Lecture 17 - *Stability of Back Substitution*

OBJECTIVE:
Solving triangular systems of linear equations is particularly easy.
The standard algorithm is successive substitution, called *back substitution* when the system is upper-triangular.
We now show in full detail that this algorithm is backward stable.

◇ TRIANGULAR SYSTEMS

We have seen that solving $\mathbf{Ax} = \mathbf{b}$ can be reduced to solving the upper-triangular system $\mathbf{Rx} = \mathbf{y}$ via $\mathbf{QR}$ factorization.

Similar upper- and lower-triangular systems arise in various other places in numerical linear algebra, including Gaussian elimination and Cholesky factorization.

Such systems are very easy to solve by a process of successive substitution, called *forward substitution* if the system is lower-triangular, and *back substitution* if it is upper-triangular.

Of course, both processes are identical mathematically!

But, for concreteness, we describe only back substitution.

Suppose we wish to solve $\mathbf{Rx} = \mathbf{b}$

$$\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ & r_{22} & \cdots & r_{2m} \\ & & \ddots & \vdots \\ & & & r_{mm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Here, $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{R} \in \mathbb{R}^{m \times m}$ with $r_{ii} \neq 0$ are given, and $\mathbf{x} \in \mathbb{R}^m$ is unknown.

In back-substitution, we solve for each x_i in succession, starting with x_m:

ALGORITHM 17.1: BACK SUBSTITUTION

$$\begin{aligned} x_m &= b_m / r_{mm} \\ x_{m-1} &= (b_{m-1} - r_{m-1,m} x_m) / r_{m-1,m-1} \\ &\vdots \\ x_j &= \left(b_j - \sum_{k=m}^{j+1} r_{jk} x_k \right) / r_{jj} \end{aligned}$$

Flop count: at step j, there are:
$m - j$ additions/subtractions
$m - j + 1$ multiplications/divisions

So total flops

$$= \sum_{j=1}^{m} 2(m-j) + 1 \sim 2 \left(m^2 - \frac{m(m+1)}{2} \right) \sim m^2$$

◇ BACKWARD STABILITY THEOREM

In the last lecture, we claimed that backward substitution was backward stable. We now justify that claim.

Theorem 1. *Let Algorithm 17.1 be used to solve*

$$\mathbf{R}\mathbf{x} = \mathbf{b}$$

Then the algorithm is backward stable in the sense that the computed solution $\tilde{\mathbf{x}} \in \mathbb{R}^m$ *satisfies*

$$(\mathbf{R} + \delta\mathbf{R})\tilde{\mathbf{x}} = \mathbf{b}$$

for some upper-triangular $\delta\mathbf{R} \in \mathbb{R}^{m\times m}$ *satisfying*

$$\frac{\|\delta\mathbf{R}\|}{\|\mathbf{R}\|} = \mathcal{O}(\epsilon_{\text{machine}})$$

Moreover, for each i, j*,*

$$\frac{|\delta r_{ij}|}{|r_{ij}|} \leq m\epsilon_{\text{machine}} + \mathcal{O}(\epsilon^2_{\text{machine}}).$$

We build up a proof in an inductive fashion.

m=1:

Algorithm 17.1 consists of 1 step

$$\begin{aligned} \tilde{x}_1 &= b_1 \oslash r_{11} \\ &= \frac{b_1}{r_{11}}(1+\epsilon_1), \qquad \text{where } |\epsilon_1| \leq \epsilon_{\text{machine}} \end{aligned}$$

However, we want to express this as if it were a perturbation to $\mathbf{R}$. To this end, set

$$\epsilon_1' = \frac{-\epsilon_1}{1+\epsilon_1}$$

so that

$$\tilde{x}_1 = \frac{b_1}{r_{11}(1+\epsilon_1')},$$

where

$$|\epsilon_1'| \leq \epsilon_{\text{machine}} + \mathcal{O}(\epsilon_{\text{machine}}^2)$$

because

$$\begin{aligned}\epsilon_1' &= -\epsilon_1(1+\epsilon_1)^{-1} \\ &= -\epsilon_1(1+\epsilon_1+\ldots) \\ &= -\epsilon_1+\mathcal{O}(\epsilon_1^2)\end{aligned}$$

Now we can see that $\tilde{x}_1$ is the exact solution to the perturbed problem

$$(r_{11}+\delta r_{11})\tilde{x}_1 = b_1,$$

where

$$\delta r_{11} = \epsilon_1' r_{11}$$

so that

$$\frac{|\delta r_{11}|}{|r_{11}|} \leq \epsilon_{\text{machine}} + \mathcal{O}(\epsilon_{\text{machine}}^2).$$

m=2:

In this case we have

$$\begin{bmatrix} r_{11} & r_{12} \\ & r_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

The first step of the $m = 2$ case is the same as the *only* step when $m = 1$:

$$\begin{aligned} \tilde{x}_2 &= b_2 \oslash r_{22} \\ &= \frac{b_2}{r_{22}(1+\epsilon_1)} \end{aligned}$$

for some ϵ_1 with

$$|\epsilon_1| \leq \epsilon_{\text{machine}} + \mathcal{O}(\epsilon_{\text{machine}}^2).$$

The second step is

$$\tilde{x}_1 = (b_1 \ominus (r_{12} \otimes \tilde{x}_2)) \oslash r_{11}$$

To establish backward stability, we must express the errors as perturbations of the entries r_{ij}.

