Numerical Matrix Analysis
Lecture Notes #18 — Systems of Equations
Gaussian Elimination / Cholesky Factorization

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Spring 2016
1 Gaussian Elimination
   - Last Time...
   - Stability
   - Backward Stability? Practical Stability?

2 Cholesky Factorization
   - Hermitian Positive Definite Matrices
   - $R^*R$-factorization

3 Reference

Peter Blomgren, (blomgren.peter@gmail.com)
Flashback: Last Time

We quickly reviewed a familiar algorithm — **Gaussian Elimination**.

If we save the multipliers generated by the elimination, we get the **LU-factorization** of $A$, i.e. $A = LU$, where $L$ is lower triangular, and $U$ is upper triangular.

In this initial form, GE/LU is completely useless (unstable), we discussed a couple of fixes, some probably familiar, some new...
In **Partial Pivoting** we rearrange the rows of the matrix $A$ (on the fly) in order to move the largest element in the “active” column to the diagonal entry — this way we can guarantee that the multiplier is bounded by one

$$ \tilde{l}_{ji} = a_{ji} \odot a_{ii} = \frac{a_{ji}}{a_{ii}} (1 + \epsilon), \quad |\epsilon| \leq \epsilon_{\text{mach}}, \quad |\delta_{ji}| \leq \epsilon_{\text{mach}} l_{ji} $$

We get $PA = LU$
Partial Pivoting is stable “most of the time.” We looked at enhancements taking scale into consideration: Scaled Partial Pivoting.

The overall work for GE/LU is $\sim \frac{2m^3}{3}$, and partial pivoting adds $O(m^2)$ operations, which is a small cost.

Sometimes Complete Pivoting — rearrangement of both the rows and columns of $A$ is necessary to achieve high accuracy. The cost is significant since the additional work adds $O(m^3)$ operations.

We get $PAQ = LU$
— We look at the stability of Gaussian elimination.

— Gaussian Elimination for **Hermitian Positive Definite Matrices:**

— Cholesky Factorization — The Hermitian (Symmetric) version of LU-factorization.
“Gaussian Elimination with partial pivoting is explosively unstable for certain matrices, yet stable in practice. This apparent paradox has a statistical explanation.”

[Trefethen-&-Bau, p.163]

The stability analysis of Gaussian Elimination with Partial Pivoting (GE w/PP) is complicated, consider the example $A = LU$

$$
\begin{bmatrix}
10^{-20} & 1 \\
1 & 1
\end{bmatrix} =
\begin{bmatrix}
1 & 0 \\
10^{20} & 1
\end{bmatrix}
\begin{bmatrix}
10^{-20} & 1 \\
0 & 1 - 10^{20}
\end{bmatrix}
$$

The likely computed $\tilde{L}$ and $\tilde{U}$ are

$$
\begin{bmatrix}
1 & 0 \\
10^{20} & 1
\end{bmatrix}
\begin{bmatrix}
10^{-20} & 1 \\
0 & -10^{20}
\end{bmatrix}
= 
\begin{bmatrix}
10^{-20} & 1 \\
1 & 0
\end{bmatrix} \neq A
$$
This behavior is quite generic — instability in Gaussian Elimination (with or without pivoting) can arise if the factors $\tilde{L}$ or $\tilde{U}$ are large compared with $A$.

In the previous example we have

$$\|A\|_F = 1.7321, \quad \|\tilde{L}\|_F = 1.0000 \times 10^{20}, \quad \|\tilde{U}\|_F = 1.0000 \times 10^{20}$$

i.e. the computed factors are 20 orders of magnitude larger than the initial matrix — no wonder we run into problems!

