An Introduction to Linear Algebra

```
info(linalg)
```

```
Library 'linalg': the linear algebra package

-- Interface:
linalg::addCol,       linalg::addRow,           linalg::adjoint,
linalg::angle,        linalg::basis,            linalg::charmat,
linalg::charpoly,     linalg::col,              linalg::companion,
linalg::concatMatrix, linalg::cond,             linalg::crossProduct,
linalg::curl,         linalg::delCol,           linalg::delRow,
linalg::det,          linalg::divergence,       linalg::eigenvalues,
linalg::eigenvectors, linalg::expr2Matrix,      linalg::factorCholesky,
linalg::factorLU,     linalg::factorQR,         linalg::frobeniusForm,
linalg::gaussElim,    linalg::gaussJordan,      linalg::grad,
linalg::gradient,     linalg::hermiteForm,      linalg::hessenberg,
linalg::hessian,      linalg::hilbert,          linalg::htranspose,
linalg::intBasis,     linalg::inverseLU,        linalg::invhilbert,
linalg::invpascal,    linalg::invvandermonde,   linalg::isHermitian,
linalg::isPosDef,     linalg::isUnitary,        linalg::jacobian,
linalg::jordanForm,   linalg::kroneckerProduct, linalg::laplacian,
linalg::matdim,       linalg::matlinsolve,      linalg::matlinsolveLU,
linalg::minpoly,      linalg::multCol,          linalg::multRow,
linalg::ncols,        linalg::nonZeros,         linalg::normalize,
linalg::nrows,        linalg::nullspace,        linalg::ogCoordTab,
linalg::orthog,       linalg::pascal,           linalg::permanent,
linalg::potential,    linalg::pseudoInverse,    linalg::randomMatrix,
linalg::rank,         linalg::row,              linalg::scalarProduct,
linalg::setCol,       linalg::setRow,           linalg::smithForm,
linalg::sqrtMatrix,   linalg::stackMatrix,      linalg::submatrix,
linalg::substitute,   linalg::sumBasis,         linalg::swapCol,
linalg::swapRow,      linalg::sylvester,        linalg::toeplitz,
linalg::toeplitzSolve, linalg::tr,               linalg::transpose,
linalg::vandermonde,  linalg::vandermondeSolve, linalg::vecdim,
linalg::vectorOf,     linalg::vectorPotential,  linalg::wiedemann,
```

Systems of linear equations

```
reset():
```

```
equations := {
   x - 2*y - 3*z + t = 7,
   x + y + z + t = 1,
   5*x - 3*y - 3*z = 2
}
```

$\{5\,x-3\,y-3\,z=2,\ t+x-2\,y-3\,z=7,\ t+x+y+z=1\}$

```
solve(equations, {x,y,z})
```

$\left\{\left[x=\frac{5}{8}-\frac{3\,t}{8},\ y=\frac{15}{2}-\frac{5\,t}{2},\ z=\frac{15\,t}{8}-\frac{57}{8}\right]\right\}$

```
linsolve(equations, {x,y,z})
```

$\left[x=\frac{5}{8}-\frac{3\,t}{8},\ y=\frac{15}{2}-\frac{5\,t}{2},\ z=\frac{15\,t}{8}-\frac{57}{8}\right]$

```
linsolve(equations, [z,y,x])
```

$\left[z=\frac{15\,t}{8}-\frac{57}{8},\ y=\frac{15}{2}-\frac{5\,t}{2},\ x=\frac{5}{8}-\frac{3\,t}{8}\right]$

```
linsolve(
   {5*cos(x)+3*exp(x) = 1, cos(x)-2*exp(x) = 0},
   {cos(x), exp(x)}
)
```

$\left[\cos(x)=\frac{2}{13},\ e^x=\frac{1}{13}\right]$

```
linsolve({5*x+3*y = 1, -2*x - 5*y = 0}, {x,y},
   Domain = Dom::IntegerMod(7)
)
```

$[x=1 \bmod 7,\ y=1 \bmod 7]$

```
eq1 := x + 2*z = 1:
eq2 := y + 4*z = 7:
eq3 := 6*x + y = 1:
```

```
linsolve({eq1,eq2,eq3}, {x,y,z})
```

$\left[x=-\frac{1}{2},\ y=4,\ z=\frac{3}{4}\right]$

```
// finding a point on each plane
solve(subs(eq1, x=0, y=0), z)
```

$\left\{\frac{1}{2}\right\}$

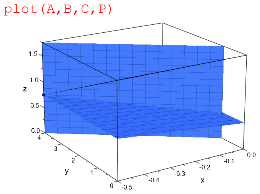
```
solve(subs(eq2,x=0, y=0),z)
```

$\left\{\frac{7}{4}\right\}$

```
solve(subs(eq3,x=0, z=0),y)
```

$\{1\}$

```
// declare planes and the point
A := plot::Plane([0,0,1/2],[1,0,2]):
B := plot::Plane([0,0,7/4],[0,1,4]):
C := plot::Plane([0,1,0],[6,1,0]):
P := plot::Point3d([-1/2,4,3/4]):
plot(A,B,C,P)
```




```
eq1 :=   x + 3*y + 2*z = 1:
eq2 := 2*x + 6*y + 4*z = 7:
eq3 :=  -x - 3*y - 2*z = 3:
linsolve({eq1,eq2,eq3}, {x,y,z})
```

