MIT 18.335, Fall 2006: Homework 2

Due October 5

1. Trefethen/Bau 6.1

2. Trefethen/Bau 6.2

3. Trefethen/Bau 7.3

4. Trefethen/Bau 7.4

5. Show how the inner loops in classical Gram-Schmidt

> **for** $i = 1$ **to** $j - 1$
> $r_{ij} = q_i^* a_j$
> $v_j = v_j - r_{ij} q_i$

and in modified Gram-Schmidt

> **for** $j = i + 1$ **to** n
> $r_{ij} = q_i^* v_j$
> $v_j = v_j - r_{ij} q_i$

can be written as matrix multiplications (without the for-loops). Implement these changes in the MATLAB codes `clgs.m` and `mgs.m` (on the next page and on the webpage). Can the inner loop in the simpler modified Gram-Schmidt (replacing a_j in classical Gram-Schmidt with v_j) be rewritten in the same way? Explain.

6. **(a)** Suppose $A \in \mathbb{C}^{m \times n}$ has *lower bandwidth* p and *upper bandwidth* q, that is, $a_{ij} = 0$ when $i > j + p$ or $j > i + q$. Find the sparsity pattern (the location of the zeros and nonzeros) in the matrices Q,R of the QR factorization of A.

(b) Adapt the Householder QR Factorization (Trefethen/Bau Algorithm 10.1) to efficiently handle the case when A has lower bandwidth p and upper bandwidth q, and determine the operation count for your modified algorithm.

(c) Implement your modifications in the MATLAB function `house.m` (on the next page and on the webpage, see Trefethen/Bau, problem 10.2 for more details). Also modify `formQ.m` to take advantage of the sparsity pattern.

For simplicity you can use the same full matrix format as in the original codes (instead of more efficient storage formats). Do not worry about the actual performance of the codes, just think in terms of the flop count. You can verify your codes by comparing with the original codes (or with MATLAB's `qr`). A random matrix with lower bandwidth p and upper bandwidth q can be created in MATLAB by

```
A=triu(tril(randn(m,n),q),-p);
```

7. Compute QR factorizations of the matrix

$$A = \begin{bmatrix} 1 & 0 & 2 \\ 1 & -2 & 4 \\ 2 & 1 & 3 \end{bmatrix}$$

in MATLAB using the functions `mgs.m` and `house.m`/`formQ.m`. Verify that the factorizations satisfy $A = QR$ and that Q and R have the right properties. Explain the results.

8. Trefethen/Bau 11.3

MATLAB Codes (also on the webpage)

```
function [Q,R]=clgs(A)

[m,n]=size(A);
Q=zeros(m,n);
R=zeros(n,n);
for j=1:n
  v=A(:,j);
  for i=1:j-1
    R(i,j)=Q(:,i)'*A(:,j);
    v=v-R(i,j)*Q(:,i);
  end
  R(j,j)=norm(v);
  Q(:,j)=v/R(j,j);
end
```

```
function [Q,R]=mgs(A)

[m,n]=size(A);
V=A;
Q=zeros(m,n);
R=zeros(n,n);
for i=1:n
  R(i,i)=norm(V(:,i));
  Q(:,i)=V(:,i)/R(i,i);
  for j=i+1:n
    R(i,j)=Q(:,i)'*V(:,j);
    V(:,j)=V(:,j)-R(i,j)*Q(:,i);
  end
end
```

```
function [W,R]=house(A)

[m,n]=size(A);
W=zeros(m,n);
for k=1:n
  v=A(k:m,k);
  v(1)=v(1)+(2*(v(1)>=0)-1)*norm(v);
  v=v/norm(v);
  W(k:m,k)=v;
  A(k:m,k:n)=A(k:m,k:n)-2*v*(v'*A(k:m,k:n));
end
R=triu(A(1:n,1:n));
```

```
function Q=formQ(W)

[m,n]=size(W);
Q=eye(m);
for k=n:-1:1
  Q(k:m,:)=Q(k:m,:)-2*W(k:m,k)*(W(k:m,k)'*Q(k:m,:));
end
```