Stability of Householder Triangularization — (1/25)
Outline

1 Reference
   - Floating Point Axioms
   - Stability Definitions
   - Accuracy
   - Householder QR

2 Stability of Algorithms
   - Householder Triangularization: Numerical Experiment
   - Householder Triangularization: Backward Stability

3 Solving $A\bar{x} = \bar{b}$
   - Theorem — Householder-Triangularization + Back-Substitution
   - Three Major Holes to Patch...
Axiom (Floating Point Representation)

\[ \forall x \in \mathbb{R}, \text{ there exists } \epsilon \text{ with } |\epsilon| \leq \epsilon_{\text{mach}}, \text{ such that } \text{fl}(x) = x(1 + \epsilon). \]

Axiom (The Fundamental Axiom of Floating Point Arithmetic)

For all \( x, y \in \mathbb{F} \) (where \( \mathbb{F} \) is the set of floating point numbers), there exists \( \epsilon \) with \( |\epsilon| \leq \epsilon_{\text{mach}} \), such that

\[ x \oplus y = (x + y)(1 + \epsilon), \quad x \odot y = (x \times y)(1 + \epsilon), \quad x \oslash y = (x/y)(1 + \epsilon) \]
Definition (Stable Algorithm)

We say that \( \tilde{f} \) is a **stable algorithm** if \( \forall \bar{x} \in X \)

\[
\frac{\| \tilde{f}(\bar{x}) - f(\tilde{\bar{x}}) \|}{\| f(\tilde{\bar{x}}) \|} = O(\epsilon_{\text{mach}}),
\]

for some \( \tilde{x} \) with

\[
\frac{\| \tilde{x} - \bar{x} \|}{\| \bar{x} \|} = O(\epsilon_{\text{mach}}).
\]

“A stable algorithm gives approximately the right answer, to approximately the right question.”
Definition (Backward Stable Algorithm)

An algorithm \( \tilde{f} \) is **backward stable** if \( \forall \bar{x} \in X \)

\[
\tilde{f}(\bar{x}) = f(\tilde{x}),
\]

for some \( \tilde{x} \) with

\[
\frac{\|\tilde{x} - \bar{x}\|}{\|\bar{x}\|} = O(\epsilon_{\text{mach}}).
\]

“A backward stable algorithm gives exactly the right answer, to approximately the right question.”

Jump to: accuracy theorem.
Definition (Accuracy)

We say that the algorithm \( \tilde{f} \) is **accurate** if \( \forall \bar{x} \in X \)

\[
\frac{\| \tilde{f}(\bar{x}) - f(\bar{x}) \|}{\| f(\bar{x}) \|} = O(\epsilon_{\text{mach}}).
\]

This is what we want to do — write algorithms that **accurately** solve problems!

Last time, we finally tied the inherent difficulty of the problem, the **conditioning**, and the quality of the algorithm, the **stability** together in a theorem —
Theorem (Computational Accuracy)

Suppose a backward stable algorithm is applied to solve a problem $f : X \rightarrow Y$ with condition number $\kappa$ in a floating point environment satisfying the floating point representation axiom, and the fundamental axiom of floating point arithmetic.

Then the relative errors satisfy

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\kappa(x)\epsilon_{mach}).$$

Recall: The definition of the relative condition number

$$\kappa(\bar{x}) = \sup_{\delta \bar{x}} \left[ \frac{\|\delta f\|}{\|f(\bar{x})\|} / \frac{\|\delta \bar{x}\|}{\|\bar{x}\|} \right]$$

as the ratio of the relative (infinitesimal) change in $f$ induced by an infinitesimal change in $\bar{x}$.
Algorithm (Householder QR-Factorization)

for $k = 1:n$

\[
\tilde{x} = A(k:m,k) \\
\tilde{v}_k = \text{sign}(x_1)\|\tilde{x}\|_2 e_1 + \tilde{x} \\
\tilde{v}_k = \tilde{v}_k / \|\tilde{v}_k\|_2 \\
A(k:m,k:n) = A(k:m,k:n) - 2\tilde{v}_k (\tilde{v}_k^* A(k:m,k:n)) \\
\tilde{b}(k:m) = \tilde{b}(k:m) - 2\tilde{v}_k (\tilde{v}_k^* \tilde{b}(k:m)) \quad \text{% Compute } Q^*\tilde{b}
\]

endfor

$A(k:m,k)$ Denotes the $k$th thru $m$th rows, in the $k$th column of $A$ — a vector quantity.