The purpose of pivoting — from the point of view of stability/accuracy — is to make sure that $\tilde{L}$ and $\tilde{U}$ are not too large.
Theorem

Let the factorization $A = LU$ of a non-singular matrix $A \in \mathbb{C}^{m \times m}$ be computed by Gaussian Elimination without pivoting in a floating point environment satisfying the floating point axioms. If $A$ has an $LU$-factorization, then for $\epsilon_{\text{mach}}$ small enough, the factorization completes successfully in floating point arithmetic (no zero pivots $\tilde{a}_{ii}$ are encountered), and the computed matrices $\tilde{L}$, and $\tilde{U}$ satisfy

$$\tilde{L}\tilde{U} = A + \delta A, \quad \frac{\|\delta A\|}{\|L\|\|U\|} = \mathcal{O}(\epsilon_{\text{mach}})$$

for some $\delta A \in \mathbb{C}^{m \times m}$.

Note that we can make the theorem apply to GE w/Pivoting by applying it to the “pre-pivoted matrix:” $A := PA[Q]$. 
If we just flash by the previous slide, the result look just like all the other backward stability results... **BUT!!!** take a closer look... we have

$$\frac{\|\delta A\|}{\|L\| \|U\|} = \mathcal{O}(\epsilon_{\text{mach}}).$$

Usually, the results contain something like

$$\frac{\|\delta A\|}{\|A\|} = \mathcal{O}(\epsilon_{\text{mach}}).$$

There is a **critical difference** here. If $\|L\| \|U\| = \mathcal{O}(\|A\|)$, then the theorem states that GE is backward stable. However, if $\|L\| \|U\| \gg \mathcal{O}(\|A\|)$, all bets are off!
Quantifying Stability

Without pivoting, both $\|L\|$ and $\|U\|$ can be unbounded, and GE w/o Pivoting is unstable by any standard.

Consider GE w/PP. By construction $|l_{ij}| \leq 1$, so that $\|L\| = \mathcal{O}(1)$ in any norm (this is true for all the pivoting schemes we have discussed). We now focus our attention to $U$; essentially GE w/PP is backward stable provided $\|U\| = \mathcal{O}(\|A\|)$.

The following quantity turns out to be very useful:

**Definition (Growth Factor)**

The **growth factor** of $A$ (and the algorithm) is defined as the ratio

$$\rho = \frac{\max_{i,j} |u_{ij}|}{\max_{i,j} |a_{ij}|}$$
The Growth Factor... and Stability

If $\rho \sim 1$, there is little growth, and the elimination process is stable. When $\rho$ is large, we expect loss of accuracy and/or instability of the algorithm... We make this precise: —

**Theorem**

Let the factorization $PA = LU$ of a non-singular matrix $A \in \mathbb{C}^{m \times m}$ be computed by GE w/PP in a floating point environment satisfying the floating point axioms. The computed matrices $\tilde{P}$, $\tilde{L}$, and $\tilde{U}$ satisfy

$$\tilde{L}\tilde{U} = \tilde{P}A + \delta A, \quad \frac{||\delta A||}{||A||} = \mathcal{O}(\rho \epsilon_{mach})$$

for some $\delta A \in \mathbb{C}^{m \times m}$, where $\rho$ is the growth factor of $A$. If $|l_{ij}| < 1$ for $i > j$, then $P = \tilde{P}$ for $\epsilon_{mach}$ small enough.
If $\rho = O(1)$ uniformly for all matrices of a given dimension $m$, then GE w/PP is backward stable; otherwise it is not.

**Let the mathematical hair-splitting begin!**

Consider the worst-case scenario

$$
\begin{bmatrix}
1 & 1 & \cdots & 1 \\
-1 & 1 & \cdots & 1 \\
-1 & -1 & 1 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots & \ddots \\
-1 & -1 & \cdots & -1 & 1 & 1 \\
-1 & -1 & \cdots & -1 & -1 & 1 \\
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
1 \\
1 \\
\vdots \\
1 \\
1 \\
\end{bmatrix}
$$
If $\rho = \mathcal{O}(1)$ uniformly for all matrices of a given dimension $m$, then GE w/PP is backward stable; otherwise it is not.