FAIL

Declarations of matrices

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

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

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

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

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

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

```
matrix::identity(4)
```

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

```
matrix(4, 5, [5,3,2,5,8], Diagonal)
```

$\begin{pmatrix} 5 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \end{pmatrix}$

```
matrix(5, 5, [1,2,3,4,5,6,7,8,9], Banded)
```

$\begin{pmatrix} 5 & 6 & 7 & 8 & 9 \\ 4 & 5 & 6 & 7 & 8 \\ 3 & 4 & 5 & 6 & 7 \\ 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$

```
matrix(5, 5, [5,3,3,5,8], Banded)
```

$$\begin{pmatrix} 3 & 5 & 8 & 0 & 0 \\ 3 & 3 & 5 & 8 & 0 \\ 5 & 3 & 3 & 5 & 8 \\ 0 & 5 & 3 & 3 & 5 \\ 0 & 0 & 5 & 3 & 3 \end{pmatrix}$$

```
matrix(4, 4, (n,m) -> exp(n + m))
```

$$\begin{pmatrix} e^2 & e^3 & e^4 & e^5 \\ e^3 & e^4 & e^5 & e^6 \\ e^4 & e^5 & e^6 & e^7 \\ e^5 & e^6 & e^7 & e^8 \end{pmatrix}$$

```
matrix(4, 4, n -> exp(n), Diagonal)
```

$$\begin{pmatrix} e & 0 & 0 & 0 \\ 0 & e^2 & 0 & 0 \\ 0 & 0 & e^3 & 0 \\ 0 & 0 & 0 & e^4 \end{pmatrix}$$

```
linalg::hilbert(5)
```

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{pmatrix}$$

```
linalg::randomMatrix(5, 5, Dom::Integer)
```

$$\begin{pmatrix} 824 & -65 & -814 & -741 & -979 \\ -764 & 216 & 663 & 880 & 916 \\ 617 & -535 & 597 & -245 & 79 \\ 747 & 477 & -535 & -906 & -905 \\ -266 & -8 & 765 & 448 & -348 \end{pmatrix}$$

```
linalg::randomMatrix(4, 4, Dom::Integer, 0..9)
```

$$\begin{pmatrix} 1 & 4 & 6 & 3 \\ 2 & 1 & 3 & 0 \\ 1 & 8 & 3 & 0 \\ 4 & 5 & 1 & 5 \end{pmatrix}$$

```
linalg::randomMatrix(6, 6,Dom::Integer, 0..100, Diagonal
)
```

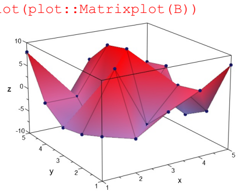
$$\begin{pmatrix} 15 & 0 & 0 & 0 & 0 & 0 \\ 0 & 24 & 0 & 0 & 0 & 0 \\ 0 & 0 & 58 & 0 & 0 & 0 \\ 0 & 0 & 0 & 63 & 0 & 0 \\ 0 & 0 & 0 & 0 & 21 & 0 \\ 0 & 0 & 0 & 0 & 0 & 36 \end{pmatrix}$$

Visualization of matrices

```
B := linalg::randomMatrix(5,5, Dom::Integer, -10..10)
```

$$\begin{pmatrix} -1 & -3 & 1 & 4 & 8 \\ -3 & 9 & -4 & 3 & -4 \\ -4 & -6 & 10 & -1 & -7 \\ -1 & -1 & 9 & -7 & 2 \\ 8 & -6 & 4 & -10 & 1 \end{pmatrix}$$

```
plot(plot::Matrixplot(B))
```




Operations on matrices

```
A := matrix(3,3,[[7,6,0],[8,0,3],[6,1,9]]);
B := matrix(3,3,[[9,7,0],[0,6,1],[0,0,8]]);
C := matrix(2,3,[[7,1,7],[0,8,4]]);
F := matrix(3,2,[[1,8],[7,5],[7,0]]);
```

$$\begin{pmatrix} 7 & 6 & 0 \\ 8 & 0 & 3 \\ 6 & 1 & 9 \end{pmatrix}$$

$$\begin{pmatrix} 9 & 7 & 0 \\ 0 & 6 & 1 \\ 0 & 0 & 8 \end{pmatrix}$$

$$\begin{pmatrix} 7 & 1 & 7 \\ 0 & 8 & 4 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 8 \\ 7 & 5 \\ 7 & 0 \end{pmatrix}$$

```
A + B
```

$$\begin{pmatrix} 16 & 13 & 0 \\ 8 & 6 & 4 \\ 6 & 1 & 17 \end{pmatrix}$$

```
3*A + 5*B
```

$$\begin{pmatrix} 66 & 53 & 0 \\ 24 & 30 & 14 \\ 18 & 3 & 67 \end{pmatrix}$$

```
A*B  // this operation can be done
```

