Error Analysis for Gauss Elimination

We now work on estimating errors in our LU decomposition algorithms. To get a sense of the central issues, we start by considering

\[ A = \begin{bmatrix} 0.0001 & 1 \\ 1 & 1 \end{bmatrix} \] in 3 digit floating point.

Now we can compute (I used Mathematica) the eigenvalues of A.

1.61806 and -0.617962

So the condition number of A is

\[ \chi(A) \approx \frac{1.61806}{-0.617962} = 2.61839 \]
This means that $A$ is well-conditioned and we should be able to solve $Ax=b$ accurately.

Let us perform LU factorization without pivoting.

Step 1.

$$l_{21} = \frac{a_{22}}{a_{11}} = \frac{4}{10^{-4}} = 10^4$$

$$u_{12} = a_{12} = 1.$$ 

$$\tilde{a}_{22} = a_{22} - l_{21} u_{12} = 4(1 - 10^4) = -10^4$$

so

$$L = \begin{bmatrix} 1 & 0 \\ 10^4 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 10^{-4} & 1 \\ 0 & -10^4 \end{bmatrix}$$

and

$$LU = \begin{bmatrix} 10^{-4} & 1 \\ 1 & 0 \end{bmatrix} \quad \text{but} \quad A = \begin{bmatrix} 10^{-4} & 1 \\ 1 & 1 \end{bmatrix}.$$
This means that anything in the A_{22} spot which rounds to $10^4$ when added to $10^4$ gives the same LU decomposition! But

$$A = \begin{bmatrix} 10^{-4} & 1 \\ 1 & 1 \end{bmatrix} \quad \text{and (for example)} \quad A' = \begin{bmatrix} 10^{-4} & 1 \\ 1 & -1 \end{bmatrix}$$

yield totally different answers to

$$Ax = b \quad \text{and} \quad A'x = b.$$

We conclude that our solver (which must give the same answer in both cases) has failed.
Example. \( Ax = [\frac{1}{2}] \). We see
\[
\begin{bmatrix}
10^{-4} & 1 \\
1 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} =
\begin{bmatrix}
1 \\
2
\end{bmatrix}
\]
has solution \([0.9999, 1.0001]\). But applying our LU decomposition, we solve
\[
\begin{bmatrix}
1 & 0 \\
10^{-4} & 1
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2
\end{bmatrix} =
\begin{bmatrix}
1 \\
2
\end{bmatrix}
\]
to get
\[
y_1 = fl(1/1) = 1
\]
\[
y_2 = fl(2 - 10^4 \cdot 1) = -10^4
\]
Then we solve
\[
\begin{bmatrix}
10^{-4} & 1 \\
0 & -10^4
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} =
\begin{bmatrix}
1 \\
-10^4
\end{bmatrix}
\]
to get
\[
x_2 = fl(-\frac{10^4}{-10^4}) = 1, \quad x_1 = fl(\frac{1 - 1}{10^{-4}}) = 0.
\]
This answer, [0], is of course completely wrong.

Example 2. We compute the condition numbers of L and U. For L, the eigenvalues are 1 and 1. But for U they are \(10^4\) and \(10^{-4}\), so the condition numbers are \(\chi(L) = 1\) \(\chi(U) = 10^8\). This, too, is a bad sign.

By comparison, if we compute this same example with partial pivoting,
we get

Step 1. Permute to get

\[
\begin{bmatrix}
  1 & 1 \\
  10^{-4} & 1
\end{bmatrix}
= PA
\]

Now

\[
l_{21} = \frac{a_{22}}{a_{11}} = f(l) \left( \frac{10^{-4}}{1} \right) = 10^{-4}
\]

\[
u_{12} = a_{12} = 1.
\]

\[
\hat{a}_{22} = a_{22} - l_{21} u_{12} = f(l - 10^{-4} \cdot 1) = 1
\]

so we get

\[
L = \begin{bmatrix}
  1 & 0 \\
  10^{-4} & 1
\end{bmatrix} \quad U = \begin{bmatrix}
  1 & 1 \\
  0 & 1
\end{bmatrix}
\]

and

\[
LU = \begin{bmatrix}
  1 & 1 \\
  10^{-4} & 1 + 10^{-4}
\end{bmatrix} \approx \begin{bmatrix}
  1 & 1 \\
  10^{-4} & 1
\end{bmatrix} = PA.
\]
Rigorous Error Bounds.

Suppose we have reordered $A$ so that all pivoting is done. We observe that each element in the upper triangular factor is given by

$$
\begin{bmatrix}
& & & \\
& & & \\
& & & \\
& & & \\
& & & \\
\end{bmatrix}
$$

updated as part of $\tilde{A}z$

for all $i < j$

copied into $U$ at step $j$.

