AN ERROR ANALYSIS OF GAUSSIAN ELIMINATION

David F. Gleich

October 24, 2019

These notes were copied from Gene Golub's CME 302 Matrix Computations class while David was a student at Stanford.

Warning We will adopt slightly non-standard notation. The quantity $\mathbf{\Delta}$ is a matrix, as are any symbols prefixed with δ, such as $\delta\boldsymbol{L}$ or $\delta\boldsymbol{U}$. Upper tildes, like $\tilde{\boldsymbol{L}}$ will denote quantities represented on the computer.

Our goal is to show that Gaussian Elimination is a backwards stable algorithm. If we show that Gaussian elimination is backwards stable, then we will show that we can compute $\tilde{\mathbf{x}}$ such that

$$(\boldsymbol{A} + \mathbf{\Delta})\tilde{\mathbf{x}} = \mathbf{b}.$$

In words, we compute the solution to a perturbed system where the perturbation is the matrix $\mathbf{\Delta}$. If so, then, if $\boldsymbol{A}\mathbf{x} = \mathbf{b}$, we have:[1]

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \frac{\rho\kappa(\boldsymbol{A})}{1 - \rho\kappa(\boldsymbol{A})}$$

where $\rho = \|\mathbf{\Delta}\|/\|\boldsymbol{A}\|$.

Thus, if we can show that $\|\mathbf{\Delta}\|$ is small, we will have a nice bound on the error of the solution.

[1] It's a good exercise to stitch this bound together from $\tilde{\mathbf{x}} + \boldsymbol{A}^{-1}\mathbf{\Delta}\tilde{\mathbf{x}} = \mathbf{x}$.

FRAMEWORK

This is a complicated operation. When we solve $\boldsymbol{A}\mathbf{x} = \mathbf{b}$ with Gaussian elimination, we have three steps:

1. Factoring $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$. We will show that we find: $\tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}} = \boldsymbol{A} + \boldsymbol{E}$. We will assume that partial pivoting is used, although, we will assume the permutation is known up front.

2. Next we have to solve $\boldsymbol{L}\mathbf{y} = \mathbf{b}$, or on the computer,

$$(\tilde{\boldsymbol{L}} + \delta\boldsymbol{L})\underbrace{(\mathbf{y} + \delta\mathbf{y})}_{=\tilde{\mathbf{y}}} = \mathbf{b}.$$

   In other words, we again solve a perturbed system exactly.

3. Finally, we have to solve $\boldsymbol{U}\mathbf{x} = \mathbf{y}$. But on the computer, this is now:

$$(\tilde{\boldsymbol{U}} + \delta\boldsymbol{U})(\mathbf{x} + \delta\mathbf{x}) = \tilde{\mathbf{y}} = \mathbf{y} + \delta\mathbf{y}.$$

Together, these results show that

$$\mathbf{b} = (\tilde{\boldsymbol{L}} + \delta\boldsymbol{L})(\tilde{\boldsymbol{U}} + \delta\boldsymbol{U})(\mathbf{x} + \delta\mathbf{x}).$$

But, we have:

$$(\tilde{\boldsymbol{L}} + \delta\boldsymbol{L})(\tilde{\boldsymbol{U}} + \delta\boldsymbol{U}) = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}} + \delta\boldsymbol{L}\tilde{\boldsymbol{U}} + \tilde{\boldsymbol{L}}\delta\boldsymbol{U} + \delta\boldsymbol{L}\delta\boldsymbol{U} = \boldsymbol{A} + \boldsymbol{E} + \delta\boldsymbol{L}\tilde{\boldsymbol{U}} + \tilde{\boldsymbol{L}}\delta\boldsymbol{U} + \delta\boldsymbol{L}\delta\boldsymbol{U}.$$

Thus,

$$(\boldsymbol{A} + \mathbf{\Delta})(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b}$$

with $\mathbf{\Delta} = \boldsymbol{E} + \delta\boldsymbol{L}\tilde{\boldsymbol{U}} + \tilde{\boldsymbol{L}}\delta\boldsymbol{U} + \delta\boldsymbol{L}\delta\boldsymbol{U}$.

Now we go through and find bounds on all of these terms.

ERRORS IN LU

Our goal now is to show that what we compute on the computer is: $\tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}} = \boldsymbol{A} + \boldsymbol{E}$ for some $\boldsymbol{E}$.

We'll show this in two steps. In the first step, we'll just introduce additive errors into each of our operations. In the second step, we'll use the properties of floating point arithmetic to bound those errors.

