Beispiele mit MuPAD

```
>> mupad

   *----*     MuPAD 1.4.1 -- Multi Processing Algebra Data Tool
  /|   /|
 *----* |     Copyright (c)  1997 - 98 by SciFace Software GmbH
 | *--|-*                    All rights reserved.
 |/   |/
 *----*       Licensed to:   FB17-Mathematik-Uni-PB

 Type ?demo, ?topics, ?news or ?changes for further information.
```

Einlesen der linalg-Bibliothek:

```
>> export(linalg);
```

Informationen zu dieser Bibliothek gibt es mit

```
>> ?linalg;
```

Definition des Matrizenbereiches (hier über dem Körper $\mathbb{Q}$):

```
>> matrix:=Dom::Matrix(Dom::Rational);

                    Dom::Matrix(Dom::Rational)
```

Im folgenden wird ein lineares Gleichungssystem $Ax = b$ gelöst.

```
>> A:=matrix([[1,1,5/2,0,1],[0,1,1/2,1,3/2],[2,1,9/2,0,1],[1,1,5/2,0,1]]);

                     +-                     -+
                     |  1, 1, 5/2, 0,   1    |
                     |                       |
                     |  0, 1, 1/2, 1, 3/2    |
                     |                       |
                     |  2, 1, 9/2, 0,   1    |
                     |                       |
                     |  1, 1, 5/2, 0,   1    |
                     +-                     -+
>> b:=matrix([-1/2,1,0,-1/2]);

                            +-       -+
                            |  -1/2   |
                            |         |
                            |    1    |
                            |         |
                            |    0    |
                            |         |
                            |  -1/2   |
                            +-       -+
```

Wir berechnen zur Illustration die Treppennormalform der erweiterten Koeffizientenmatrix $B = (A \mid b)$.

```
>> B:=concatMatrix(A,b);
```

```
+-                            -+
|  1, 1, 5/2, 0,  1,  -1/2  |
|                            |
|  0, 1, 1/2, 1, 3/2,    1   |
|                            |
|  2, 1, 9/2, 0,  1,    0    |
|                            |
|  1, 1, 5/2, 0,  1,  -1/2  |
+-                            -+
```

```
>> gaussJordan(B);
```

```
+-                          -+
|  1, 0,  2,  0,  0,  1/2  |
|                            |
|  0, 1, 1/2, 0,  1,   -1  |
|                            |
|  0, 0,  0,  1, 1/2,  2   |
|                            |
|  0, 0,  0,  0,  0,   0   |
+-                          -+
```

```
>> linearSolve(A,b);
```

```
-- +-      -+  -- +-       -+  +-       -+ -- --
|  |  1/2   |  |  |   -2    |  |    0    |  |  |
|  |        |  |  |         |  |         |  |  |
|  |   -1   |  |  |  -1/2   |  |   -1    |  |  |
|  |        |  |  |         |  |         |  |  |
|  |   0    |, |  |    1    |, |    0    |  |  |
|  |        |  |  |         |  |         |  |  |
|  |   2    |  |  |    0    |  |  -1/2   |  |  |
|  |        |  |  |         |  |         |  |  |
|  |   0    |  |  |    0    |  |    1    |  |  |
-- +-      -+  -- +-       -+  +-       -+ -- --
```

Der erste Eintrag in der Liste ist eine spezielle Lösung des lineare Gleichungssystem $Ax = b$, danach folgt eine Basis des Lösungsraums des homogenen Systems $Ax = 0$. Ein weiteres Beispiel:

```
>> b:=matrix([1,2,3,4]);
```

```
+-   -+
|  1  |
|     |
|  2  |
|     |
|  3  |
|     |
|  4  |
+-   -+
```

```
>> linearSolve(A,b);
```

```
                                    []
```

Das bedeutet, daß das lineare Gleichungssystem $Ax = b$ keine Lösung hat, was man anhand des Kriteriums aus der Vorlesung sehen kann:

```
>> B:=concatMatrix(A,b);

                +-                       -+
                |  1, 1, 5/2, 0,  1,  1   |
                |                         |
                |  0, 1, 1/2, 1, 3/2, 2   |
                |                         |
                |  2, 1, 9/2, 0,  1,  3   |
                |                         |
                |  1, 1, 5/2, 0,  1,  4   |
                +-                       -+

>> gaussJordan(B);

              +-                        -+
              |  1, 0,  2,  0,  0,  0    |
              |                          |
              |  0, 1, 1/2, 0,  1,  0    |
              |                          |
              |  0, 0,  0,  1, 1/2, 0    |
              |                          |
              |  0, 0,  0,  0,  0,  1    |
              +-                        -+
```