First we can write

$$\tilde{x}_1 = (b_1 \ominus r_{12}\tilde{x}_2(1+\epsilon_2)) \oslash r_{11}, \qquad |\epsilon_2| \leq \epsilon_{\text{machine}}.$$

Now formally applying the axioms of floating-point arithmetic, we can write

$$\begin{aligned} \tilde{x}_1 &= (b_1 - r_{12}\tilde{x}_2(1+\epsilon_2))(1+\epsilon_3) \oslash r_{11} \\ &= \frac{(b_1 - r_{12}\tilde{x}_2(1+\epsilon_2))(1+\epsilon_3)}{r_{11}}(1+\epsilon_4), \end{aligned}$$

where

$$|\epsilon_3|, |\epsilon_4| \leq \epsilon_{\text{machine}}.$$

We can re-write this as

$$\begin{aligned}\tilde{x}_1 &= \frac{b_1 - r_{12}\tilde{x}_2(1+\epsilon_2)}{r_{11}(1+\epsilon_3')(1+\epsilon_4')} \\ &= \frac{b_1 - r_{12}\tilde{x}_2(1+\epsilon_2)}{r_{11}(1+2\epsilon_5)}\end{aligned}$$

where

$$|\epsilon_3'|, |\epsilon_4'| \leq \epsilon_{\text{machine}} + \mathcal{O}(\epsilon_{\text{machine}}^2)$$

and

$$|\epsilon_5| \leq \epsilon_{\text{machine}} + \mathcal{O}(\epsilon_{\text{machine}}^2)$$

We can interpret this as meaning $\tilde{x}_1$ is the exact solution to the problem where $r_{11}, r_{12},$ and r_{22} are perturbed by factors $(1+2\epsilon_5), (1+\epsilon_2),$ and $(1+\epsilon_1)$ respectively.

In other words

$$(\mathbf{R} + \delta\mathbf{R})\tilde{\mathbf{x}} = \mathbf{b}$$

where the entries δr_{ij} of $\delta\mathbf{R}$ satisfy

$$\begin{bmatrix} |\delta r_{11}|/|r_{11}| & |\delta r_{12}|/|r_{12}| \\ & |\delta r_{22}|/|r_{22}| \end{bmatrix} = \begin{bmatrix} 2|\epsilon_5| & |\epsilon_2| \\ & |\epsilon_1| \end{bmatrix}$$
$$\leq \begin{bmatrix} 2 & 1 \\ & 1 \end{bmatrix} \epsilon_{\text{machine}} + \mathcal{O}(\epsilon_{\text{machine}}^2).$$

Note 1. *Inequalities with matrices and vectors are usually interpreted* componentwise*.*

Thus, $\frac{\|\delta\mathbf{R}\|}{\|\mathbf{R}\|} = \mathcal{O}(\epsilon_{\text{machine}})$ and so back substitution for the 2×2 case is backward stable.

m=3:

Again we use the results from the first two cases to derive

$$\begin{aligned}
\tilde{x}_3 &= b_3 \oslash r_{33} \\
&= \frac{b_3}{r_{33}(1+\epsilon_1)} \\
\tilde{x}_2 &= (b_2 \ominus (r_{23} \otimes \tilde{x}_3)) \oslash r_{22} \\
&= \frac{b_2 - r_{23}\tilde{x}_3(1+\epsilon_2)}{r_{22}(1+2\epsilon_3)},
\end{aligned}$$

where

$$\begin{bmatrix} 2|\epsilon_3| & |\epsilon_2| \\ & |\epsilon_1| \end{bmatrix} \leq \begin{bmatrix} 2 & 1 \\ & 1 \end{bmatrix} \epsilon_{\text{machine}} + \mathcal{O}(\epsilon_{\text{machine}}^2).$$