$A(k:m,k:n)$ Denotes the $k$th thru $m$th rows, in the $k$th thru $n$th columns of $A$ — a matrix quantity.
With our new toolbox in hand, we re-visit some of the algorithms previously discussed. This second look will reveal, in a more rigorous way, why the algorithms perform the way they do...

The **Householder Triangularization** method of computing the QR-factorization is a backward stable (HT-QR for short).

First, we look at some numerical experiments showcasing this; and then we combine HT-QR with other backward stable algorithmic fragments to build a stable solver for our fundamental problem

\[ A\tilde{x} = \tilde{b}. \]
We generate a matrix $A$ with known QR-factorization, and compute the Householder QR-factorization using Matlab:

$$R = \text{triu}(\text{randn}(64));$$
$$[Q, X] = \text{qr}(\text{randn}(64));$$
$$A = Q \cdot R;$$
$$[Q2, R2] = \text{qr}(A);$$
It turns out that $Q_2$ and $R_2$ are quite far from $Q$ and $R$:

\[
\frac{\text{norm}(Q_2 - Q)}{\text{norm}(Q)} = 0.003427 \\
\frac{\text{norm}(R_2 - R)}{\text{norm}(R)} = 0.000440
\]

It seems like a disaster have occurred.
It turns out that $Q_2$ and $R_2$ are quite far from $Q$ and $R$:

$$\frac{\text{norm}(Q_2-Q)}{\text{norm}(Q)} = 0.003427$$
$$\frac{\text{norm}(R_2-R)}{\text{norm}(R)} = 0.000440$$

It seems like a disaster have occurred, but

$$\frac{\text{norm}(A-Q_2*R_2)}{\text{norm}(A)} = 1.032309e-15$$
It turns out that $Q_2$ and $R_2$ are quite far from $Q$ and $R$:

$$\frac{\text{norm}(Q_2-Q)}{\text{norm}(Q)} = 0.003427$$
$$\frac{\text{norm}(R_2-R)}{\text{norm}(R)} = 0.000440$$

It seems like a disaster have occurred, but

$$\frac{\text{norm}(A-Q_2*R_2)}{\text{norm}(A)} = 1.032309e-15$$

Now consider $Q_3$ and $R_3$

$$Q_3 = Q + 1e-4*\text{randn}(64)$$
$$R_3 = R + 1e-4*\text{randn}(64)$$

$$\frac{\text{norm}(Q_3-Q)}{\text{norm}(Q)} = 0.001595$$
$$\frac{\text{norm}(R_3-R)}{\text{norm}(R)} = 0.000129$$
$$\frac{\text{norm}(A-Q_3*R_3)}{\text{norm}(A)} = 1.065451e-03$$
The Moral of the Story

The errors in $Q_2$ and $R_2$ are known as **forward errors**. Large forward errors are the result of an ill-conditioned problem and/or an unstable algorithm. — In our example it is the former

$$\kappa(A) = \text{cond}(A) = 2.0223e+16.$$  

The error in the result of the matrix product $Q_2R_2$ is known as the **backward error**, or **residual**. The fact that the backward error is small **suggests** that Householder Triangularization is backward stable.
The Moral of the Story

The errors in $Q_2$ and $R_2$ are known as **forward errors**. Large forward errors are the result of an ill-conditioned problem and/or an unstable algorithm. — In our example it is the former

$$\kappa(A) = \text{cond}(A) = 2.0223e+16.$$ 

The error in the result of the matrix product $Q_2 R_2$ is known as the **backward error**, or **residual**.