GAUSSIAN ELIMINATION WITH ERRORS

Suppose we are computing the LU factorization of $\boldsymbol{A}$. We'll represent this as a sequence of changes to the matrix

$$\boldsymbol{A} = \boldsymbol{A}^{(1)} \xrightarrow{\text{zero 1st column}} \boldsymbol{A}^{(2)} \xrightarrow{\text{zero 2st column}} \boldsymbol{A}^{(3)} \to \cdots \to \boldsymbol{A}^{(n-1)}.$$

Thus, $\boldsymbol{A}^{(k)}$ is the matrix after $k-1$ columns have been zeroed. To move to the k+1st step, we compute:

$$A_{ij}^{(k+1)} = A_{ij}^{(k)} - L_{ik}A_{kj}^{(k)}, \quad L_{ik} = \frac{A_{ik}^{(k)}}{A_{kk}^{(k)}}.$$

Let $\boldsymbol{B}^{(k)}$ be the matrix after $k-1$ columns have been zeroed in floating point arithmetic. We have:

$$B_{ij}^{(k+1)} = B_{ij}^{(k)} - \tilde{L}_{ik}B_{kj}^{(k)} + \mu_{i,j}^{(k+1)}, \quad \tilde{L}_{ik} = \frac{B_{ik}^{(k)}}{B_{kk}^{(k)}}(1+\eta_{ik}).$$

In this expression, $\mu_{i,j}^{(k+1)}$ represents the floating point error in computing $B_{ij}^{(k+1)}$ from the intermediate terms. For this expression, note that μ does not need to include the effect from η because we are analyzing this expression with $\tilde{L}$ – the computed quantity, not the exact quantity. For each element B_{ij} there is a maximum k such that we will stop looking at that element in the future.[2] Thus, when we stop looking at an element B_{ij} there are two reasons: 1) it's in the upper triangle and $i \le k$, or 2) it's zero in the lower-triangle with $j < k$.

So we'll divide our analysis into two cases that correspond to these two outcomes. First, suppose we are in the upper-triangle, so $j \ge i$. Then,

$$\begin{aligned}
B_{ij}^{(2)} &= B_{ij}^{(1)} - \tilde{L}_{i,1}B_{1,j}^{(1)} + \mu_{ij}^{(2)} \\
B_{ij}^{(3)} &= B_{ij}^{(2)} - \tilde{L}_{i,2}B_{2,j}^{(2)} + \mu_{ij}^{(3)} \\
&\cdots \\
B_{ij}^{(i)} &= B_{ij}^{(i-1)} - \tilde{L}_{i,i-1}B_{i,j}^{(i-1)} + \mu_{ij}^{(i)}.
\end{aligned}$$

The goal here is a relationship between $B_{ij}^{(1)}$ and $B_{ij}^{(i)}$. If we sum up all of these expressions, we have:

$$\sum_{k=2}^{i} B_{ij}^{(k)} = \sum_{k=1}^{i-1} B_{ij}^{(k)} - \sum_{k=1}^{i-1} \tilde{L}_{i,k}B_{k,j}^{(k)} + \sum_{k=2}^{i} \mu_{ij}^{(k)}.$$

Note that this sum telescopes! In other words, we get massive cancellation of the $\boldsymbol{B}_{ij}^{(k)}$ terms. After all of them are removed, we have:

$$B_{ij}^{(i)} = B_{ij}^{(1)} - \sum_{k=1}^{i-1} \tilde{L}_{i,k}B_{k,j}^{(k)} + \sum_{k=2}^{i} \mu_{ij}^{(k)}.$$

We can rearrange this to show that:

$$B_{ij}^{(1)} + E_{ij} = B_{ij}^{(i)} + \sum_{k=1}^{i-1} \tilde{L}_{i,k}B_{k,j}^{(k)}$$

[2] This is a straightforward observation if you look at LU in exact arithmetic:

$$\boldsymbol{A}^{(k)} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{C} \\ 0 & \boldsymbol{D} \end{bmatrix},$$

where $\boldsymbol{U}$ is $(k-1)$-by-$(k-1)$.

where $E_{ij} = \sum_{k=2}^{i} \mu_{ij}^{(k)}$.

We are half done with showing the error in the LU factorization. At this point, we've shown that the upper-triangular piece of our factorization is correct for a matrix $\boldsymbol{A} + \boldsymbol{E}$, with a precise accounting of where the errors occur. Now, we just have to show that same thing holds in the lower-triangular region.

