LU Matrix Factorization

We wish to factor the matrix A as $A = L\,U$ where L is a lower triangular matrix and U is an upper triangular matrix. One reason for doing this is to facilitate finding the solution to the vector equation $A\mathbf{x} = \mathbf{b}$.

```
> restart;
> with(LinearAlgebra):
```

We want to solve the following linear system. We assume Gaussian elimination can be performed on $A\mathbf{x} = \mathbf{b}$ without row changes.

```
> A:=Matrix(4,4,[[1,1,0,3],[2,1,-1,1],[3,-1,-1,2],[-1,2,3,-1]]);
```

$$A := \begin{bmatrix} 1 & 1 & 0 & 3 \\ 2 & 1 & -1 & 1 \\ 3 & -1 & -1 & 2 \\ -1 & 2 & 3 & -1 \end{bmatrix}$$

```
> b:=Vector(4,[4,1,-3,4]);
```

$$b := \begin{bmatrix} 4 \\ 1 \\ -3 \\ 4 \end{bmatrix}$$

We use the LUDecomposition command to do the factorization. The variables in the set on the left are the names we wish to give to the **permutation**, **lower triangular**, and **upper triangular** matrices.

```
> (P,L,U):=LUDecomposition(A);
```

$$P, L, U := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 4 & 1 & 0 \\ -1 & -3 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 & 3 \\ 0 & -1 & -1 & -5 \\ 0 & 0 & 3 & 13 \\ 0 & 0 & 0 & -13 \end{bmatrix}$$

Since the permutation matrix is the identity, we ignore it here.

```
> Equal(Multiply(L,U),A);
```

true

The matrix equation $A\mathbf{x} = \mathbf{b}$ now can be written as $L\,U\mathbf{x} = \mathbf{b}$. We let $U\mathbf{x} = \mathbf{y}$ and solve $L\mathbf{y} = \mathbf{b}$ for $\mathbf{y}$.

```
> B:=<L|b>;
```

$$B := \begin{bmatrix} 1 & 0 & 0 & 0 & 4 \\ 2 & 1 & 0 & 0 & 1 \\ 3 & 4 & 1 & 0 & -3 \\ -1 & -3 & 0 & 1 & 4 \end{bmatrix}$$

We use ForwardSubstitute to solve by forward substitution.

```
> y:=ForwardSubstitute(B);
```

$$y := \begin{bmatrix} 4 \\ -7 \\ 13 \\ -13 \end{bmatrix}$$

Finally, we solve $U\mathbf{x} = \mathbf{y}$ for $\mathbf{x}$.

```
> C:=<U|y>;
```

$$C := \begin{bmatrix} 1 & 1 & 0 & 3 & 4 \\ 0 & -1 & -1 & -5 & -7 \\ 0 & 0 & 3 & 13 & 13 \\ 0 & 0 & 0 & -13 & -13 \end{bmatrix}$$

```
> x:=BackwardSubstitute(C);
```

$$x := \begin{bmatrix} -1 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

Now suppose row interchanges are needed in the Gaussian elimination.

```
> A:=Matrix(4,4,[[1,1,-1,0],[1,1,4,3],[2,1,4,1],[1,-2,2,-2]]);
```

$$A := \begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & 1 & 4 & 3 \\ 2 & 1 & 4 & 1 \\ 1 & -2 & 2 & -2 \end{bmatrix}$$

```
> (P,L,U):=LUDecomposition(A);
```

$$P, L, U := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 3 & -3 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & -1 & 0 \\ 0 & -1 & 6 & 1 \\ 0 & 0 & 5 & 3 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

