Least Squares with Gram-Schmidt

Recall the Modified Gram-Schmidt QR factorization:

$$A = QR, \qquad \text{where}$$

in exact arithmetic $Q \in \mathbb{R}^{m \times n}$ satisfies $Q^t Q = I$ and $R \in \mathbb{R}^{n \times n}$ is upper triangular with condition number $\kappa(R) = \kappa(A)$. The cost is $2mn^2 + \mathrm{O}(mn)$ flops. If A is overwritten by Q, then only $\frac{1}{2}n^2 + \mathrm{O}(n)$ extra words of memory are required. If $\tilde{Q}$ and $\tilde{R}$ are the computed versions of Q and R, then there exists $\delta A \in \mathbb{R}^{m \times n}$ with $A + \delta A = \tilde{Q}\tilde{R}$, where $\|\delta A\| = \|A\|\mathrm{O}(\mu)$, and $\|\tilde{Q}^t\tilde{Q} - I\| = \kappa(A)\mathrm{O}(\mu)$.

Now to solve the least squares problem (LS) $\min_x \|Ax - b\|_2$ we can use back substitution to solve $Rx = Q^t b$ (to see this substitute $A = QR$ into the normal equations: $A^t A x = A^t b \Rightarrow R^t Q^t Q R x = R^t Q^t b \Rightarrow Q^t Q R x = Q^t b$). Notice that when we write $Rx = Q^t b$ we are assuming $Q^t Q = I$. This is true in exact arithmetic, but the result above says that in finite precision, the orthogonality of Q depends on $\kappa(A)$. Unfortunately, this – combined with the conditioning of R – gives a $[\kappa(A)]^2$ factor in the backward error for x. Here we will show how to avoid this to get a backward error result for MGS which is equivalent to that of the Householder QR applied to (LS).

Let the MGS QR factorization of $[A, b]$ be written as

$$\begin{bmatrix} A & b \end{bmatrix} = \begin{bmatrix} Q & q \end{bmatrix} \begin{bmatrix} R & r \\ 0 & \rho \end{bmatrix},$$

where $q^t Q = 0$ (q is just q_{n+1}). Note that $A = QR$ and $b = Qr + \rho q$. Then $Q^t b = Q^t(Qr + \rho q) = r$, and (LS) is solved by backward substitution: $Rx = r$. Just for fun, show (both algebraically and geometrically) that $\rho = \min_x \|Ax - b\|_2$.

If you already had the MGS factorization $A = QR$ before b arrived, no worries. The partitioning above is just a nice way to package the algebra of one more MGS step:

```
w = b
for i=1:n,
    r(i) = Q(:,i)'*w
    w = w - r(i)*Q(:,i)
end
rho = norm(w);  q = w/rho;
```

Now do you really think that this removes the $\kappa(A)$ factor which came from the orthogonality errors in Q? Why should it? I applaud you for your skepticism. The answer lies in the (substantial) differences in behavior between MGS and CGS. Explicitly computing $Q^t b$ as $r_c = Q^t * b$ is the CGS way, but MGS (the loop above) adapts to the errors made in each inner product, giving an r which has, (to the extent that it can), “accounted for” nonorthogonality in the columns of Q.