Computergestuetzte Mathematik (Lineare Algebra mit Maple)

Lektion 7 (3. Dez.)

▼ Vektoren und Matrizen

```
> x := <1,2,3>;
> y := <4|5|6>;
> A := << 1 | 2 | 3 >,
        < 4 | 5 | 6 >,
        < 7 | 8 | 9 >>;

> about(A);
> about(x);
> AA := Matrix(3, 3, [[1, 2, 3], [4, 5, 6], [7, 8, 9]]); #
  alternative Eingabe
> xx := Vector(3,[1,2,3]);
> A*x;
> A . x;
> B := < < 1, 4, 7 > | < 2, 5, 8 > | < 3 , 6 , 9 >>;
> A - B;
> A *~ B;
> A^2;
> < A | B >;
> < A , B >;
> Id := A^0; #Achtung
> about(Id);
> C := A + 1;
> C^(-1);
> % . (A + 1);
> (A + 1) . ??;
> Matrix(3, shape = identity);
> Matrix( <1,2,3>, shape = diagonal);
> circ := Matrix(4, (i,j) -> (i-j)^3);
> circ^(-1);
> hilbert := Matrix(4, (i,j) -> 1/(i+j-1));
> with(LinearAlgebra);
> Transpose(A);
> B := A;
> B[1,2] := 222;
```

```
> B;
> A;
> A[1,2] := 2;
> B := Copy(A);
> B[1,2] := 777;
> B;
> A;
```

Lineare Gleichungssysteme

```
> restart:
> g1 := x + y - z = 1;
> g2 := 2*x + y - 3*z = 0;
> g3 := x - 2*z = -1;
> solve({g1, g2, g3}, {x,y,z});
> subs(%, {g1, g2 , g3});
> with(LinearAlgebra):
> B := GenerateMatrix( [g1, g2, g3], [x, y, z], augmented = true)
 ;
> A := SubMatrix(B, 1..3, 1..3);
> SubMatrix(B, 1..3, 4..4);
> whattype(%);
> b := convert(??, Vector);
> whattype(%);
> x := LinearSolve(A, b);
> A . x;
> ReducedRowEchelonForm(B);
```

Zeilenweise Manipulation

```
> B;
> A1 := RowOperation(B, [2,1], -2);
> A2 := RowOperation(A1, [3,1], -1);
> A3 := RowOperation(A2, [3,2], -1);
> A4 := RowOperation(A3, [1,2], 1);
> A5 := RowOperation(A4, 2, -1);
> B
```

Rang und Determinante

```
> Rank(B)
> Determinant(A);
```

Normalformen

```
> A; Eigenvalues(A);
> ew,T := Eigenvectors(A);
> J := Matrix(ew, shape=diagonal);
> T . J . T^(-1);
> M := << -14 |   -18 |     3 |    11 |   -1 |    16>,
         < -28 |   -36 |    18 |    24 |   -6 |    40>,
         <-134 |  -182 |    90 |   126 |  -16 |   198>,
         < -12 |   -12 |     2 |    10 |   -2 |     8>,
         < 190 |   254 |  -126 |  -178 |   24 |  -278>,
         <  46 |    62 |   -32 |   -46 |    4|   -66>>;
> J, T := JordanForm(M, output = ['J', 'Q']);
> T . J . T^(-1) - M;
```

Andere Operationen mit Matrizen

```
> restart:
> with(LinearAlgebra):
> v := Vector(3, symbol = x, orientation = column);
> w := Vector(3, symbol=y, orientation = column);
> w[1];
> v . w;
> v . w assuming real;
> CrossProduct(v, w);
> VectorNorm(v);
> VectorNorm(v, 2);
> with(plots):
> A := <<3, 1, 2>|<5, 2, 1>>;
> map( x -> x^2, A);
> B:=map( x -> sin(x/2), A);
> matrixplot(evalf(B), heights = histogram,gap=0.1,axes=frame,
  orientation=[-20,60]);

>
```