Accuracy of Gaussian Elimination in finite precision arithmetic

For an $n \times n$ matrix $A$ assume that in exact arithmetic $A = LU$. If we use calculations on a finite precision computer we will compute $\hat{L}$ and $\hat{U}$ and not the true $L$ and $U$. Therefore $\hat{L}\hat{U} \neq A$. Assume $\hat{L}\hat{U} = A + E$.

**Best possible:** Let $E = (e_{ij})$. In finite precision using a floating point number system, numbers that are stored on the computer may have a relative error as large as relative machine precision or machine $\epsilon$. In standard double precision computer arithmetic $\epsilon$ is small, approximately $10^{-16}$. This error occurs just from storing a number (for example there will be a small error if one stores $\pi$ keeping a finite number of digits) prior to any computations. Since $\epsilon$ is a bound on the relative error in storing an element $a_{ij}$ of $A$, $|e_{ij}| \leq \epsilon |a_{ij}|$ for $i, j = 1, \ldots, n$. We conclude that when there are errors in storing the matrix $A$ but no other errors:

$$\max_{i,j=1,\ldots,n} |e_{ij}| \leq \epsilon \max_{i,j=1,\ldots,n} |a_{ij}|.$$  \hspace{1cm} (1)

With finite precision calculations: In finite precision computations errors can be introduced at each step of an algorithm, for example from storing intermediate results. Let $a_{ij}^{(k)}$ indicate an element of the current $A$ or $U$ (but not $L$) matrix at the $k^{th}$ step of Gaussian elimination. So $a_{ij}^{(1)}$ will be the $i, j^{th}$ element of the original $A$ and $a_{ij}^{(n)}$ will be the $i, j^{th}$ element of the final upper triangular matrix $U$. In finite precision we have:

**Theorem 1.1** For Gaussian elimination without pivoting on a computer with relative machine precision $\epsilon$, the computed $\hat{L}$ and $\hat{U}$ satisfy

$$\hat{L}\hat{U} = A + E$$ \hspace{1cm} (2)

with

$$\max_{i,j=1,\ldots,n} |e_{ij}| \leq \rho f(n) \epsilon \max_{i,j=1,\ldots,n} |a_{ij}|.$$ \hspace{1cm} (3)

where $f(n)$ is a function that is not large and is about the same for all methods and

$$\rho = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}$$

is called the growth factor.

The bound (3) also applies for Gaussian elimination with pivoting if (2) is replaced by $\hat{L}\hat{U} = P(A + E)Q$ where $P$ and $Q$ are permutation matrices.

Comparing equation (3) with equation (1) shows that the factors $\rho$ and $f(n)$ in equation (3) affect the growth of the errors due to the computations in the algorithm. Since $f(n)$ is similar in size for all methods the key to comparing the accuracy of different algorithms is the growth factor $\rho$.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\rho$ in theory</th>
<th>$\rho$ in practice</th>
<th>stable in theory</th>
<th>stable in practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENP</td>
<td>$\infty$</td>
<td>usually 10 or less, perhaps $n$ at times</td>
<td>no, exponentially large</td>
<td>often no</td>
</tr>
<tr>
<td>GEPP</td>
<td>$2^{n-1}$</td>
<td>usually 10 or less, perhaps $n$ at times</td>
<td>?, not exponential</td>
<td>yes</td>
</tr>
<tr>
<td>GECP</td>
<td>$2\sqrt{nn\ln(n)/4}$</td>
<td>usually 10 or less, perhaps $n$ at times</td>
<td>?, not exponential</td>
<td>yes</td>
</tr>
<tr>
<td>GERP</td>
<td>$1.5n^3\ln(n)/4$</td>
<td>usually 10 or less, perhaps $n$ at times</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

Open questions:
• Why is GEPP stable in practice. Is Murphy’s law (what can go wrong will go wrong) incorrect.
• For random matrices can a probabilistic bounds on $\rho$ be developed for GEPP?
• What are sharper theoretical bounds on $\rho$ for GECP or GERP?