Let the mathematical hair-splitting begin!

Consider the worst-case scenario

$$
\begin{bmatrix}
1 & 1 & 1 \\
-1 & 1 & 1 \\
-1 & -1 & 1 \\
\vdots & \vdots & \vdots \\
-1 & -1 \ldots & -1 & 1 \\
-1 & -1 \ldots & -1 & 1 \\
\end{bmatrix}
= 
\begin{bmatrix}
1 & -1 & -1 \\
1 & -1 & -1 \\
1 & -1 & -1 \\
\vdots & \vdots & \vdots \\
1 & -1 \ldots & -1 \\
1 & -1 \ldots & -1 \\
\end{bmatrix}
\begin{bmatrix}
1 & 1 & 1 \\
1 & 2 \\
1 & 4 \\
\vdots \\
1 & 2^{m-2} \\
1 & 2^{m-1} \\
\end{bmatrix}
$$

Here $\rho = 2^{m-1}$, which is the maximal value $\rho$ can take for GE w/PP.
A growth factor of $2^{m-1}$ corresponds to a loss of $\sim (m - 1)$ bits of information (Recall: we have at most 52 binary digits in IEEE-754-1985 double precision floating point computations).

According the worst-case estimate we cannot safely operate on matrices of dimension larger than $52 \times 52$, and in that case only have one bit of information! This is an intolerable state of affairs for practical computations!!!
On the other hand... We have a uniform bound \((2^{m-1})\) on the growth factor for \(m \times m\)-matrices, thus according to our previous definitions of backward stability; \textbf{GE w/PP is backward stable.}

Clearly, \textbf{for practical purposes}, this is an absurd conclusion. In this context, let’s put the previous formal definition of backward stability aside; and say that the worst-case scenario indicates that \textbf{GE w/PP can be unstable.}
Now... If GE w/PP is so unstable, why is it so famous and popular?!?

“Despite worst-case examples, GE w/PP is utterly stable in practice. Large factors U like the one in the worst-case scenario never seem to appear in real applications. In 50 years of computing no matrix problems that excite explosive instability are known to have arisen under natural circumstances.”

[Trefethen-&-Bau (1997), p.166]

In “Matrix Computations” by Golub & Van-Loan, the upper bounds for the growth factors for partial and complete pivoting are given as

$$\rho_{PP} \leq 2^{m-1}, \quad \rho_{CP} \leq 1.8m^{\left(\frac{\ln m}{4}\right)}.$$
The number of matrices with large growth factors is very small — if we select a random matrix in $\mathbb{C}^{m \times m}$ it turns out that a practical bound on $\rho_{PP}$ is given by $\sqrt{m}$. This is illustrated below.

**Figure:** The growth factors for GE w/PP for 500 random matrices ranging in size from $2 \times 2$ to $1448 \times 1448$. The blue line (left panel) corresponds to the practical bound $\sqrt{m}$; and the red line (right panel only) corresponds to the worst-case bound for complete pivoting, $\rho_{cp}$.
Where is the $\rho_{pp}$ line?!

Figure: The corresponding values for $\rho_{pp}$ are $\geq \{2, 8, 16, 128, 10^3, 10^4, 10^6, 10^9, 10^{13}, 10^{18}, 10^{26}, 10^{38}, 10^{54}, 10^{76}, 10^{108}, 10^{153}, 10^{217}, 10^{307}, 10^{435}\}$, whereas in this ($m \in \{2, \ldots, 1448\}$) range, $\rho_{cp} < 10^7$. 
The bottom line is that GE w/PP works well “almost always.”

It is almost impossible to prove any useful result in this context.

Vigorous hand-waving and numerical recovery of the probability density functions for the growth-factor vs. the matrix size can be used to get indications that the number of matrices with large growth factors is exponentially small in a probabilistic sense.