The fact that the backward error is small suggests that Householder Triangularization is backward stable.

**Note:** Due to the specific way the Householder reflections are performed, the algorithm above may have to be run a couple of times in order to produce (similar) results. A relative error in $Q_2$ of size $\sim 2$ indicates that the initial random $Q$ and $R$ could not possibly have come from a HT-QR algorithm (due to “sign-flips.”)
It turns out that \( \tilde{HT}-QR \) is backward stable for all matrices \( A \) in any floating-point environment satisfying the floating point axioms.

The formal result takes the form

\[ \tilde{Q} \tilde{R} = A + \delta A, \quad \delta A \text{ “small,”} \]

where \( \tilde{R} \) is the upper triangular matrix constructed by the \( \tilde{HT}-QR \) algorithm.

Since the \( \tilde{HT}-QR \) algorithm does not explicitly compute \( \tilde{Q} \) (in the “fast mode,”) we must define what we mean by \( \tilde{Q} \).

Let \( \tilde{Q}_k \) denote the \textit{exactly unitary} reflector defined by the floating point vector \( \tilde{v}_k \)

\[ \tilde{Q}_k = I - 2 \frac{\tilde{v}_k \tilde{v}_k^*}{\tilde{v}_k^* \tilde{v}_k}. \]
Now, we define $\tilde{Q}$ to be the exactly unitary matrix

$$\tilde{Q} = \tilde{Q}_1 \tilde{Q}_2 \cdots \tilde{Q}_n,$$

this matrix will take the place of the computed $Q$ in our discussion.

This approach is natural since in general the matrix $Q$ is not formed explicitly, but rather used implicitly to get the action $Q^*\tilde{b}$.

With these definitions, we are ready to state the theorem...
Theorem (Backward Stability of Householder QR)

Let the QR-factorization $A = QR$ of a matrix $A \in \mathbb{C}^{m \times n}$ be computed by Householder triangularization in a floating-point environment satisfying the floating-point axioms, and let the computed factors $\tilde{Q}$ and $\tilde{R}$ be as discussed on the previous two slides. Then we have

$$\tilde{Q}\tilde{R} = A + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{mach})$$

for some $\delta A \in \mathbb{C}^{m \times n}$.

The full proof can be found in: —
Computing the QR-factorization is not an end in itself. Usually it is one of the first steps in trying to solve a system of linear equations, a least squares problem, or an eigenvalue problem.
Computing the QR-factorization is not an end in itself. Usually it is one of the first steps in trying to solve a system of linear equations, a least squares problem, or an eigenvalue problem.

At this point we know that $HT-QR$ is backward stable, but is that enough?!? As we have seen, the individual factors $Q$ and $R$ may carry large forward errors.
Computing the QR-factorization is not an end in itself. Usually it is one of the first steps in trying to solve a system of linear equations, a least squares problem, or an eigenvalue problem.

At this point we know that HT-QR is backward stable, but is that enough?!? As we have seen, the individual factors $Q$ and $R$ may carry large forward errors.

The **good news** is that accuracy of the product $QR$ is sufficient for most purposes.

We consider the following algorithm of $A\bar{x} = \bar{b}$

1. $QR = A$ Compute the QR-factorization by HT-QR
2. $\bar{y} = Q^*\bar{b}$ Construct $Q^*\bar{b}$ by HT-QR
3. $\bar{x} = R^{-1}\bar{y}$ Solve by back substitution
It turns out that this algorithm is backward stable. The three steps are backward stable, for now we state these results without proof, and then combine them to form the larger result.

We have already expressed the backward stability of HT-QR in a previous theorem.
It turns out that this algorithm is backward stable. The three steps are backward stable, for now we state these results without proof, and then combine them to form the larger result.

We have already expressed the backward stability of HT–QR in a previous theorem.