Also ist $\mathrm{Rg}(A) < \mathrm{Rg}(B)$:

```
>> rank(A); rank(B);
                              3

                              4
```

Weitere Befehle. Zunächst die transponierte Matrix zu A:

```
>> transpose(A);

              +-                       -+
              |   1,    0,    2,    1   |
              |                         |
              |   1,    1,    1,    1   |
              |                         |
              |  5/2, 1/2,  9/2,  5/2   |
              |                         |
              |   0,    1,    0,    0   |
              |                         |
              |   1,   3/2,   1,    1   |
              +-                       -+
```

Multiplikation mit Skalaren:

```
>> 4*A;
```

```
+-                -+
|  4, 4, 10, 0, 4  |
|                  |
|  0, 4,  2, 4, 6  |
|                  |
|  8, 4, 18, 0, 4  |
|                  |
|  4, 4, 10, 0, 4  |
+-                -+
```

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

```
+-                -+
|   1, 2,  3,  4   |
|                  |
|  -1, 2,  3,  4   |
|                  |
|   1, 2, -3,  4   |
|                  |
|  -1, 2,  3, -4   |
+-                -+
```

Die Adjunkte von A:

```
>> adjoint(A);
```

```
+-                     -+
|  96, -96,  0,    0    |
|                       |
|   0,  0,   48,  48    |
|                       |
|  32,  0,  -32,  0     |
|                       |
|   0,  24,  0,  -24    |
+-                     -+
```

```
>> A*adjoint(A);
```

```
+-                       -+
|  192,  0,    0,    0    |
|                         |
|   0,  192,  0,    0     |
|                         |
|   0,   0,  192,  0      |
|                         |
|   0,   0,    0,  192    |
+-                       -+
```

Die Determinante von A:

```
>> det(A);
```

```
192
```

Die Inverse von A:

```
>> A^(-1);
```

```
+-                         -+
|  1/2, -1/2,    0,     0   |
|                           |
|   0,     0,   1/4,  1/4   |
|                           |
|  1/6,    0,  -1/6,    0   |
|                           |
|   0,   1/8,    0,  -1/8   |
+-                         -+
```

Wir berechnen die LU-Zerlegung einer Matrix. Mit MuPAD funktioniert dies nur für quadratische Matrizen.

```
>> factorLU(A);
```

```
+-                 -+
|    1, 2,   3,   4  |
|                   |
|   -1, 4,   6,   8  |  +-           -+
|                   |,  | 1, 2, 3, 4 |, [-8, -6, 4, 1]
|    1, 0, -6,   0  |  +-           -+
|                   |
|   -1, 1,   0, -8  |
+-                 -+
```

In der linken Matrix der Ausgabe befinden sich die Matrizen L und U (s.u.), der mittleren Zeile entnimmt man, welche Permutationen vorgenommen wurden (hier: keine), die rechte Zeile enthält die sogenannten Pivot-Elemente.

```
>> L:=matrix([[1,0,0,0],[-1,1,0,0],[1,0,1,0],[-1,1,0,1]]);
```

```
+-              -+
|    1, 0, 0, 0  |
|                |
|   -1, 1, 0, 0  |
|                |
|    1, 0, 1, 0  |
|                |
|   -1, 1, 0, 1  |
+-              -+
```

```
>> U:=matrix([[1,2,3,4],[0,4,6,8],[0,0,-6,0],[0,0,0,-8]]);
```

```
+-                -+
|  1, 2,   3,   4  |
|                  |
|  0, 4,   6,   8  |
|                  |
|  0, 0,  -6,   0  |
|                  |
|  0, 0,   0,  -8  |
+-                -+
```