See e.g. Trefethen-&-Bau pp.166–170, for some discussion.
We now turn our attention to application of Gaussian Elimination / LU-Factorization to a special class of matrices —

**Definition (Hermitian Positive Definite)**

\[
A \in \mathbb{C}^{m \times m} \text{ is Hermitian Positive Definite if } A = A^*, \text{ and } \\
\tilde{x}^* A \tilde{x} > 0, \quad \forall \tilde{x} \in \mathbb{C}^m - \{\tilde{0}\}.
\]

This type of matrices show up *many* applications — due to symmetry (reciprocity) in physical systems.

My favorite application is *optimization* (Math 693a), where we constantly build second order models

\[
m_k(\tilde{p}) = f(\tilde{x}_k) + \tilde{p} \nabla f(\tilde{x}_k) + \frac{1}{2} \tilde{p}^* B_k \tilde{p}_k
\]

where the matrix \( B_k \approx \nabla^2 f(\tilde{x}_k) \) is symmetric (Hermitian) positive definite.
Let $A \in \mathbb{C}^{m \times m}$ be HPD.

- $\forall X \in \mathbb{C}^{m \times n}, m \geq n, \text{rank}(X) = n; X^*AX$ is also HPD.
- By selecting $X \in \mathbb{C}^{m \times n}$ to be a matrix with a 1 in each column, and zeros everywhere else, we can write any $n \times n$ principal sub-matrix of $A$ in the form $X^*AX$. It follows that every principal sub-matrix of $A$ must be HPD, and $a_{ii} \in \mathbb{R}^+$.  
  
- $\lambda(A) \in \mathbb{R}^+$.  
- Eigenvectors that correspond to distinct eigenvalues of a Hermitian matrix are orthogonal.
We now turn to the main task at hand — decomposing a HPD matrix into triangular factors, $R^*R$...

We assume that $A$ is an HPD matrix, and write it in the form

$$
\begin{pmatrix}
\alpha & \tilde{w}^* \\
\tilde{w} & B
\end{pmatrix}
= 
\begin{pmatrix}
\beta & \tilde{0}^* \\
\tilde{w} / \beta & I(n-1)
\end{pmatrix}
\begin{pmatrix}
1 & \tilde{0} \\
\tilde{0} & B - ww' / a
\end{pmatrix}
\begin{pmatrix}
\beta & \tilde{w}^* / \beta \\
\tilde{0} & I(n-1)
\end{pmatrix}
$$

Where

$$
\beta = \sqrt{\alpha}, \quad \tilde{0} \text{ the zero-vector}, \quad B - ww' / a := B - \tilde{w}\tilde{w}^* / \alpha,
$$

$I(n-1)$ the $(n - 1) \times (n - 1)$-identity matrix

Before moving forward, we check the matrix identity...
We have

\[
\begin{bmatrix}
\beta \\
\tilde{w} / \beta
\end{bmatrix}
\begin{bmatrix}
\tilde{0}^* \\
I(n-1)
\end{bmatrix}
\begin{bmatrix}
1 \\
\tilde{0}
\end{bmatrix}
\begin{bmatrix}
\tilde{0} \\
B - \tilde{w}\tilde{w}' / a
\end{bmatrix}
\begin{bmatrix}
\beta \\
\tilde{w}^* / \beta
\end{bmatrix}
\begin{bmatrix}
\tilde{0} \\
I(n-1)
\end{bmatrix}
\]

Multiplying the first two matrices, and then third together gives

\[
\begin{bmatrix}
\beta \\
\tilde{w} / \beta
\end{bmatrix}
\begin{bmatrix}
\tilde{0}^* \\
B - \tilde{w}\tilde{w}' / a
\end{bmatrix}
\begin{bmatrix}
\beta \\
\tilde{w}^* / \beta
\end{bmatrix}
\begin{bmatrix}
\tilde{0} \\
I(n-1)
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\alpha \\
\tilde{w}
\end{bmatrix}
\begin{bmatrix}
\tilde{w}^*
\end{bmatrix}
\begin{bmatrix}
\tilde{w} \\
B
\end{bmatrix}
\]

as desired.
It can be shown (see slides 31–32) that the sub-matrix $B - \tilde{\mathbf{w}}\tilde{\mathbf{w}}^*/\alpha$ is also HPD.