The second step computes $\tilde{Q}^*\tilde{b}$, due to floating-point errors, the result $\tilde{y}$ is not equal to $\tilde{y} = \tilde{Q}^*\tilde{b}$, but the operation is backward stable

$$(\tilde{Q} + \delta Q)\tilde{y} = \tilde{b}, \quad \|\delta Q\| = O(\epsilon_{mach}).$$

The solution $\tilde{x}$ of the back substitution in the third step satisfies

$$(\tilde{R} + \delta R)\tilde{x} = \tilde{y}, \quad \frac{\|\delta R\|}{\|\tilde{R}\|} = O(\epsilon_{mach}).$$
With these unproven (for now) building blocks, we are ready to state and prove the following theorem

**Theorem**

The three step algorithm described above for solving $A\tilde{x} = \tilde{b}$ is backward stable, satisfying

$$(A + \Delta A)\tilde{x} = \tilde{b}, \quad \frac{\|\Delta A\|}{\|A\|} = O(\epsilon_{mach}),$$

for some $\Delta A \in \mathbb{C}^{m \times m}$. 
Proof: From step #2 and step #3 we have

\[(\tilde{Q} + \delta Q)\tilde{y} = \tilde{b}, \quad \text{and} \quad (\tilde{R} + \delta R)\tilde{x} = \tilde{y},\]

combining the two gives

\[\tilde{b} = (\tilde{Q} + \delta Q)(\tilde{R} + \delta R)\tilde{x} = \left[\tilde{Q}\tilde{R} + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R)\right]\tilde{x}.\]
Solving $A\tilde{x} = \tilde{b}$

**Proof:** From step #2 and step #3 we have

$$(\tilde{Q} + \delta Q)(\tilde{y}) = \tilde{b}, \quad \text{and} \quad (\tilde{R} + \delta R)\tilde{x} = \tilde{y},$$

combining the two gives

$$\tilde{b} = (\tilde{Q} + \delta Q)(\tilde{R} + \delta R)\tilde{x} = \left[ \tilde{Q}\tilde{R} + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R) \right] \tilde{x}.$$ 

Now, using the result for step #1

$$\tilde{Q}\tilde{R} = A + \delta A$$

we get

$$\tilde{b} = \left[ A + \delta A + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R) \right] \tilde{x}.$$
Next, we must show that the perturbation

$$\Delta A = \delta A + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R)$$

is small relative to $A$. 

Peter Blomgren, ⟨blomgren.peter@gmail.com⟩
Next, we must show that the perturbation
\[ \Delta A = \delta A + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R) \]
is small relative to \( A \).

Since \( \tilde{Q}\tilde{R} = A + \delta A \), and \( \tilde{Q} \) is unitary we have
\[ \frac{\|\tilde{R}\|}{\|A\|} \leq \|\tilde{Q}^*\| \frac{\|A + \delta A\|}{\|A\|} = \mathcal{O}(1), \quad \epsilon_{\text{mach}} \to 0. \]
Next, we must show that the perturbation

$$\Delta A = \delta A + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R)$$

is small relative to $A$.

Since $\tilde{Q}\tilde{R} = A + \delta A$, and $\tilde{Q}$ is unitary we have

$$\frac{\|\tilde{R}\|}{\|A\|} \leq \|\tilde{Q}^*\| \frac{\|A + \delta A\|}{\|A\|} = O(1), \quad \epsilon_{\text{mach}} \to 0.$$  

Hence, the relative size of the second term is bounded

$$\frac{\| (\delta Q)\tilde{R} \|}{\| A \|} \leq \| (\delta Q) \| \frac{\| \tilde{R} \|}{\| A \|} = O(\epsilon_{\text{mach}}).$$
Now, consider the third term

$$\Delta A = \delta A + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R)$$

$$\frac{\|\tilde{Q}(\delta R)\|}{\|A\|} \leq \frac{\|\tilde{Q}\|\|\tilde{R}\|}{\|A\|} = \frac{\|\tilde{Q}\|\|(\delta R)\|\|\tilde{R}\|}{\|A\|}. $$
Now, consider the third term

$$\Delta A = \delta A + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R)$$

$$\frac{\|\tilde{Q}(\delta R)\|}{\|A\|} \leq \|\tilde{Q}\| \frac{\|(\delta R)\|}{\|A\|} = \|\tilde{Q}\| \frac{\|(\delta R)\|}{\tilde{R}} \|\tilde{R}\| \frac{\|\tilde{R}\|}{\|A\|}.$$ 

