# LU factorization

Prof. Dr. Heinrich Voss
voss@tuhh.de

Technische Universität Hamburg-Harburg

Wintersemester 2008/2009

TUHH
Technische Universität Hamburg-Harburg

# Gaussian elimination

The simplest way to solve a linear system (by hand or on a computer) is Gaussian elimination.

It transforms a linear system to an equivalent one with upper-triangular system matrix by applying simple linear transformations.

Let $A \in \mathbb{C}^{m\times n}$ be given. The idea is to transform $A$ into an upper-triangular matrix by introducing zeros below the diagonal, first in column 1, then in column 2 , etc. This is done by subtracting suitable multiples of each row from the subsequent ones.

This elimination process is equivalent to multiplying $A$ by a sequence of lower triangular matrices $L_j$ on the left:

$$L_{n-1} L_{n-2} \cdots L_1 A = U.$$

# LU factorization

Setting $L := L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1}$ gives $A = LU$. Thus we obtain an LU factorization of $A$

$$A = LU,$$

where $U$ is upper-triangular, and $L$ is (as a product of lower-triangular matrices) lower-triangular.

It turns out that $L$ can be chosen such that all diagonal entries are equal to 1. A matrix with this property is called unit lower-triangular.

# Example

$$A = \begin{pmatrix} 2 & 1 & 3 & 4 \\ -2 & 1 & -1 & -2 \\ 4 & 4 & 5 & 11 \\ -2 & 1 & -7 & -1 \end{pmatrix}$$

The first step of Gaussian elimination looks like this: The first row is added to the second one, twice the first row is subtracted from the third one, and the first row is added to third on. This can be written as

$$L_1 A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 & 4 \\ -2 & 1 & -1 & -2 \\ 4 & 4 & 5 & 11 \\ -2 & 1 & -7 & -1 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 3 & 4 \\ 0 & 2 & 2 & 2 \\ 0 & 2 & -1 & 3 \\ 0 & 2 & -4 & 3 \end{pmatrix}$$

Next we subtract the second row from the third and the fourth row:

$$L_2 L_1 A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 & 4 \\ 0 & 2 & 2 & 2 \\ 0 & 2 & -1 & 3 \\ 0 & 2 & -4 & 3 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 3 & 4 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & -3 & 1 \\ 0 & 0 & -6 & 1 \end{pmatrix}$$

# Example ct.

$$L_2 L_1 A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 & 4 \\ 0 & 2 & 2 & 2 \\ 0 & 2 & -1 & 3 \\ 0 & 2 & -4 & 3 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 3 & 4 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & -3 & 1 \\ 0 & 0 & -6 & 1 \end{pmatrix}$$

Finally we subtract twice the third row from the fourth row

$$L_3 L_2 L_1 A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 & 4 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & -3 & 1 \\ 0 & 0 & -6 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 3 & 4 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & -3 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

To exhibit the full factorization $A = LU$ we need to compute the product $L = L_1^{-1} L_2^{-1} L_3^{-1}$.

Surprisingly, this turns out to be trivial. The inverse of $L_j$, $j = 1, 2, 3$ is just $L_j$ itself, but with each entry below the diagonal negated:

# Example ct.

$$L_1^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}, \ldots$$

The product $L_1^{-1}L_2^{-1}L_3^{-1}$ is just the unit lower-triangular matrix with the nonzero subdiagonal entries of $L_1^{-1}$, $L_2^{-1}$ and $L_3^{-1}$ inserted in the appropriate places:

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 0 \\ -1 & 1 & 2 & 1 \end{pmatrix}.$$

Together we have

$$A = \begin{pmatrix} 2 & 1 & 3 & 4 \\ -2 & 1 & -1 & -2 \\ 4 & 4 & 5 & 11 \\ -2 & 1 & -7 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 0 \\ -1 & 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 & 4 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & -3 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix} = LU.$$

# General case

Suppose $x_k$ is the $k$th column of the matrix at the beginning of step $k$. Then the transformation has to be chosen such that

$$x_k = \begin{pmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ x_{k,k+1} \\ \vdots \\ x_{mk} \end{pmatrix} \rightarrow \begin{pmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

To this end we subtract $\ell_{jk}$ times row $k$ from row $j$ with $\ell_{jk} = x_{jk}/x_{kk}$, $k < j \leq m$, which is performed multiplying by

$$L_k = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\ell_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & -\ell_{mk} & & 1 \end{pmatrix}.$$

# General case ct.

In the numerical example, we noted that $L_k$ can be inverted by negating its subdiagonal entries, and that $L$ can be formed by collecting the entries $\ell_{jk}$ in the appropriate places. These observations are true in the general case.

