P.12

2.5. Error analysis in GEPP

the cost of each individual floating-point operation or *flop* is satisfactory: if $*$ is one of the arithmetic operations, then

$$\frac{|\lambda * \mu - \mathrm{fl}(\lambda * \mu)|}{|\lambda * \mu|} < \varepsilon \quad \text{if } m \le |\lambda * \mu| \le M.$$

The GEPP algorithm takes as input the pair (A, b) and proceeds by computing a factorization

$$A = PLU$$

with P a permutation, L unit lower triangular, and U upper triangular. Once this is achieved, the equation

$$A x = b$$

is solved permuting the entries of b and applying forward and backward substitution. In an actual implementation, computations are performed numerically, that is, using floating-point arithmetic, thus producing an approximate solution

$$x_{\mathrm{GEPP}}.$$

To control the error in this computation, we apply the two steps:

(1) analyze roundoff errors to show the existence of a matrix A_{GEPP} such that $A_{\mathrm{GEPP}} \, x_{\mathrm{GEPP}} = b$ having a small relative error with respect to A *(backward analysis)*,

(2) apply perturbation theory to bound the relative error of x_{GEPP} with respect to x in terms of the condition number of A and the precision of our floating-point number system.

We next study this strategy for the ∞-norm, although we might equally well consider any other standard norm like the 1-norm or the 2-norm. Rounding A gives a matrix

$$\widehat{A} = A + \delta A$$

whose entries approximate in relative error those of A, up to the machine epsilon ε, as explained in (2.5). Similarly rounding b gives a vector $\widehat{b}$ whose coefficients approximate those of b with a relative error bounded by ε too. Then

$$\|\delta A\|_\infty = \max_i \sum_{j=1}^n |\delta a_{i,j}| \le \max_i \sum_{j=1}^n \varepsilon \, |a_{i,j}| \le \varepsilon \max_i \sum_{j=1}^n |a_{i,j}| = \varepsilon \, \|A\|_\infty,$$

$$\|\delta b\|_\infty = \max_i |\delta b_i| \le \max_i \varepsilon \, |b_i| \le \varepsilon \max_i |b_i| = \varepsilon \, \|b\|_\infty,$$

and so

$$\frac{\|\delta A\|_\infty}{\|A\|_\infty}, \frac{\|\delta b\|_\infty}{\|b\|_\infty} < \varepsilon.$$

By the perturbation theorem (2.10), this error will be amplified to

$$\frac{\|\delta x\|_\infty}{\|x\|_\infty} \le \frac{\kappa(A)}{1 - \varepsilon \, \kappa(A)} \, 2\,\varepsilon.$$

Hence to keep the quality of this bound, we need to show that $A_{\mathrm{GEPP}} = A + \delta A$ with

$$\frac{\|\delta A\|_\infty}{\|A\|_\infty} \le c\,\varepsilon$$

for a small constant $c > 0$. To this end, we must be careful about *pivoting*.

EXAMPLE 2.5.1. Consider the matrix

$$A = \begin{bmatrix} \eta & 1 \\ 1 & 1 \end{bmatrix}$$

with η a power of the base β that is smaller than ε. Its LU factorization (without pivoting) is $A = LU$ with

$$L = \begin{bmatrix} 1 & 0 \\ \eta^{-1} & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} \eta & 1 \\ 0 & 1 - \eta^{-1} \end{bmatrix}.$$

We have that

$$A^{-1} = \begin{bmatrix} \frac{1}{\eta-1} & \frac{-1}{\eta-1} \\ \frac{-1}{\eta-1} & \frac{\eta}{\eta-1} \end{bmatrix}, \quad L^{-1} = \begin{bmatrix} 1 & 0 \\ -\eta^{-1} & 1 \end{bmatrix}, \quad U^{-1} = \begin{bmatrix} \eta^{-1} & \frac{1}{1-\eta} \\ 0 & \frac{\eta}{\eta-1} \end{bmatrix}$$

and so

$$\kappa(A) \approx 4, \quad \kappa(L) \approx \eta^{-2}, \quad \kappa(U) \approx \eta^{-2}.$$

The disparity between the condition number of A and those of L and U indicates that this procedure is be numerically unstable. We next verify that this is indeed the case.

Set $b = (1, 2)$, so that the solution to the equation $Ax = b$ is

$$(x_1, x_2) = \left(\frac{1}{1-\eta}, \frac{1-2\eta}{1-\eta} \right) \approx (1, 1).$$

On the other hand, by the definition of the machine epsilon we have that

$$1 \ominus \eta^{-1} = \eta^{-1}(1 \ominus \eta).$$

Hence in floating-point arithmetic, the computed factors of A are

$$L_{\text{GEWP}} = \begin{bmatrix} 1 & 0 \\ \eta^{-1} & 1 \end{bmatrix} \quad \text{and} \quad U_{\text{GEWP}} = \begin{bmatrix} \eta & 1 \\ 0 & 1 \ominus \eta^{-1} \end{bmatrix} = \begin{bmatrix} \eta & 1 \\ 0 & -\eta^{-1} \end{bmatrix}.$$

Solving $L_{\text{GEWP}} \, y = b$ gives

$$y_1 = 1 \quad \text{and} \quad y_2 = 2 \ominus \eta^{-1} = -\eta^{-1}.$$

Then $U_{\text{GEWP}} \, x = y$ gives

$$x_2 = \frac{-\eta^{-1}}{-\eta^{-1}} = 1 \quad \text{and} \quad x_1 = \frac{1 \ominus 1}{\eta} = 0.$$

We obtain $x_{\text{GEWP}} = (1, 0)$, which is *not* close to the actual solution: the relative error

$$\frac{\|\delta x\|_\infty}{\|x\|_\infty} \approx 1$$

cannot be bounded by $c\varepsilon$ for a small constant $c > 0$, and so GEWP is not backward stable.