We can now define the Cholesky Factorization recursively:

$$R^{(n)} = \begin{bmatrix} \beta & \tilde{\mathbf{w}}^*/\beta \\ \tilde{\mathbf{0}} & R^{(n-1)} \end{bmatrix}$$

Where $R^{(n-1)} = R^{(n-1)}$ is the Cholesky factor $R$ associated with $B - \tilde{\mathbf{w}}\tilde{\mathbf{w}}^*/\alpha$, i.e. $[R^{(n)}]^*[R^{(n)}] = B - \tilde{\mathbf{w}}\tilde{\mathbf{w}}^*/\alpha$.

A note on the implementation (next slide): Since we only need to compute one of the triangular parts (it’s Hermitian, remember?!?) of the factorization, the Cholesky factorization uses about 1/2 the operations of a general $LU$-factorization.
% Cholesky Factorization of an m-by-m matrix A
for i=1:m
    %
    % compute \( \tilde{\mathbf{w}}^*/\beta \)
    %
    A(i,i) = sqrt(A(i,i));
    A(i,(i+1):m) = A(i,(i+1):m) / A(i,i);
    %
    % compute the upper triangular part of \( B - \tilde{\mathbf{w}}\tilde{\mathbf{w}}^*/\alpha \)
    %
    for j=(i+1):m
        A(j,j:m) = A(j,j:m) - A(i,j:m) * A(i,j)';
    end
    %
    % We zero out the super-diagonal elements, since
    % the answer is an upper triangular matrix.
    %
    A((i+1):m,i) = zeros(m-i,1);
end
Theorem

Every HPD matrix $A \in \mathbb{C}^{m \times m}$ has a unique Cholesky factorization.

The existence follows from the argument on slides 31–32, and uniqueness from the algorithm. □

Compared with standard Gaussian elimination / LU-factorization, we are saving about half the operations since we only form the upper triangular part $R$
Usually when we see this table

<table>
<thead>
<tr>
<th>Method</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cholesky $R^* R$ Factorization</td>
<td>$\frac{m^3}{3}$</td>
</tr>
<tr>
<td>LU-Factorization</td>
<td>$\frac{2m^3}{3}$</td>
</tr>
<tr>
<td>QR: Householder</td>
<td>$\frac{4m^3}{3}$</td>
</tr>
<tr>
<td>QR: Gram-Schmidt</td>
<td>$2m^3$</td>
</tr>
<tr>
<td>SVD</td>
<td>$13m^3$</td>
</tr>
</tbody>
</table>

we note that with increased cost comes increased stability. The Cholesky factorization is the one pleasant exception!

All the subtle things that can go wrong in general LU-factorization (Gaussian elimination) are safe in the Cholesky factorization context!

**Cholesky factorization is always stable!** (For HPD matrices, that is.)
In the 2-norm we have $\|R\| = \|R^*\| = \sqrt{\|A\|}$, thus the growth factor cannot be large. We also note that we can safely compute the Cholesky factorization **without pivoting**.

**Theorem**

Let $A \in \mathbb{C}^{m \times m}$ be HPD, and let $R^* R = A$ be computed using the Cholesky factorization algorithm in a floating point environment satisfying the floating point axioms. For sufficiently small $\epsilon_{\text{mach}}$, this process is guaranteed to run to completion (no zero or negative entries $r_{kk}$ will arise), generating a computed factor $\tilde{R}$ that satisfies

$$\tilde{R}^* \tilde{R} = A + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{\text{mach}})$$

for some $\delta A \in \mathbb{C}^{m \times m}$. 
If $A$ is HPD, the standard (best) way to solve $A\tilde{x} = \tilde{b}$ is by Cholesky decomposition.