Probe: Es ist $A = LU$.

```
>> L*U;
```

```
+-               -+
|   1, 2,  3,  4  |
|                 |
|  -1, 2,  3,  4  |
|                 |
|   1, 2, -3,  4  |
|                 |
|  -1, 2,  3, -4  |
+-               -+
```

Wir wechseln nun den Grundring und nehmen $\mathbb{Z}/6\mathbb{Z}$:

```
>> matrix:=Dom::Matrix(Dom::IntegerMod(6));

                  Dom::Matrix(Dom::IntegerMod(6))


>> A:=matrix([[5,1,0,3,0],[3,2,0,1,0],[0,3,1,0,4],[0,1,2,3,5],[4,0,1,0,3]]);

         +-                                                -+
         |  5 mod 6, 1 mod 6, 0 mod 6, 3 mod 6, 0 mod 6  |
         |                                                  |
         |  3 mod 6, 2 mod 6, 0 mod 6, 1 mod 6, 0 mod 6  |
         |                                                  |
         |  0 mod 6, 3 mod 6, 1 mod 6, 0 mod 6, 4 mod 6  |
         |                                                  |
         |  0 mod 6, 1 mod 6, 2 mod 6, 3 mod 6, 5 mod 6  |
         |                                                  |
         |  4 mod 6, 0 mod 6, 1 mod 6, 0 mod 6, 3 mod 6  |
         +-                                                -+

>> det(A);

                              5 mod 6
```

A ist also invertierbar. Zur Berechnung der Inversen funktioniert folgendes aber nicht:

```
>> A^(-1);

                                FAIL
```

Daher nutzt man die Theorie aus:

```
>> B:=det(A)^(-1)*adjoint(A);

         +-                                                -+
         |  2 mod 6, 3 mod 6, 1 mod 6, 1 mod 6, 3 mod 6  |
         |                                                  |
         |  3 mod 6, 3 mod 6, 4 mod 6, 4 mod 6, 0 mod 6  |
         |                                                  |
         |  1 mod 6, 3 mod 6, 5 mod 6, 2 mod 6, 4 mod 6  |
         |                                                  |
         |  0 mod 6, 4 mod 6, 1 mod 6, 1 mod 6, 3 mod 6  |
         |                                                  |
         |  5 mod 6, 3 mod 6, 5 mod 6, 4 mod 6, 5 mod 6  |
         +-                                                -+
```

Probe: es ist $AB = E_5$.

```
>> A*B;
```

```
+-                                                 -+
|  1 mod 6, 0 mod 6, 0 mod 6, 0 mod 6, 0 mod 6  |
|                                                   |
|  0 mod 6, 1 mod 6, 0 mod 6, 0 mod 6, 0 mod 6  |
|                                                   |
|  0 mod 6, 0 mod 6, 1 mod 6, 0 mod 6, 0 mod 6  |
|                                                   |
|  0 mod 6, 0 mod 6, 0 mod 6, 1 mod 6, 0 mod 6  |
|                                                   |
|  0 mod 6, 0 mod 6, 0 mod 6, 0 mod 6, 1 mod 6  |
+-                                                 -+
```

Man kann mit MuPAD auch symbolisch rechnen. Ein Beispiel:

```
>> matrix:=Dom::Matrix(Dom::ExpressionField(normal,iszero));

            Dom::Matrix(Dom::ExpressionField(normal, iszero))


>> A:=matrix([[a,b],[c,d]]);

                              +-      -+
                              |  a, b  |
                              |        |
                              |  c, d  |
                              +-      -+


>> A^(-1);

                    +-                          -+
                    |      d             b       |
                    |   ---------,  - ---------  |
                    |   a d - b c     a d - b c  |
                    |                            |
                    |       c           a        |
                    |  - ---------,  ---------   |
                    |    a d - b c   a d - b c   |
                    +-                          -+


>> det(A);

                               a d - b c
```

Man beachte, daß hierbei die Parameter a, b, c, d nicht vorher schon definiert wurden. Zum Abschluß ein Beispiel aus einem anderen Bereich:
Den größten gemeinsamen Teiler d zweier ganzer Zahlen a und b berechnet `igcd`. `igcdex` berechnet außerdem noch x und y mit $d = xa + yb$:

```
>> igcd(256003,231);

                                     11

>> igcdex(256003,231);

                                 11, -4, 4433
```