Sage4-Vorlesung-Live

```
3+4
```

```
7
```

```
M=matrix([[1,2],[3,4]])
show(M)
```

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
N=matrix(3,2,[1,2,3,4,5,6])
show(N)
```

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

```
A=matrix(2,2,[1,x,0,2])
A
```

```
[1 x]
[0 2]
```

```
N[1,0]
```

```
3
```

```
show(parent(A))
```

$$\mathrm{Mat}_{2\times 2}(\mathrm{SR})$$

```
A+M
```

```
[     2 x + 2]
[     3     6]
```

```
A*M
```

```
[3*x + 1 4*x + 2]
[      6       8]
```

```
identity_matrix(10)
```

```
[1 0 0 0 0 0 0 0 0 0]
[0 1 0 0 0 0 0 0 0 0]
[0 0 1 0 0 0 0 0 0 0]
[0 0 0 1 0 0 0 0 0 0]
[0 0 0 0 1 0 0 0 0 0]
[0 0 0 0 0 1 0 0 0 0]
[0 0 0 0 0 0 1 0 0 0]
[0 0 0 0 0 0 0 1 0 0]
[0 0 0 0 0 0 0 0 1 0]
```

```
[0 0 0 0 0 0 0 0 0 1]
```

```
zero_matrix(2,5)
```

```
[0 0 0 0 0]
[0 0 0 0 0]
```

```
B=matrix(20,20,{(0,0):1,(3,4):-4})
show(B)
```

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

```
A
```

```
[1 x]
[0 2]
```

```
A[1]
```

```
(0, 2)
```

```
A.row(1)
```

```
(0, 2)
```

```
A.column(1)
```

```
(x, 2)
```

```
M
```

```
[1 2]
[3 4]
```

```
M[1,1]=42
```

```
M
```

```
[ 1  2]
[ 3 42]
```

```
H=matrix(6,6,lambda i,j: 1/(i+j+1))
H
```

```
[   1  1/2  1/3  1/4  1/5  1/6]
[ 1/2  1/3  1/4  1/5  1/6  1/7]
[ 1/3  1/4  1/5  1/6  1/7  1/8]
[ 1/4  1/5  1/6  1/7  1/8  1/9]
[ 1/5  1/6  1/7  1/8  1/9 1/10]
[ 1/6  1/7  1/8  1/9 1/10 1/11]
```

```
H1=matrix(6,6,lambda i,j: i+j)
H1
```

```
[ 0  1  2  3  4  5]
[ 1  2  3  4  5  6]
[ 2  3  4  5  6  7]
[ 3  4  5  6  7  8]
[ 4  5  6  7  8  9]
[ 5  6  7  8  9 10]
```

```
b=matrix(6,1,lambda i,j: 1/(i+1)^2)
b
```

```
[   1]
[ 1/4]
[ 1/9]
[1/16]
[1/25]
[1/36]
```

```
H^(-1)*b
```

```
[ 71/15]
[   -35]
[   140]
[  -280]
[ 525/2]
[-462/5]
```

```
H\b
```

```
[ 71/15]
[   -35]
[   140]
[  -280]
[ 525/2]
[-462/5]
```

```
H.inverse()
```

```
[      36     -630     3360    -7560     7560    -2772]
[    -630    14700   -88200   211680  -220500    83160]
[    3360   -88200   564480 -1411200  1512000  -582120]
[   -7560   211680 -1411200  3628800 -3969000  1552320]
[    7560  -220500  1512000 -3969000  4410000 -1746360]
```

```
[    -2772     83160   -582120   1552320 -1746360    698544]
```

```
H.inverse()*b
```

```
[ 71/15]
[   -35]
[   140]
[  -280]
[ 525/2]
[-462/5]
```

```
H.solve_right(b)
```

```
[ 71/15]
[   -35]
[   140]
[  -280]
[ 525/2]
[-462/5]
```

```
A=matrix([[1,2,3],[2,4,5],[0,0,1]])
A
```

```
[1 2 3]
[2 4 5]
[0 0 1]
```

```
rank(A)
```

```
2
```

```
A \ vector([1,2,0])
```

```
(1, 0, 0)
```

```
A.right_kernel()
```

```
Free module of degree 3 and rank 1 over Integer Ring
Echelon basis matrix:
[ 2 -1  0]
```

```
V=QQ^5
V
```

```
Vector space of dimension 5 over Rational Field
```

```
v1=V([1,1,1,0,0])
v2=V([1,-1,1,0,0])
v3=V([1,0,1,0,0])
v1
```

```
(1, 1, 1, 0, 0)
```

```
show(parent(v1))
```

$\mathbf{Q}^5$

```
U=V.subspace([v1,v2,v3])
```

```
U
```

```
Vector space of degree 5 and dimension 2 over Rational Field
Basis matrix:
[1 0 1 0 0]
[0 1 0 0 0]
```

```
U.basis()
```

```
[
(1, 0, 1, 0, 0),
(0, 1, 0, 0, 0)
]
```

```
v4=V([0,0,0,1,0])
v4
```

```
(0, 0, 0, 1, 0)
```

```
v4 in U
```

```
False
```

```
v1 in U
```

```
True
```

```
W=V.subspace([V(v4)])
W
```

```
Vector space of degree 5 and dimension 1 over Rational Field
Basis matrix:
[0 0 0 1 0]
```

Ist $\dim(U) \leq \dim(V)$?