So

$$U_{jk} = a_{jk} - \sum_{i=1}^{j-1} l_{ji} U_{ik}$$

(because the update at step $i$ was $

a_{jk} = a_{jk} - l_{ji} u_{ik}$

On the other hand, below the diagonal we get
scaled and copied into \( L \)
at step \( i = K \)

updated as part of \( \hat{A}_{zz} \)
for \( i < K \)

so

\[
l_{jk} = \frac{a_{jk} - \sum_{i=1}^{k-1} l_{ji} u_{ik}}{u_{kk}}
\]

so

\[
u_{jk} = \left( a_{jk} - \sum_{i=1}^{j-1} l_{ji} u_{ik} (1+\delta_i) \right) (1+\delta')
\]

where \( |\delta_i| \leq (j-1)\varepsilon \) and \( |\delta'| \leq \varepsilon \). So

\[
a_{jk} = \frac{1}{1+\delta'} a_{jk} + \sum_{i=1}^{j-1} l_{ji} u_{ik} (1+\delta_i)
\]

If we let \( \frac{1}{1+\delta'} = 1+\delta'_j \) and use \( l_{jj} = 1 \)
we get

\[ a_{jk} = (1+\delta_j) \sum_{i} l_{ji} u_{jk} + \sum_{i=1}^{j-1} (1+\delta_i) l_{ji} u_{ik} \]

\[ = \sum_{i=1}^{j} l_{ji} u_{ik} + \sum_{i=1}^{j} l_{ji} u_{ik} \delta_i \]

\[ = \sum_{i=1}^{j} l_{ji} u_{ik} + E_{jk}. \]

Now we can bound \( E_{jk} \):

\[ |E_{jk}| = \left| \sum_{i=1}^{j} l_{ji} \cdot u_{ik} \cdot \delta_i \right| \]

\[ \leq \sum_{i=1}^{j} |l_{ji}| \cdot |u_{ik}| \cdot |\delta_i| \cdot n \leq n. \]

since each \( \delta_i \leq (j-1) \epsilon \leq n \epsilon \). Now recall that \( l_{ji} = 0 \) for \( i > j \) since \( L \) is lower triangular. So this is really entry \( jk \) of the matrix product \( |L||U| \) where \( |A| = \) the matrix of \( a \) entries \( |l_{ai}j| \).
\[ |E_{jk}| \leq ne(\|L\|\|U\|)_{jk} \]

Now we need to do a similar analysis of the error in \( |E_{jk} | \) to write a formula for \( a_{jk} \) where \( j \geq k \). The details are in the book, but we get (in total)

\[ A = LU + E \]

where \( |E_{ij}| \leq ne |L| |U|_{ij} \). Taking norms, we get \( \| E \| \leq ne \| L \| \| U \| \). (Using homework problem 7).

It turns out to be the case that if we solve \( Ly = b \) by back substitution, the solution \( \hat{y} \) obeys
the equation

\[(L + \delta L) \hat{y} = b\]

where \(|\delta L_{ij}| < ne |L_{ij}|\) and similarly solving \(Ux = \hat{y}\) gives a solution satisfying

\[(U + \delta U) \hat{x} = \hat{y}\]

where \(|\delta U_{ij}| < ne |U_{ij}|\). So

\[b = (L + \delta L) \hat{y}\]
\[= (L + \delta L)(U + \delta U) \hat{x}\]
\[= (LU + (\delta L)U + L(\delta U) + (\delta L)(\delta U)) \hat{x}\]
\[= (A - E + (\delta L)U + L(\delta U) + (\delta L)(\delta U)) \hat{x}\]
\[= (A + \delta A) \hat{x}\]
Now (componentwise) we have

\[ |SA_{ij}| \leq |E_{ij}| + |(SL)_{ij}| + |(LSU)_{ij}| + |(L(SU))_{ij}| + |(SLSU)_{ij}| \]

\[ \leq |E_{ij}| + |(SLI \cdot LU)|_{ij} + |(LI \cdot ISU)|_{ij} + (ISLI \cdot ISU)|_{ij} \]

\[ \leq ne \ |LI \cdot LU|_{ij} + (ne \ |LI \cdot LU|)|_{ij} + (|LI| \cdot ne \ |LU|)|_{ij} + (ne \ |LI| \cdot ne \ |LU|)|_{ij} \]

\[ \approx 3ne \ (|LI \cdot LU|)|_{ij} \]

Thus to get backward stability

for a solution by LU decomposition, we want

\[ \frac{||SA||}{||A||} \approx O(e) \text{ or } 3ne \ ||LI \cdot LU|| \ll O(e) \ ||A|| \]
Unfortunately, matrices exist for which

\[
\frac{\max |U_{ij}|}{\max |A_{ij}|} = 2^{n-1}
\]

and for these matrices, the GEPP algorithm fails. (It never gets worse, see Prop. 2.1 in Demmel.)