If $i > j$, then

$$\begin{aligned} B_{ij}^{(2)} &= B_{ij}^{(1)} - \tilde{L}_{i,1} B_{1,j}^{(1)} + \mu_{ij}^{(2)} \\ B_{ij}^{(3)} &= B_{ij}^{(2)} - \tilde{L}_{i,2} B_{2,j}^{(2)} + \mu_{ij}^{(3)} \\ &\cdots \\ B_{ij}^{(j)} &= B_{ij}^{(j-1)} - \tilde{L}_{i,j-1} B_{j-1,j}^{(j-1)} + \mu_{ij}^{(j)}. \end{aligned}$$

This is, of course, the same.[3] But we also have:

$$0 = B_{ij}^{(j)} - \tilde{L}_{i,j} B_{jj}^{(j)} + \mu_{ij}^{(j+1)}$$

because B_{ij} becomes o in the $(j+1)$st step. After a similar cancellation of terms, we get:

$$B_{ij}^{(1)} = 0 + \sum_{k=1}^{j} \tilde{L}_{i,k} B_{k,j}^{(k)} + E_{ij}$$

where $E_{ij} = \sum_{k=2}^{j+1} \mu_{ij}^{(k)}$.

Thus, what we compute on the computer is:

$$\tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}} = \begin{bmatrix} 1 & & & & \\ \tilde{L}_{2,1} & 1 & & & \\ \tilde{L}_{3,1} & \tilde{L}_{3,2} & 1 & & \\ \vdots & \vdots & & \ddots & \\ \tilde{L}_{n,1} & \tilde{L}_{n,2} & \cdots & \cdots & 1 \end{bmatrix} \begin{bmatrix} B_{1,1}^{(1)} & B_{1,2}^{(1)} & B_{1,3}^{(1)} & \cdots & B_{1,n}^{(1)} \\ & B_{2,2}^{(2)} & B_{2,3}^{(2)} & \cdots & B_{2,n}^{(2)} \\ & & B_{3,3}^{(3)} & & \vdots \\ & & & \ddots & \vdots \\ & & & & B_{n,n}^{(n)} \end{bmatrix}.$$

Using our equations that we derived, we can show:

$$\tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}} = \boldsymbol{B}^{(1)} + \boldsymbol{E} = \boldsymbol{A} + \boldsymbol{E}.$$

BOUNDING THE ERRORS

Now, we need to bound each element in $\boldsymbol{E}$. We have:

$$\tilde{L}_{i,k} = \mathrm{fl}\left(B_{ik}^{(k)} / B_{kk}^{(k)}\right) = (B_{ik}^{(k)} / B_{kk}^{(k)})(1 + \eta_{ik})$$

and

$$\mathrm{fl}\left(\tilde{L}_{i,k} B_{k,j}^{(k)}\right) = (\tilde{L}_{i,k} B_{k,j}^{(k)})(1 + \theta_{ij}^{(k)}),$$

so that:

$$B_{ij}^{(k+1)} = \mathrm{fl}\left(B_{ij}^{(k)} - (\tilde{L}_{i,k} B_{k,j}^{(k)})(1 + \theta_{ij}^{(k)})\right) = \left(B_{ij}^{(k)} - (\tilde{L}_{i,k} B_{k,j}^{(k)})(1 + \theta_{ij}^{(k)})\right)(1 + \phi_{ij}^{(k)}).$$

The quantities η, θ, and ϕ all obey $|\cdot| \leq u$, the machine round-off error. By reworking this bound for a while, we get:

$$\mu_{ij}^{(k+1)} = B_{ij}^{(k+1)} \left(\frac{\phi_{ij}^{(k)}}{1 + \phi_{ij}^{(k)}} \right) - \tilde{L}_{i,k} B_{k,j}^{(k)} \theta_{ij}^{(k)}.$$

Using the bound $|\tilde{L}_{ij}| \leq 1$ from using partial pivoting, we find:

$$|\mu_{ij}^{(k+1)}| \leq |B_{ij}^{(k+1)}| \frac{u}{1-u} + |B_{kj}^{(k)}| u.$$

[3] I think there might be an index mistake in here, be wary.

We are getting close to a bound. We now need to understand how big elements in $\boldsymbol{B}$ can get! Here, we'll use exact computation again. Let $|A_{ij}| \leq a$ for all i, j. Using

$$A_{ij}^{(k+1)} = A_{ij}^{(k)} - L_{ik}A_{kj}^{(k)}, \quad L_{ik} = \frac{A_{ik}^{(k)}}{A_{kk}^{(k)}}$$

we find

$$\begin{aligned}
|A_{ij}^{(2)}| &\leq |A_{ij}^{(1)}| + |A_{kj}^{(1)}| \leq 2a \\
|A_{ij}^{(3)}| &\leq |A_{ij}^{(2)}| + |A_{kj}^{(2)}| \leq 4a \\
&\vdots \\
|A_{ij}^{(n)}| &\leq 2^{n-1}a.
\end{aligned}$$

We'll return to this bound in a second. It's pretty absurd.