Once we have $R^*R\tilde{x} = \tilde{b}$, we get the solution by solving $R^*\tilde{y} = \tilde{b}$ (by forward substitution), followed by $R\tilde{x} = \tilde{y}$ (by backward substitution). Each triangular solve requires $\sim m^2$ operations, so the total work is $\sim \frac{1}{3}m^3$. 
We have the following important result

**Theorem**

The solution of an HPD system $A\tilde{x} = \tilde{b}$ via Cholesky factorization is backward stable, generating a computed solution $\tilde{x}$ that satisfies

$$(A + \Delta A)\tilde{x} = \tilde{b}, \quad \frac{\|\Delta A\|}{\|A\|} = \mathcal{O}(\epsilon_{\text{mach}})$$

for some $\Delta A \in \mathbb{C}^{m \times m}$.
If we have a Hermitian matrix $A \in \mathbb{C}^{m \times m}$ the best way to check if it is also Positive Definite is to try to compute the Cholesky factorization.

If $A$ is not HPD, then the Cholesky factorization will break down in the sense that

$$\sqrt{r_{kk}} \quad \text{or, if you want} \quad \sqrt{A(i,i)}$$

will fail (if $r_{kk} < 0$) or the subsequent division by $\sqrt{r_{kk}}$ will fail (if $r_{kk} = 0$).

Usually, in applications (such as optimization) we require $A$ to be sufficiently HPD, meaning that we must have $r_{kk} \geq \delta > 0$ for some $\delta$. Quite possibly $\delta \in \{\sqrt{\epsilon_{\text{mach}}}, 3\sqrt{\epsilon_{\text{mach}}}\}$. 
If $A$ is HPD, and $X$ is a non-singular matrix, then $B = X^*AX$ is also HPD: since $X$ is non-singular $\tilde{x} \neq 0 \Rightarrow X\tilde{x} \neq 0$, hence

$$\forall \tilde{x} \neq 0, \quad \tilde{x}^*B\tilde{x} = \tilde{x}^*X^*AX\tilde{x} = (X\tilde{x})^*A(X\tilde{x}) > 0$$

Now, with the representation

$$A = \begin{bmatrix} \beta^2 & \tilde{w}^* \\ \tilde{w} & B \end{bmatrix}$$

We select

$$X = \begin{bmatrix} \frac{1}{\beta} & -\tilde{w}^*/\beta^2 \\ \tilde{0} & I(n-1) \end{bmatrix}, \quad X^* = \begin{bmatrix} 1/\beta & \tilde{0}^* \\ -\tilde{w}/\beta^2 & I(n-1) \end{bmatrix}$$
Now,

\[ X^*AX = \begin{bmatrix} 1/\beta & \tilde{0}^* \\ -\tilde{w}/\beta^2 & I(n-1) \end{bmatrix} \begin{bmatrix} \beta^2 & \tilde{w}^* \\ \tilde{w} & B \end{bmatrix} \begin{bmatrix} 1/\beta & -\tilde{w}^*/\beta^2 \\ \tilde{0} & I(n-1) \end{bmatrix} \]

\[ = \begin{bmatrix} \beta & \tilde{w}^*/\beta \\ \tilde{0} & B - \tilde{w}\tilde{w}^*/\alpha \end{bmatrix} \begin{bmatrix} 1/\beta & -\tilde{w}^*/\beta^2 \\ \tilde{0} & I(n-1) \end{bmatrix} = \begin{bmatrix} 1 & \tilde{0} \\ \tilde{0} & B - \tilde{w}\tilde{w}^*/\alpha \end{bmatrix} \]

It now follows from the definition (use \( \tilde{x} \neq 0 \) such that \( x_1 = 0 \)) that \( B - \tilde{w}\tilde{w}^*/\beta^2 \) is also HPD.