With

$$\ell_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \ell_{k+1,k} \\ \vdots \\ \ell_{mk} \end{pmatrix}$$

the matrix $L_k$ can be written $L_k = I - \ell_k e_k^H$, where $e_k$ is the $k$th standard unit vector. The sparsity pattern of $\ell_k$ implies $e_k^H \ell_k = 0$, and therefore

$$(I - \ell_k e_k^H)(I + \ell_k e_k^H) = I - \ell_k e_k^H \ell_k e_k^H = I.$$

In other words, the inverse of $L_k$ is $I + \ell_k e_k^H$.

# General case ct.

That $L = L_1^{-1} \cdots L_m^{-1}$ can be formed by collecting the entries $\ell_{jk}$ in the appropriate places is proved by induction.

Assume that

$$L_1^{-1} \cdots L_k^{-1} = I + \sum_{j=1}^{k} \ell_j e_j^H.$$

Then it follws from $\ell_j e_j^H \ell_{k+1} = 0$ for $j = 1, \ldots, k$

$$\begin{aligned} L_1^{-1} \cdots L_k^{-1} L_{k+1}^{-1} &= (I + \sum_{j=1}^{k} \ell_j e_j^H)(I + \ell_{k+1} e_{k+1}^H) \\ &= I + \sum_{j=1}^{k+1} \ell_j e_j^H + \sum_{j+1}^{k} \ell_j e_j^H \ell_{k+1} e_{k+1}^H. = I + \sum_{j=1}^{k+1} \ell_j e_j^H. \end{aligned}$$

# Gaussian elimination

In practical Gaussian elimination, the matrices $L_k$ are never formed and multiplied explicitly. The multipliers $\ell_k$ are computed and stored directly into $L$, and the transformations $L_k$ are then applied implicitly.

## Gaussian elimination without pivoting

$U = A$, $L = I$
**for** k=1:m-1 **do**
  **for** j=k+1:m **do**
    $\ell_{jk} = u_{jk} / u_{kk}$
    $u_{j,k:m} = u_{j,k:m} - \ell_{jk} u_{k,k_m}$
  **end for**
**end for**

Three matrices $A$, $L$, $U$ are not really needed; to minimize memory use on the computer, both $L$ and $U$ can be written into the same array as $A$.

# Linear systems

If $A$ is factored into $L$ and $U$, a system of equations $Ax = b$ is reduced to the form $LUx = b$.

Thus it can be solved by solving two triangular systems: first $Ly = b$ for the unknown $y$ (forward substitution), then $Ux = y$ for the unknown $x$ (back substitution).

This is particularly advantageous, if several linear systems with the same system matrix have to be solved.

# Failure of Gaussian elimination

Unfortunately, Gaussian elimination as presented so far is unusable for solving general linear systems, for it is not stable.

The instability is related to another, more obvious difficulty. For certain matrices, Gaussian elimination fails entirely, because it attempts division by zero. For example, consider

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

This matrix has full rank. Nevertheless, Gaussian elimination fails at the first step.

# Pivoting

At step $k$ of Gaussian elimination, multiples of row $k$ are subtracted from rows $k+1, \ldots, m$ of the working matrix $X$ in order to introduce zeros in entry $k$ of these rows.
In this operation row $k$, column $k$, and especially the entry $x_{kk}$ play special roles. We call $x_{kk}$ the pivot.

From every entry in the submatrix $X_{k+l:m,k:m}$ is subtracted the product of a number in row $k$ and a number in column $k$, divided by $x_{kk}$

However, there is no reason why the $k$th row and column must be chosen for the elimination. For example, we could just as easily introduce zeros in column $k$ by adding multiples of some row $i$ with $k < i < m$ to the other rows.

Similarly, we could introduce zeros in column $j$ rather than column $k$ to eliminate in linear system with system matrix the unknown $x_j$ from all remaining equations but one.

# Pivoting ct.

All in all, we are free to choose any entry of $X_{k:m,k:m}$ as the pivot, as long as it is nonzero. The possibility that an entry $X_{kk} = 0$ might arise implies that some flexibility of choice of the pivot may sometimes be necessary, even from a pure mathematical point of view.

For numerical stability, however, it is desirable to pivot even when $x_{kk}$ is nonzero if there is a larger element available. In practice, it is common to pick as pivot the largest number among a set of entries being considered as candidates.