The formula for $\tilde{x}_1$ is

$$\begin{aligned}
\tilde{x}_1 &= [(b_1 \ominus (r_{12} \otimes \tilde{x}_2) \ominus (r_{13} \otimes \tilde{x}_3)] \oslash r_{11} \\
&= [b_1 \ominus r_{12}\tilde{x}_2(1+\epsilon_4) \ominus r_{13}\tilde{x}_3(1+\epsilon_5)] \oslash r_{11} \\
&= \frac{(b_1 - r_{12}\tilde{x}_2(1+\epsilon_4))(1+\epsilon_6)(1+\epsilon_7)}{r_{11}(1+\epsilon'_8)} \\
&\qquad - \frac{r_{13}\tilde{x}_3(1+\epsilon_5)(1+\epsilon_7)}{r_{11}(1+\epsilon'_8)} \\
&= \frac{(b_1 - r_{12}\tilde{x}_2(1+\epsilon_4))(1+\epsilon_6)}{r_{11}(1+\epsilon'_8)(1+\epsilon'_7)} \\
&\qquad - \frac{r_{13}\tilde{x}_3(1+\epsilon_5)}{r_{11}(1+\epsilon'_8)(1+\epsilon'_7)}
\end{aligned}$$

Now if we write

$$\begin{aligned}
r_{13}\tilde{x}_3(1+\epsilon_5) &= \frac{r_{13}\tilde{x}_3(1+\epsilon_5)(1+\epsilon_6)}{1+\epsilon_6} \\
&= r_{13}\tilde{x}_3(1+\epsilon_5)(1+\epsilon'_6)(1+\epsilon_6)
\end{aligned}$$

Now the common factor of $(1+\epsilon_6)$ can be moved to the denominator as $(1+\epsilon_6')$ to obtain

$$\tilde{x}_1 = \frac{b_1 - r_{12}\tilde{x}_2(1+\epsilon_4) - r_{13}(1+\epsilon_5)(1+\epsilon_6')}{r_{11}(1+\epsilon_6')(1+\epsilon_7')(1+\epsilon_8')}$$

So r_{12} has one perturbation of size at most $\epsilon_{\text{machine}}$, and r_{13}, r_{11} have two and three such perturbations respectively.

This allows us to write

$$(\mathbf{R} + \delta\mathbf{R})\tilde{\mathbf{x}} = \mathbf{b}$$

where the entries δr_{ij} of $\delta\mathbf{R}$ satisfy

$$\begin{bmatrix} |\delta r_{11}|/|r_{11}| & |\delta r_{12}|/|r_{12}| & |\delta r_{13}|/|r_{13}| \\ & |\delta r_{22}|/|r_{22}| & |\delta r_{23}|/|r_{23}| \\ & & |\delta r_{33}|/|r_{33}| \end{bmatrix}$$
$$\leq \begin{bmatrix} 3 & 1 & 2 \\ & 2 & 1 \\ & & 1 \end{bmatrix} \epsilon_{\text{machine}} + \mathcal{O}(\epsilon_{\text{machine}}^2).$$

◇ GENERAL m

For $m > 3$, the arguments are similar. The pattern (say for $m = 5$) is as follows

$$\frac{|\delta \mathbf{R}|}{|\mathbf{R}|} \leq \begin{bmatrix} 5 & 1 & 2 & 3 & 4 \\ & 4 & 1 & 2 & 3 \\ & & 3 & 1 & 2 \\ & & & 2 & 1 \\ & & & & 1 \end{bmatrix} \epsilon_{\text{machine}} + \mathcal{O}(\epsilon_{\text{machine}}^2).$$

The terms are obtained as follows:

Each multiplication $r_{jk}\tilde{x}_k$ introduces $\epsilon_{\text{machine}}$ perturbations in the pattern

$$\otimes : \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ & 0 & 1 & 1 & 1 \\ & & 0 & 1 & 1 \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix}$$

The divisions by r_{kk} introduce perturbations in the pattern

$$\oslash : \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

Finally, the subtractions occur in the pattern for $\otimes$.

This adds up to

$$\ominus : \begin{bmatrix} 4 & 0 & 1 & 2 & 3 \\ & 3 & 0 & 1 & 2 \\ & & 2 & 0 & 1 \\ & & & 1 & 0 \\ & & & & 0 \end{bmatrix}$$

Adding them all up gives the result stated earlier.

◇ FINAL REMARKS

1. More than one error bound can be given for a specific algorithm.

   Sometimes results are expressed in terms of perturbations of $\mathbf{b}$.

   However, our analysis is particularly clear.

2. We have derived a *componentwise* backward error bound.

   i.e., each perturbation δr_{ij} is small relative to the quantity being perturbed (r_{ij}) and not just $\|\mathbf{R}\|$.

   If $r_{ij} = 0$ then, $\delta r_{ij} = 0$!
   $\Rightarrow \delta\mathbf{R}$ has the same *sparsity pattern* as $\mathbf{R}$.

   Such componentwise bounds are not always possible. But, they are often more desirable since they provide sharper bounds and algorithms satisfying such bounds are less sensitive to scaling of variables.

3. Normwise bounds have the advantage of being norm-independent, easy to remember, and not as pessimistic as componentwise bounds that must account for growth factors that depend on m and n $\rightarrow$ these are the bounds that people typically remember!