In the following, note the use of the "." for matrix multiplication.

```
> Equal(P.A,L.U);
```

true

If $A\mathbf{x} = \mathbf{b}$, then since $A = P^t LU$, we have $P^t LU\mathbf{x} = \mathbf{b}$. Then $L\ U\mathbf{x} = P\mathbf{b}$. Let $U\mathbf{x} = \mathbf{y}$ and solve $L\mathbf{y}=P\mathbf{b}$.

```
> B:=<L|P.b>;
```

$$B := \begin{bmatrix} 1 & 0 & 0 & 0 & 4 \\ 2 & 1 & 0 & 0 & -3 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 3 & -3 & 1 & 4 \end{bmatrix}$$

```
> y:=ForwardSubstitute(B);
```

$$y := \begin{bmatrix} 4 \\ -11 \\ -3 \\ 24 \end{bmatrix}$$

Finally, we solve U**x**=**y** for **x**.

```
> C:=<U|y>;
```

$$C := \begin{bmatrix} 1 & 1 & -1 & 0 & 4 \\ 0 & -1 & 6 & 1 & -11 \\ 0 & 0 & 5 & 3 & -3 \\ 0 & 0 & 0 & 4 & 24 \end{bmatrix}$$

```
> x:=BackwardSubstitute(C);
```

$$x := \begin{bmatrix} 8 \\ -\frac{41}{5} \\ -\frac{21}{5} \\ 6 \end{bmatrix}$$

We check our answer by using **rref**.

```
> ReducedRowEchelonForm(<A|b>);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 8 \\ 0 & 1 & 0 & 0 & -\frac{41}{5} \\ 0 & 0 & 1 & 0 & -\frac{21}{5} \\ 0 & 0 & 0 & 1 & 6 \end{bmatrix}$$

The Decomposition Process

How is it done? We start with **A** and use Gaussian elimination to get **U**.

```
> A;
```

$$\begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & 1 & 4 & 3 \\ 2 & 1 & 4 & 1 \\ 1 & -2 & 2 & -2 \end{bmatrix}$$

We do not need to change any rows in **A** for our first pivoting step, so we left multiply by $E_1 = I_4$.

```
> E1:=IdentityMatrix(4);
```

$$E1 := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> A1:=E1.A;
```

$$A1 := \begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & 1 & 4 & 3 \\ 2 & 1 & 4 & 1 \\ 1 & -2 & 2 & -2 \end{bmatrix}$$

We pivot on the (1,1) element.

```
> A2:=Pivot(A1,1,1,2..4);
```

$$A2 := \begin{bmatrix} 1 & 1 & -1 & 0 \\ 0 & 0 & 5 & 3 \\ 0 & -1 & 6 & 1 \\ 0 & -3 & 3 & -2 \end{bmatrix}$$

We form a **Gaussian transformation matrix** for the above operation by starting with the identity matrix and placing below it the numbers we multiplied the first row by to get the 0's in the first column.

```
> M1:=Matrix([[1,0,0,0],[-1,1,0,0],[-2,0,1,0],[-1,0,0,1]]);
```

$$M1 := \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

We can also do our pivoting by multiplying **A1** on the left by **M1**.

```
> A2:=M1.A1;
```

$$A2 := \begin{bmatrix} 1 & 1 & -1 & 0 \\ 0 & 0 & 5 & 3 \\ 0 & -1 & 6 & 1 \\ 0 & -3 & 3 & -2 \end{bmatrix}$$

We cannot pivot on the (2,2) element without interchanging rows. We interchange rows 2 and 3 by multiplying on the left by the permutation matrix.

```
> E2:=RowOperation(E1,[2,3]);
```

$$E2 := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> A3:=E2.A2;
```

$$A3 := \begin{bmatrix} 1 & 1 & -1 & 0 \\ 0 & -1 & 6 & 1 \\ 0 & 0 & 5 & 3 \\ 0 & -3 & 3 & -2 \end{bmatrix}$$

We form the new **Gaussian transformation matrix** and left multiply by it.

```
> M2:=Matrix([[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,-3,0,1]]);
```

$$M2 := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -3 & 0 & 1 \end{bmatrix}$$

```
> A4:=M2.A3;
```

$$A4 := \begin{bmatrix} 1 & 1 & -1 & 0 \\ 0 & -1 & 6 & 1 \\ 0 & 0 & 5 & 3 \\ 0 & 0 & -15 & -5 \end{bmatrix}$$

We can pivot on the (3,3) element, so we have no need to interchange rows andleft multiply by $E_3 = I_4$.