Since

$$\|\tilde{Q}\| = O(1), \quad \frac{\|(\delta R)\|}{\|\tilde{R}\|} = O(\epsilon_{mach}), \quad \text{and} \quad \frac{\|\tilde{R}\|}{\|A\|} = O(1),$$
Now, consider the third term
\[ \Delta A = \delta A + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R) \]

\[
\frac{\|\tilde{Q}(\delta R)\|}{\|A\|} \leq \frac{\|\tilde{Q}\|}{\|	ilde{R}\|} \cdot \frac{\|\tilde{Q}\|}{\|	ilde{R}\|} = \frac{\|\tilde{Q}\|}{\|	ilde{R}\|} \cdot \frac{\|\tilde{R}\|}{\|A\|}.
\]

Since
\[ \|\tilde{Q}\| = O(1), \quad \frac{\|(\delta R)\|}{\|	ilde{R}\|} = O(\epsilon_{\text{mach}}), \quad \text{and} \quad \frac{\|\tilde{R}\|}{\|A\|} = O(1), \]

we have
\[ \frac{\|\tilde{Q}(\delta R)\|}{\|A\|} = O(\epsilon_{\text{mach}}). \]
Finally, the fourth term

\[ \Delta A = \delta A + (\delta Q)\hat{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R) \]

\[ \frac{\|(\delta Q)(\delta R)\|}{\|A\|} \leq \|(\delta Q)\| \frac{\|(\delta R)\|}{\|A\|} \]
Finally, the fourth term

$$\Delta A = \delta A + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R)$$

We know

$$\|\delta Q\| = O(\epsilon_{mach}), \quad \text{and} \quad \frac{\|\delta R\|}{\|A\|} = O(\epsilon_{mach})$$

So,

$$\frac{\|\delta Q\|\|\delta R\|}{\|A\|} = O(\epsilon_{mach}^2)$$
We collect our findings, and note that as required

\[
\frac{\|\Delta A\|}{\|A\|} \leq \frac{\|\delta A\|}{\|A\|} + \frac{\|Q(\delta R)\tilde{R}\|}{\|A\|} + \frac{\|Q(\delta R)\|}{\|A\|} + \frac{\|Q(\delta R)\|}{\|A\|} = O(\epsilon_{\text{mach}}).
\]

This completes the proof. □
We collect our findings, and note that as required

\[
\frac{\|\Delta A\|}{\|A\|} \leq \frac{\|\delta A\|}{\|A\|} + \frac{\|Q(\delta R)\tilde{R}\|}{\|A\|} + \frac{\|\tilde{Q}(\delta R)\|}{\|A\|} + \frac{\|Q(\delta R)(\delta R)\|}{\|A\|} = O(\epsilon_{\text{mach}}).
\]

This completes the proof. □

If we combine this result with the accuracy theorem we showed last time, we get the following result about the accuracy of solutions of \( A\tilde{x} = \tilde{b} \) using the Householder-Triangularization + Back-substitution algorithm:
The solution $\tilde{x}$ computed by the Householder-Triangularization + Back-substitution algorithm satisfies

$$\frac{\|\tilde{x} - \bar{x}\|}{\|\bar{x}\|} = O(\kappa(A)\epsilon_{mach})$$
We have left three major holes in the argument — the statement, without proof, that the individual steps are backward stable.

It is instructive to see at least one such proof from “scratch.” — Next time, we turn our attention to the back-substitution algorithm.

Even though back substitution is one of the easiest problems of numerical linear algebra, the stability proof is quite lengthy...