.cornell.edu with the four files as attachments. Clarifications (if necessary) will be posted on the course website <http://www.cs.cornell.edu/courses/cs621/2004fa/>.

1. This problem is about minimizing $\|AXB^T - C\|_F$ where $A, B \in \mathbb{R}^{m \times n}$ are possibly rank deficient and $C \in \mathbb{R}^{m \times m}$. QR with column pivoting is to be used and in that setting if

$$Q^T F \Pi = R \quad F \in \mathbb{R}^{m \times n}, n \leq m$$

we say that F has δ -rank p if p is the largest index such that $|r_{pp}| > \delta|r_{11}|$. Complete the following MATLAB function so that it performs as specified:

```
function [X,rA,rB] = MatrixLS(A,B,C,delta)
% A and B are m-by-n matrices (m>=n) and C is m-by-m.
% delta > 0
% X is n-by-n, minimizes norm(A*X*B' - C,'fro'), and has at most rA*rB nonzero
% components where rA and rB are the delta-ranks of A and B respectively.
```

Use the MATLAB qr function. Your implementation should be flop-efficient and vectorized and this includes operations that involve permutation matrices.

2. This problem is about solving diagonally dominant linear systems of the form

$$\begin{bmatrix} D & 0 & 0 & E \\ 0 & D & 0 & E \\ 0 & 0 & D & E \\ F^T & F^T & F^T & D \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

where $D, E, F \in \mathbb{R}^{n \times n}$ and $x_i, b_i \in \mathbb{R}^n$. Denote the matrix of coefficients by $A(D, E, F, k)$ where k specifies the block dimension. (In the above example, $k = 4$.) Complete the following MATLAB function so that it performs as specified:

```
function x = BlockDD(D,E,F,k,b,j)
% D, E, and F are n-by-n matrices and k > 2 is a positive integer.
% Assume that M = A(D,E,F,k) is diagonally dominant so that it has a stable LU factorization.
% b is a column nk-by-1 vector and j is a positive integer.
% x is a column nk-by-1 vector that solves (M^j)x = b.
```

Make effective use of the MATLAB lu function.

3. This problem is about perturbing the r_i in the symmetric positive definite Toeplitz matrix

$$T(r) = \begin{bmatrix} 1 & r_1 & r_2 & r_3 \\ r_1 & 1 & r_1 & r_2 \\ r_2 & r_1 & 1 & r_1 \\ r_3 & r_2 & r_1 & 1 \end{bmatrix}$$

so that it becomes indefinite. Complete the following MATLAB function so that it performs as specified:

```
function q = SingularToeplitz(r)
% Assume that r is a column n-1 vector and that the symmetric Toeplitz matrix
% T(r) is positive definite.
% q is a column n-1 vector with the property that if p = r(1:j) + q(j)*I(:,j), then
% T(p) is singular
```

You'll want to study Algorithm 4.7.1 in GVL.

4. This problem is about perturbing a matrix that has a “safe” LU factorization so that it has a “borderline safe” LU factorization. A matrix $A \in \mathbb{R}^{n \times n}$ has a *safe* LU factorization if $A = LU$ and $|\ell_{ij}| < 1$ for all $j < i$. A matrix $A \in \mathbb{R}^{n \times n}$ has a *borderline safe* LU factorization if $A = LU$, $|\ell_{ij}| \leq 1$ for all $j < i$ with equality in at least one subdiagonal component.

Now assume $k \leq n - 1$ and consider this blocking of A 's safe LU factorization:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}$$

where $A_{11}, L_{11}, U_{11} \in \mathbb{R}^{(k-1) \times (k-1)}$, $A_{22}, L_{22}, U_{22} \in \mathbb{R}^{(n-k+1) \times (n-k+1)}$. Note that

$$L_{22}U_{22} = A_{22} - A_{21}A_{11}^{-1}A_{12} \equiv C$$

and so C must have a safe LU factorization. In particular, $L(k:n, k) = C(2:n-k+1, 1)/C(1, 1)$ has components that are all less than one in absolute value. For $i=1:n$, consider how you might (minimally) change a_{ik} to $a_{ik} + t_{ik}$ so that this is not the case. Complete the following function so that it performs as specified:

```
function T = BorderLine(A)
% A is an n-by-n matrix with a safe LU factorization
% T is an n-by-(n-1) matrix with the property that T(i,k) is the smallest tau (in absolute value)
% with the property that if A1 = A + tau*ei*ek' (ei = I(:,i), ek = I(:,k))
% and C = A1(k:n,k:n) - A1(k:n,1:k-1)*inv(A1(1:k-1,1:k-1))*A1(1:k-1,k:n) then
% C(2:n-k+1,1)/C(1,1) has a component equal to one in absolute value.
```