The structure of the elimination process quickly becomes confusing if zeros are introduced in arbitrary patterns through the matrix. To see what is going on, we want to retain the triangular structure, and there is an easy way to do this. We shall not think of the pivot $x_{ij}$ as left in place. Instead, at step k, we shall imagine that the rows and columns of the working matrix are permuted so as to move $x_{ij}$ into the $(k, k)$ position. Then, when the elimination is done, zeros are introduced into entries $k + 1, \ldots, m$ of column $k$, just as in Gaussian elimination without pivoting. This interchange of rows and perhaps columns is what is usually thought of as pivoting.

# Partial pivoting

If every entry of $X_{k:m,k:m}$ is considered as a possible pivot at step $k$, there are $(m-k)^2$ entries to be examined to determine the largest. This expensive strategy is called complete pivoting.

In practice, equally good pivots can be found by considering a much smaller number of entries. The standard method for doing this is partial pivoting. Here, only rows are interchanged.

The pivot at each step is chosen as the largest of the $m-k+1$ subdiagonal entries in column $k$. To bring the $k$th pivot into the $(k,k)$ position, no columns need to be permuted; only row $k$ is swapped with the row containing the pivot.

As usual in numerical linear algebra, this algorithm can be expressed as a matrix product. We saw in the last lecture that an elimination step corresponds to left-multiplication by an elementary lower-triangular matrix $L_k$. Partial pivoting complicates matters by applying a permutation matrix $P_k$ on the left of the working matrix before each elimination. After $m-1$ steps, $A$ becomes an upper-triangular matrix $U$:

$$L_{m-1}P_{m-1}L_{m-2}\cdots L_2P_2L_1P_1A = U.$$

# Example

$$A = \begin{pmatrix} 2 & 1 & 3 & 4 \\ -2 & 1 & -1 & -2 \\ 4 & 4 & 5 & 11 \\ -2 & 1 & -7 & -1 \end{pmatrix}$$

Since $|a_{31}| \geq |a_{j1}|$ for $j = 1, 2, 3, 4$ we interchange rows three and one:

$$P_1 A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 & 4 \\ -2 & 1 & -1 & -2 \\ 4 & 4 & 5 & 11 \\ -2 & 1 & -7 & -1 \end{pmatrix} = \begin{pmatrix} 4 & 4 & 5 & 11 \\ -2 & 1 & -1 & -2 \\ 2 & 1 & 3 & 4 \\ -2 & 1 & -7 & -1 \end{pmatrix}$$

Next we eliminate the subdiagonal elements of the first column

$$L_1 P_1 A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ -1/2 & 0 & 1 & 0 \\ 1/2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 & 4 & 5 & 11 \\ -2 & 1 & -1 & -2 \\ 2 & 1 & 3 & 4 \\ -2 & 1 & -7 & -1 \end{pmatrix} = \begin{pmatrix} 4 & 4 & 5 & 11 \\ 0 & 3 & 3/2 & 7/2 \\ 0 & -1 & 1/2 & -3/2 \\ 0 & 3 & -9/2 & 9/2 \end{pmatrix}$$

# Example ct.

$|x_{22}|$ is already maximal in the second column; no permutation is necessary

$$L_2 L_1 P_1 A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1/3 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 & 4 & 5 & 11 \\ 0 & 3 & 3/2 & 7/2 \\ 0 & -1 & 1/2 & -3/2 \\ 0 & 3 & -9/2 & 9/2 \end{pmatrix} = \begin{pmatrix} 4 & 4 & 5 & 11 \\ 0 & 3 & 3/2 & 7/2 \\ 0 & 0 & 1 & -1/3 \\ 0 & 0 & -6 & 1 \end{pmatrix}$$

$P_3$ permutes rows four and three

$$P_3 L_2 L_1 P_1 A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 4 & 4 & 5 & 11 \\ 0 & 3 & 3/2 & 7/2 \\ 0 & 0 & 1 & -1/3 \\ 0 & 0 & -6 & 1 \end{pmatrix} = \begin{pmatrix} 4 & 4 & 5 & 11 \\ 0 & 3 & 3/2 & 7/2 \\ 0 & 0 & -6 & 1 \\ 0 & 0 & 1 & -1/3 \end{pmatrix}$$

And the final elimination step yields

$$P_3 L_2 L_1 P_1 A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/6 & 1 \end{pmatrix} \begin{pmatrix} 4 & 4 & 5 & 11 \\ 0 & 3 & 3/2 & 7/2 \\ 0 & 0 & -6 & 1 \\ 0 & 0 & 1 & -1/3 \end{pmatrix} = \begin{pmatrix} 4 & 4 & 5 & 11 \\ 0 & 3 & 3/2 & 7/2 \\ 0 & 0 & -6 & 1 \\ 0 & 0 & 0 & -1/6 \end{pmatrix}$$