```
> E3:=IdentityMatrix(4);
```

$$E3 := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> A5:=E3.A4;
```

$$A5 := \begin{bmatrix} 1 & 1 & -1 & 0 \\ 0 & -1 & 6 & 1 \\ 0 & 0 & 5 & 3 \\ 0 & 0 & -15 & -5 \end{bmatrix}$$

We form the final **Gaussian transformation matrix** and left multiply by it.

```
> M3:=Matrix([[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,3,1]]);
```

$$M3 := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 3 & 1 \end{bmatrix}$$

```
> A6:=M3.A5;
```

$$A6 := \begin{bmatrix} 1 & 1 & -1 & 0 \\ 0 & -1 & 6 & 1 \\ 0 & 0 & 5 & 3 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

This is our upper matrix **U**.

```
> U:=A6;
```

$$U := \begin{bmatrix} 1 & 1 & -1 & 0 \\ 0 & -1 & 6 & 1 \\ 0 & 0 & 5 & 3 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

Our lower matrix **L** is the inverse of the product M3*M2*M1.

```
> L:=MatrixInverse(M3.M2.M1);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 1 & 3 & -3 & 1 \end{bmatrix}$$

The permutation matrix P is the transpose (or inverse) of the product E3*E2*E1.

```
> P:=Transpose(E3.E2.E1);
```

$$P := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We see **LU=PA**.

```
> Equal(L.U,P.A);
```

$$true$$

```
> L.U;
```

$$\begin{bmatrix} 1 & 1 & -1 & 0 \\ 2 & 1 & 4 & 1 \\ 1 & 1 & 4 & 3 \\ 1 & -2 & 2 & -2 \end{bmatrix}$$

```
> P.A;
```

$$\begin{bmatrix} 1 & 1 & -1 & 0 \\ 2 & 1 & 4 & 1 \\ 1 & 1 & 4 & 3 \\ 1 & -2 & 2 & -2 \end{bmatrix}$$

We will find the **LU Decomposition** for the problem above that requires row interchanges by using the MatrixDecomposition command from the Numerical Analysis package.

```
> with(Student[NumericalAnalysis]);
```

[*AbsoluteError*, *AdamsBashforth*, *AdamsBashforthMoulton*, *AdamsMoulton*, *AdaptiveQuadrature*, *AddPoint*, *ApproximateExactUpperBound*, *ApproximateValue*, *BackSubstitution*, *BasisFunctions*, *Bisection*, *CubicSpline*, *DataPoints*, *Distance*, *DividedDifferenceTable*, *Draw*, *Euler*, *EulerTutor*, *ExactValue*, *FalsePosition*, *FixedPointIteration*, *ForwardSubstitution*, *Function*, *InitialValueProblem*, *InitialValueProblemTutor*, *Interpolant*, *InterpolantRemainderTerm*, *IsConvergent*, *IsMatrixShape*, *IterativeApproximate*, *IterativeFormula*, *IterativeFormulaTutor*, *LeadingPrincipalSubmatrix*, *LinearSolve*, *LinearSystem*, *MatrixConvergence*, *MatrixDecomposition*, *MatrixDecompositionTutor*, *ModifiedNewton*, *NevilleTable*, *Newton*, *NumberOfSignificantDigits*, *PolynomialInterpolation*, *Quadrature*, *RateOfConvergence*, *RelativeError*, *RemainderTerm*, *Roots*, *RungeKutta*, *Secant*, *SpectralRadius*, *Steffensen*, *Taylor*, *TaylorPolynomial*, *UpperBoundOfRemainderTerm*, *VectorLimit*]

```
> A:=Matrix(4,4,[[1,1,-1,0],[1,1,4,3],[2,1,4,1],[1,-2,2,-2]]);
```

$$A := \begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & 1 & 4 & 3 \\ 2 & 1 & 4 & 1 \\ 1 & -2 & 2 & -2 \end{bmatrix}$$

In each of the steps below, we have, first, the **Gaussian transformation matrix**, then the **permutation matrix** for **partial pivoting**, then the matrix we are trying to use **partial pivoting** on, and, finally, the result of the **partial pivoting**. You can check the product by matrix multiplication.