In this example, pivoting eliminates the instability just illustrated. Indeed, GEPP gives the factorization is $A = PLU$ with

(2.12) $$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 0 \\ \eta & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 1 \\ 0 & 1 - \eta \end{bmatrix}.$$

We have that $\kappa(P) = 1$, $\kappa(L) \approx 1$ and $\kappa(U) \approx 4$, and so all these factors are well-conditioned. Moreover,

$$P_{\text{GEPP}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad L_{\text{GEPP}} = \begin{bmatrix} 1 & 0 \\ \eta & 1 \end{bmatrix}, \quad U_{\text{GEPP}} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \ominus \eta \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

and successively solving $P_{\text{GEPP}}\, z = b$, $L_{GEPP}y = z$ and $U_{\text{GEPP}}x = y$ gives the accurate solution $x_{\text{GEPP}} = (1, 1)$.

2.6. Backward error analysis of GEPP

To see that GEPP is a numerically stable algorithm we need to show that the computed solution x_{GEPP} satisfies $A_{\text{GEPP}}\, x_{\text{GEPP}} = b$ with $A_{\text{GEPP}} = A + \delta A$ such that

$$\frac{\|\delta A\|_\infty}{\|A\|_\infty} \le c\,\varepsilon$$

for a small constant $c > 0$ *(backward stability)*.

As shown in Example 2.5.1, crucial information might be lost when intermediate quantities of disparate size are added together. This is an actual risk for us, because GEPP does perform a lot of additions and subtractions and, as a matter of fact, GEPP is *not* backward stable. Still we can give an interesting upper bound for the relative error it produces. In the sequel we will explain this bound and give some remarks of practical interest.

We denote by $|A|$ the $n \times n$ matrix whose entries are the absolute values of those of A, that is

$$|A| = \big[|a_{i,j}|\big]_{i,j}.$$

Analyzing the roundoff errors of the involved computations, one can find an $n \times n$ matrix A_{GEPP} with $A_{\text{GEPP}}\, x = b$ such that setting $\delta A = A - A_{\text{GEPP}}$ we have that

$$|\delta A| \le (3\,n\,\varepsilon + n^2\,\varepsilon^2)\,|L_{\text{GEPP}}|\,|U_{\text{GEPP}}|.$$

see [**Dem97**, pages 47-49] for the details. Taking norms, this readily implies that

(2.13) $$\frac{\|\delta A\|_\infty}{\|A\|_\infty} \le (3\,n\,\varepsilon + n^2\,\varepsilon^2)\,\frac{\|L_{\text{GEPP}}\|_\infty\,\|U_{\text{GEPP}}\|_\infty}{\|A\|_\infty}.$$

The practice of numerical linear solving is that GEPP *almost* always keeps

$$\|L_{\text{GEPP}}\|_\infty\,\|U_{\text{GEPP}}\|_\infty \approx \|A\|_\infty,$$

as in (2.12). If this were the case, then we would have

$$\frac{\|\delta_{\text{GEPP}}A\|}{\|A\|} \lesssim n\,\varepsilon.$$

Thus we say that GEPP is backward stable *in practice*.

For a theoretical upper bound we can consider the *pivot growth factor*

$$g_{\text{GEPP}} = \frac{\max_{i,j}|u_{i,j}|}{\max_{i,j}|a_{i,j}|}.$$

GEPP guarantees that the entries of L_{GEPP} are bounded by 1 in absolute value and so

$$\|L\|_\infty \le n \quad \text{and} \quad \|U\|_\infty \le n\,g_{\text{GEPP}}\,\|A\|_\infty.$$

Together with (2.13) this implies that

(2.14) $$\frac{\|\delta A\|_\infty}{\|A\|_\infty} \le 3\,n^3\varepsilon\,g_{\text{GEPP}}.$$

Backward stability amounts to the fact that g_{GEPP} is small or grows slowly as a function of n. In general, we have that

$$g_{\text{GEPP}} \le 2^{n-1},$$

and unfortunately, there are rare cases where this exponential bound is attained: for instance, when $n = 4$ we have that

$$
\begin{bmatrix}
1 & 0 & 0 & 1 \\
-1 & 1 & 0 & 1 \\
-1 & -1 & 1 & 1 \\
-1 & -1 & -1 & 1
\end{bmatrix}
=
\begin{bmatrix}
1 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 \\
-1 & -1 & 1 & 0 \\
-1 & -1 & -1 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 2 \\
0 & 0 & 1 & 4 \\
0 & 0 & 0 & 8
\end{bmatrix},
$$

and this example can be easily generalized to an $n \times n$ matrix with pivot growth 2^{n-1}