Instead, let's bound $|B_{ij}^{(k)}| \leq Ga$ where G is called the *growth factor*. The result above suggests that the growth factor is 2^{n-1}. But let's just go ahead and use G. In that case, we get: [4]

[4] Recall that $\frac{1}{1-u} = 1 + u + u^2 + \ldots$.

$$|\mu_{ij}^{(k+1)}| \leq Ga\frac{u}{1-u} + Gau = Ga(2u - u^2)(1 + u + u^2 + \ldots) \approx 2uGa + O(u^2).$$

This is, finally, some progress. We now have a bound on all the terms μ_{ij} in the summation formulas that define the matrix E. Recall that

$$E_{ij} = \sum_{k=2}^{i} \mu_{ij}^{(k)}$$

if $j \geq i$. Thus,

$$|E_{ij}| \leq (i-1)2uGa$$

for any element in the upper triangular region. For elements in the lower-triangular region

$$|E_{ij}| = \left|\sum_{k=2}^{j+1} \mu_{ij}^{(k)}\right| \leq j2uGa.$$

We can summarize this analysis via a matrix equation:

$$|\boldsymbol{E}| \leq 2uGa \begin{bmatrix} 0 & 0 & 0 & \cdots & \cdots & 0 \\ 1 & 1 & 1 & \cdots & \cdots & 1 \\ 1 & 2 & 2 & \cdots & \cdots & 2 \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & \vdots & & n-2 & n-2 \\ 1 & 2 & 3 & \cdots & n-1 & n-1 \end{bmatrix} + O(u^2).$$

Thus, $\boldsymbol{A} + \boldsymbol{E} = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}}$.

And we're done with part 1.

GROWTH FACTORS

What we showed is that $|E_{ij}| \leq 2uGa$ where $G \leq 2^{n-1}$. That is not exactly small. And it can occur! The matrix

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix}$$

has an LU factorization with $U_{n,n} = 2^{n-1}$. This is entirely general.

While this exponential explosion in the growth factor *can* occur. It never seems to occur naturally. It only arises in a few examples that are designed to elicit it. This has provoked much study of *why* this occurs.

Sankar, Spielman, and Teng recently took up this issue. Their paper "Smoothed Analysis of the Condition Numbers and Growth Factors of Matrices" (SIMAX 2006) shows some remarkable new results about the growth factor of a random perturbation of a matrix. Here's the abstract.

> Let $\hat{\boldsymbol{A}}$ be an arbitrary matrix and let $\boldsymbol{A}$ be a slight random perturbation of $\hat{\boldsymbol{A}}$. We prove that it is unlikely that $\boldsymbol{A}$ has a large condition number. Using this result, we prove that it is unlikely that $\boldsymbol{A}$ has large growth factor under Gaussian elimination without pivoting. By combining these results, we show that the smoothed precision necessary to solve $\boldsymbol{A}\mathbf{x} = \mathbf{b}$, for any $\mathbf{b}$, using Gaussian elimination without pivoting is logarithmic. Moreover, when $\hat{\boldsymbol{A}}$ is an all-zero square matrix, our results significantly improve the average-case analysis of Gaussian elimination without pivoting performed by Yeung and Chan (SIAM J. Matrix Anal. Appl., 18 (1997), pp. 499-517).

ERRORS IN FORWARD-AND-BACK-SUBSTITUTION

Here, we'll consider the problem: [5]

$$\boldsymbol{T}\mathbf{v} = \mathbf{h}$$

where $\boldsymbol{T}$ is lower-triangular, $n \times n$, non-singular. Element-wise, we find:

$$\begin{bmatrix} T_{11} & & \\ \vdots & \ddots & \\ T_{n1} & \cdots & T_{nn} \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ h_n \end{bmatrix}.$$

Thus, we get the forward substitution procedure:

$$\begin{aligned} v_1 &= h_1/T_{11} \\ &\vdots \\ v_k &= \frac{h_k - T_{k,1}v_1 - T_{k,2}v_2 - \cdots - T_{k,k-1}v_{k-1}}{T_{kk}}. \end{aligned}$$