```
> P,L,U:=MatrixDecomposition(A,method=PLU,showsteps=true);
```

M_1 . E_1 . A = A_1:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 \\ -\frac{1}{2} & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & 1 & 4 & 3 \\ 2 & 1 & 4 & 1 \\ 1 & -2 & 2 & -2 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 4 & 1 \\ 0 & \frac{1}{2} & 2 & \frac{5}{2} \\ 0 & \frac{1}{2} & -3 & -\frac{1}{2} \\ 0 & -\frac{5}{2} & 0 & -\frac{5}{2} \end{bmatrix}$$

M_2 . E_2 . A_1 = A_2:

$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&\frac{1}{5}&1&0\\0&\frac{1}{5}&0&1\end{bmatrix}\cdot\begin{bmatrix}1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0\end{bmatrix}\cdot\begin{bmatrix}2&1&4&1\\0&\frac{1}{2}&2&\frac{5}{2}\\0&\frac{1}{2}&-3&-\frac{1}{2}\\0&-\frac{5}{2}&0&-\frac{5}{2}\end{bmatrix}=\begin{bmatrix}2&1&4&1\\0&-\frac{5}{2}&0&-\frac{5}{2}\\0&0&-3&-1\\0&0&2&2\end{bmatrix}$$

```
M_3 . E_3 . A_2 = U:
```

$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&\frac{2}{3}&1\end{bmatrix}\cdot\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\cdot\begin{bmatrix}2&1&4&1\\0&-\frac{5}{2}&0&-\frac{5}{2}\\0&0&-3&-1\\0&0&2&2\end{bmatrix}=\begin{bmatrix}2&1&4&1\\0&-\frac{5}{2}&0&-\frac{5}{2}\\0&0&-3&-1\\0&0&0&\frac{4}{3}\end{bmatrix}$$

$$P, L, U := \begin{bmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{bmatrix}, \begin{bmatrix}1&0&0&0\\\frac{1}{2}&1&0&0\\\frac{1}{2}&-\frac{1}{5}&1&0\\\frac{1}{2}&-\frac{1}{5}&-\frac{2}{3}&1\end{bmatrix}, \begin{bmatrix}2&1&4&1\\0&-\frac{5}{2}&0&-\frac{5}{2}\\0&0&-3&-1\\0&0&0&\frac{4}{3}\end{bmatrix}$$

We now drop the **showsteps=true** option and add **output=Mmatrices**, which lists the **Gaussian transformation** matrices in order.

```
> MM:=MatrixDecomposition(A,method=PLU,output=Mmatrices);
```

$$MM := \left[\begin{bmatrix}1&0&0&0\\-\frac{1}{2}&1&0&0\\-\frac{1}{2}&0&1&0\\-\frac{1}{2}&0&0&1\end{bmatrix}, \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&\frac{1}{5}&1&0\\0&\frac{1}{5}&0&1\end{bmatrix}, \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&\frac{2}{3}&1\end{bmatrix}\right]$$

L is then the inverse of the product of the Gaussian transformation matrices in reverse order.

```
> LL:=MatrixInverse(MM[3].MM[2].MM[1]);
```

$$LL := \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & -\frac{1}{5} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{5} & -\frac{2}{3} & 1 \end{bmatrix}$$

We now change to **output=Ematrices**, which lists the permutation matrices in order.

```
> EE:=MatrixDecomposition(A,method=PLU,output=Ematrices);
```

$$EE := \left[\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right]$$

P is then the inverse of the product of these matrices in reverse order.

```
> PP:=MatrixInverse(EE[3].EE[2].EE[1]);
```

$$PP := \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

U is simply the matrix resulting from Gaussian elimination.

We see **LU=PA**.

```
> Equal(LL.U,PP.A);
```

true

```
> LL.U;
```

$$\begin{bmatrix} 2 & 1 & 4 & 1 \\ 1 & -2 & 2 & -2 \\ 1 & 1 & -1 & 0 \\ 1 & 1 & 4 & 3 \end{bmatrix}$$

```
> PP.A;
```

$$\begin{bmatrix} 2 & 1 & 4 & 1 \\ 1 & -2 & 2 & -2 \\ 1 & 1 & -1 & 0 \\ 1 & 1 & 4 & 3 \end{bmatrix}$$

We can also see that factorizations are not unique.