MTH 451-551, Fall 2017, Assignment 3. Each problem is worth 10 points.

Instructions: *Please write neatly. A summary of calculations in 1) will suffice. Underline the major steps and calculations rather than engage in reporting tedious arithmetic. Other instructions for code etc as in Assignment 2.*

Extra credit is turned in on separate paper.

Consider the matrix $A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & r \\ 0 & r & 0 \\ r & 0 & 0 \end{bmatrix}$ in which $r \in \mathbb{R}$ is some parameter.

(Do not turn in) For an exercise, set $r = 1$, and find the orthogonal basis for $Col(A)$ using the classical Gram-Schmidt algorithm and the modified Gram-Schmidt algorithm (call it `cGS` and `mGS`, respectively). Document your steps and write out the QR decomposition of A. You can check your answer using the code in Pbm 2.

1. Now set $r = \epsilon$ in A, with ϵ so small that $\epsilon^2 \approx \epsilon_{machine}$. (This means that, e.g., (*) when computing $1 + \epsilon^2$ in MATLAB you will get 1. However, when computing $\sqrt{2\epsilon^2}$ you should get $\sqrt{2}|\epsilon|$.)

Apply `cGS` and `mGS` to A by hand calculation, "pretending" you are a computer (i.e., that (*) applies). Your results should show the loss of orthogonality in Q when calculated with `cGS`, but which does not arise in `mGS`. (In particular, check if $q_2 \cdot q_3 = 0$.)

Hint: the loss of precision of `cGS` is due to the strong dependence of `cGS` on the accuracy of the many many steps; `mGS` is less sensitive (more stable). In particular, watch out for r_{23}.

2. (Computational). The (very naive) code below realizes `cGS` on the matrix A from problem 1 and shows the loss of orthogonality.

```
>> r=1e6*eps; a=[1 1 1; 0 0 r; 0 r 0; r 0 0];
>> a1=a(:,1);a2=a(:,2);a3=a(:,3);
>> r11=norm(a1);q1=a1/r11
>> r12=a2'*q1;q2=a2-r12*q1
>> r22=norm(q2);q2=q2/r22
>> r13=a3'*q1;r23=a3'*q2;q3=a3-r13*q1-r23*q2
>> r33=norm(q3);q3=q3/r33
>> q3'*q2
```

a) (451-551) Modify the code above to execute `mGS` (show the code). Comment on any difference with respect to the results obtained with `cGS`.

(551 students) should write a proper loop such as that in Algorithm 7.1 on p51 in textbook.

b) Apply QR decomposition to the matrix from (1) in MATLAB using the `qr` function. Do you see any loss of orthogonality, i.e., is $Q' * Q = I$?? Is $QR = A$ to the desired accuracy? Are you satisfied with the accuracy and orthogonality? If yes, that is because the QR algorithm in MATLAB uses a numerically stable Householder procedure rather than `cGS` or `mGS`.