# $PA = LU$

The matrix equation $L_{m-1}P_{m-1}L_{m-2}\cdots L_2P_2L_1P_1A = U$ in our example reads

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1/2 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 1/ & -1/3 & 1/6 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 & 4 \\ -2 & 1 & -1 & -2 \\ 4 & 4 & 5 & 11 \\ -2 & 1 & -7 & -1 \end{pmatrix} = \begin{pmatrix} 4 & 4 & 5 & 11 \\ 0 & 3 & 3/2 & 7/2 \\ 0 & 0 & -6 & 1 \\ 0 & 0 & 0 & -1/6 \end{pmatrix}$$

If we knew in advance the permutations of the rows which are performed in the course of the Gaussian elimination we could apply these permutations first (which corresponds to one matrix multiplication of $A$ by a permutation matrix $P$) and determine the LU factorization of $PA$ without pivoting:

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 & 4 \\ -2 & 1 & -1 & -2 \\ 4 & 4 & 5 & 11 \\ -2 & 1 & -7 & -1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & 0 \\ -1/1 & 1 & 1 & 0 \\ 1/ & -1/3 & -1/6 & 1 \end{pmatrix} \begin{pmatrix} 4 & 4 & 5 & 11 \\ 0 & 3 & 3/2 & 7/2 \\ 0 & 0 & -6 & 1 \\ 0 & 0 & 0 & -1/6 \end{pmatrix}$$

# $PA = LU$ ct.

The factorization $PA = LU$ can be determined in the course of the Gaussian elimination not knowing the permutations in advance.

Gaussian elimination generates the decomposition

$$L_{m-1}P_{m-1}L_{m-2}P_{m-2}L_{m-3}\cdots L_2P_2L_1P_1A = U.$$

The lefthand side can be rewritten as

$$\begin{aligned}
&L_{m-1}P_{m-1}L_{m-2}P_{m-1}^{-1}P_{m-1}P_{m-2}L_{m-3}\cdots L_2P_2L_1P_1A\\
&= L_{m-1}L'_{m-2}P_{m-1}P_{m-2}L_{m-3}\cdots L_2P_2L_1P_1A\\
&=: L_{m-1}L'_{m-2}P_{m-1}P_{m-2}L_{m-3}P_{m-2}^{-1}P_{m-1}^{-1}P_{m-1}P_{m-2}P_{m-3}L_{m-4}\cdots L_2P_2L_1P_1A\\
&=: L_{m-1}L'_{m-2}L'_{m-3}P_{m-1}P_{m-2}P_{m-3}L_{m-4}\cdots L_2P_2L_1P_1A = \cdots\\
&= (L_{m-1}L'_{m-2}\cdots L'_1)(P_{m-1}P_{m-2}\cdots P_1)A = U
\end{aligned}$$

with

$$L'_k = P_{m-1}P_{m-2}\cdots P_{k+1}L_kP_{k+1}^{-1}\cdots L_{m-2}^{-1}L_{m-1}^{-1},\ k = 1,\ldots,m-2.$$

# $PA = LU$ ct.

Multiplying $L_k$ by $P_{k+1}$ on the left exchanges rows $k+1$ and $\ell$ for some $\ell > k+1$, and multiplying by $P_{k+1}^{-1}$ on the right exchanges columns $k+1$ and $\ell$. Hence, $P_{k+1} L_k P_{k+1}^{-1}$ has the same structure as $L_k$, and this structure is kept when multiplying with further permutations $P_{k+2}, \ldots, P_{m-1}$ and their inverses on the left and right, respectively.

The matrices $L'_k$ are unit lower-triangular and easily invertible by negating the subdiagonal entries, just as in Gaussian elimination without pivoting.

Writing $L = (L_{m-1} L'_{m-2} \cdots L'_1)^{-1}$ and $P = P_{m-1} \cdots P_2 P_1$, we have

$$PA = LU.$$

# Gaussian elimination

## Gaussian elimination with partial pivoting

$U = A$, $L = I$, $P = I$
**for** k=1:m-1 **do**
  selct $i \geq k$ to maximize $|u_{ik}|$
  $u_{k,k:m} \leftrightarrow u_{i,k:m}$
  $\ell_{k,1:k-1} \leftrightarrow \ell_{i,1:k-1}$
  $p_{k,:} \leftrightarrow p_{i,:}$
  **for** j=k+1:m **do**
    $\ell_{jk} = u_{jk} / u_{kk}$
    $u_{j,k:m} = u_{j,k:m} - \ell_{jk} u_{k,k_m}$
  **end for**
**end for**