```
U <= V
```

```
True
```

```
W <=U
```

```
True
```

$U + W$ ist der kleinste Untervektorraum von V, welcher U und W enthält.

```
U+W
```

```
Vector space of degree 5 and dimension 3 over Rational Field
Basis matrix:
[1 0 1 0 0]
[0 1 0 0 0]
[0 0 0 1 0]
```

```
U.intersection(W)
```

```
Vector space of degree 5 and dimension 0 over Rational Field
Basis matrix:
[]
```

```
Qx=QQ[x]
```

```
x=Qx(x)
```

```
pp=[1+x+x^2,1-x+x^2,x^3-x,2+x^3]
show(pp)
```

$[x^2 + x + 1, x^2 - x + 1, x^3 - x, x^3 + 2]$

```
p0=pp[0]
p0
```

```
x^2 + x + 1
```

```
pp[3].coefficients()
```

```
[2, 1]
```

```
kv0=[p.coefficients(sparse=False) for p in pp]
kv0
```

```
[[1, 1, 1], [1, -1, 1], [0, -1, 0, 1], [2, 0, 0, 1]]
```

```
n=max([p.degree() for p in pp]) +1
n
```

```
4
```

```
def verlaengern(v,n):
    for i in range(len(v),n):
        v.append(0)
    return v
```

```
kv=[verlaengern(v,n) for v in kv0]
kv
```

```
[[1, 1, 1, 0], [1, -1, 1, 0], [0, -1, 0, 1], [2, 0, 0, 1]]
```

```
kv0
```

```
[[1, 1, 1, 0], [1, -1, 1, 0], [0, -1, 0, 1], [2, 0, 0, 1]]
```

```
def linearunabhaengig(pp):
    n=max([p.degree() for p in pp]) +1
    kv0=[p.coefficients(sparse=False) for p in pp]
    kv=[verlaengern(v,n) for v in kv0]
    r=rank(matrix(kv))
    return (r==len(pp))
```

```
linearunabhaengig(pp)
```

```
True
```

```
g=(1..4)
g
```

```
<generator object at 0x15df935f0>
```

```
g.next()
```

```
1
```

```
g.next()
```

```
2
```

```
g.next()
```

```
3
```

```
g.next()
```

```
4
```

```
g.next()
```

```
Traceback (click to the left of this block for traceback)
...
StopIteration
```

```
g=(1..4)
```

```
g.next()
```

```
1
```

```
list(g)
```

```
[2, 3, 4]
```

```
g.next()
```

```
Traceback (click to the left of this block for traceback)
...
StopIteration
```

```
g=(1..4)
for i in g:
    print i
```

```
1
2
3
4
```

```
g.next()
```

```
Traceback (click to the left of this block for traceback)
...
StopIteration
```

```
nn=(1..)
nn.next()
```

```
1
```

```
nn.next()
```

```
2
```

```
g=(1..4)
g2=(i^2 for i in g)
```

```
g2
```

```
<generator object <genexpr> at 0x16008a4b0>
```

```
g2.next()
```

```
1
```

```
g2.next()
```

```
4
```

```
g2.next()
```

```
9
```

```
g2.next()
```

```
16
```

```
g2.next()
```

```
Traceback (click to the left of this block for traceback)
...
StopIteration
```

```
def abc():
    yield 'a'
    yield 'b'
    yield 'c'
```

```
g=abc()
g
```

```
<generator object abc at 0x16008a730>
```

```
g.next()
```

```
'a'
```

```
g.next()
```

```
'b'
```

```
g.next()
```

```
'c'
```

```
g.next()
```

```
Traceback (click to the left of this block for traceback)
...
StopIteration
```

```
g=abc()
```

```
g.next()
```

```
'a'
```

```
def abc(cnicht):
    yield 'a'
    yield 'b'
```

```
    if cnicht:
        return
    yield 'c'
```

```
g=abc(True)
```

```
g.next()
```

    'a'

```
g.next()
```

    'b'

```
g.next()
```

    Traceback (click to the left of this block for traceback)
    ...
    StopIteration

```
g=abc(False)
```

```
g=abc(False)
```

```
g.next()
```

    'a'

```
g.next()
```

    'b'

```
g.next()
```

    'c'

```
def primzw(n=oo):
    p=3
    while p<n:
        q=next_prime(p)
        if q==p+2:
            yield [p,q]
        p=q
```

```
g=primzw()
```

```
g.next()
```

    [3, 5]

```
g.next()
```

    [5, 7]

```
g.next()
```

```
[11, 13]
```

```
g.next()
```

```
[17, 19]
```

```
def cart(S,n):
    if n==1:
        for s in S:
            yield [s]
    else:
        for s in S:
            for p in cart(S,n-1):
                yield [s]+p
```

```
list(cart([1,2],4))
```

```
[[1, 1, 1, 1],
 [1, 1, 1, 2],
 [1, 1, 2, 1],
 [1, 1, 2, 2],
 [1, 2, 1, 1],
 [1, 2, 1, 2],
 [1, 2, 2, 1],
 [1, 2, 2, 2],
 [2, 1, 1, 1],
 [2, 1, 1, 2],
 [2, 1, 2, 1],
 [2, 1, 2, 2],
 [2, 2, 1, 1],
 [2, 2, 1, 2],
 [2, 2, 2, 1],
 [2, 2, 2, 2]]
```