Let $\tilde{v}_k$ be the computed value in floating point. In floating point, this gives us:

$$\begin{aligned} \tilde{v}_k &= \left(\frac{\left(h_k - \sum_{i=1}^{k-1} T_{k,i}\tilde{v}_i(1+\omega_{k,i})\right)(1+\alpha_k)}{T_{k,k}} \right)(1+\tau_k) \\ &= \frac{h_k - \sum_{i=1}^{k-1} T_{k,i}\tilde{v}_i(1+\omega_{k,i})}{T_{k,k}/(1+\alpha_k)(1+\tau_k)}. \end{aligned}$$

With some more *coffee-shop manipulations*, we arrive at:

$$\sum_{i=1}^{k} T_{k,i}\tilde{v}_i(1+\lambda_{k,i}) = h_k$$

or

$$\boldsymbol{T}\tilde{\mathbf{v}} + \begin{bmatrix} \lambda_{1,1}T_{1,1} & & \\ \lambda_{2,1}T_{2,1} & \lambda_{2,1}T_{2,1} & \\ \vdots & \vdots & \ddots \end{bmatrix} \tilde{\mathbf{v}} = \mathbf{h}.$$

Equivalently,[6]

$$(\boldsymbol{T} + \delta\boldsymbol{T})\mathbf{v} = \mathbf{h}$$

where

$$|\delta\boldsymbol{T}| \leq u \begin{bmatrix} |T_{1,1}| & & & & \\ |T_{2,1}| & 2|T_{2,2}| & & & \\ 2|T_{3,1}| & |T_{3,2}| & 3|T_{3,3}| & & \\ \vdots & \vdots & & \ddots & \\ (n-1)|T_{n,1}| & \cdots & \cdots & |T_{n,n-1}| & n|T_{n,n}| \end{bmatrix} + O(u^2)$$

Thus, for solving a triangular system, we have errors:

$$(\boldsymbol{T} + \delta\boldsymbol{T})\mathbf{v} = \mathbf{h} \text{ where } |\delta T_{ij}| \leq nut + O(u^2) \text{ and } |T_{ij}| \leq t.$$

[5] I'm slightly less confident in the notes for this section, reader beware; Trefethen, Lecture 17 and Golub and van Loan Section 3.1 have this analysis.

[6] This matrix can take a few different forms depending on how the summation is evaluated.

OVERALL ERRORS

In our framework, the solution $\tilde{\mathbf{x}} = \mathbf{x} + \delta x$ satisfies:

$$(\boldsymbol{A} + \boldsymbol{\Delta})\mathbf{x} = \mathbf{b}$$

where $\boldsymbol{\Delta} = \boldsymbol{E} + \delta\boldsymbol{L}\tilde{\boldsymbol{U}} + \tilde{\boldsymbol{L}}\delta\boldsymbol{U} + \delta\boldsymbol{L}\delta\boldsymbol{U}$. Let's look at the maximum errors in each of these terms:

$$\max_{ij} |\delta\tilde{L}_{ij}| \leq nu + O(u^2)$$
$$\max_{ij} |\delta\tilde{U}_{ij}| \leq nuGa + O(u^2).$$

This tells us:

$$\begin{aligned}\max_{ij} |Delta_{ij}| &\leq \max_{ij} |E_{ij}| + \max_{ij} |(\delta\boldsymbol{L}\tilde{\boldsymbol{U}})_{ij}| + \max_{ij} |(\tilde{\boldsymbol{L}}\delta\boldsymbol{U})_{ij}| + \max_{ij} |(\delta\boldsymbol{L}\delta\boldsymbol{U})_{ij}| \\ &\leq 2uGan + n^2Gau + n^2Gau + O(u^2).\end{aligned}$$

Thus,

$$\|\boldsymbol{\Delta}\|_\infty \leq 2n^2(n+1)uGa.$$

This merits a theorem.

THEOREM 1 *Gaussian Elimination is Backwards Stable!*

Now, we can go even further, and bound the error in the solution as well. Let $\rho = \frac{\|\boldsymbol{\Delta}\|}{\|\boldsymbol{A}\|}$. Then

$$\rho \geq 2n^2(n+1)G$$

because $\|\boldsymbol{A}\|_\infty \geq a$. Going back to the beginning, we now have:

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \frac{\rho\kappa(\boldsymbol{A})}{1 - \rho\kappa(\boldsymbol{A})} = \frac{2n^2(n+1)G\kappa(\boldsymbol{A})}{1 - 2n^2(n+1)G\kappa(\boldsymbol{A})}$$