$$\begin{pmatrix} 63 & 85 & 6 \\ 72 & 56 & 24 \\ 54 & 48 & 73 \end{pmatrix}$$

```
B*F  // this operations can be done also
```

$$\begin{pmatrix} 58 & 107 \\ 49 & 30 \\ 56 & 0 \end{pmatrix}$$

```
F*B // this operations cannot be done
Error: The dimensions do not match. [(Dom::Matrix(Dom::ExpressionField()))::_mult2]
```

```
A^(-1)
```

$$\begin{pmatrix} \frac{1}{115} & \frac{18}{115} & -\frac{6}{115} \\ \frac{18}{115} & -\frac{21}{115} & \frac{7}{115} \\ -\frac{8}{345} & -\frac{29}{345} & \frac{16}{115} \end{pmatrix}$$

```
float(A^(-1))
```

$$\begin{pmatrix} 0.008695652174 & 0.1565217391 & -0.05217391304 \\ 0.1565217391 & -0.1826086957 & 0.06086956522 \\ -0.0231884058 & -0.08405797101 & 0.1391304348 \end{pmatrix}$$

```
linalg::transpose(A)
```

$$\begin{pmatrix} 7 & 8 & 6 \\ 6 & 0 & 1 \\ 0 & 3 & 9 \end{pmatrix}$$

```
g := x -> x*exp(-x*3);
```

$x \rightarrow x\, e^{-3x}$

```
A := matrix([[7,6,0], [8,0,3], [6,1,9]]):
```

```
map(A, g) //apply a function to all operands of an object
```

$$\begin{pmatrix} 7e^{-21} & 6e^{-18} & 0 \\ 8e^{-24} & 0 & 3e^{-9} \\ 6e^{-18} & e^{-3} & 9e^{-27} \end{pmatrix}$$

```
map(A, x -> x*exp(-x*3))
```

$$\begin{pmatrix} 7e^{-21} & 6e^{-18} & 0 \\ 8e^{-24} & 0 & 3e^{-9} \\ 6e^{-18} & e^{-3} & 9e^{-27} \end{pmatrix}$$

```
f := (x,y) -> x*y:
zip(A,B,f)
```

$$\begin{pmatrix} 63 & 42 & 0 \\ 0 & 0 & 3 \\ 0 & 0 & 72 \end{pmatrix}$$

```
f := (x,y)->max(x,y):
zip(A,B,f)
```

$$\begin{pmatrix} 9 & 7 & 0 \\ 8 & 6 & 3 \\ 6 & 1 & 9 \end{pmatrix}$$

```
g := (x,y)->min(x,y):
zip(A,B,g)
```

$$\begin{pmatrix} 7 & 6 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 8 \end{pmatrix}$$

```
h := (n,m) -> gcd(n,m):
zip(A,B,h)
```

$$\begin{pmatrix} 1 & 1 & 0 \\ 8 & 6 & 1 \\ 6 & 1 & 1 \end{pmatrix}$$

```
A := matrix(3,3,[[3, 2, 5],[-1, 5, 3],[1, 2, -5]])
```

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

```
linalg::det(A)
```

$-132$

Solving systems of linear equations in matrix form

```
equations := [7*x+6*y = 1, 8*x+3*z = 2, 6*x+y+3*z = 3]
```

$[7\,x + 6\,y = 1,\ 8\,x + 3\,z = 2,\ 6\,x + y + 3\,z = 3]$

```
AM := linalg::expr2Matrix(equations,[x,y,z])
```

$$\begin{pmatrix} 7 & 6 & 0 & 1 \\ 8 & 0 & 3 & 2 \\ 6 & 1 & 3 & 3 \end{pmatrix}$$

```
B := linalg::col(AM,4)
```

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

```
A := linalg::delCol(AM,4)
```

$$\begin{pmatrix} 7 & 6 & 0 \\ 8 & 0 & 3 \\ 6 & 1 & 3 \end{pmatrix}$$

```
linalg::matlinsolve(A,B)
```

$$\begin{pmatrix} -\frac{5}{19} \\ \frac{9}{19} \\ \frac{26}{19} \end{pmatrix}$$

```
solve(equations, {x,y,z} )
```

$$\left\{\left[x = -\frac{5}{19},\ y = \frac{9}{19},\ z = \frac{26}{19}\right]\right\}$$

Gaussian elimination

```
reset():
```

```
A := Dom::Matrix(Dom::Integer)([[3, 2, 5, 4], [-1, 5, 3, -1], [1, 2, -5, 0]]);
B := Dom::Matrix(Dom::Rational)([[3, 2, 5, 4], [-1, 5, 3, -1], [1, 2, -5, 0]]);
C := Dom::Matrix(Dom::Float)([[3, 2, 5, 4], [-1, 5, 3, -1], [1, 2, -5, 0]])
```

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

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

$$\begin{pmatrix} 3.0 & 2.0 & 5.0 & 4.0 \\ -1.0 & 5.0 & 3.0 & -1.0 \\ 1.0 & 2.0 & -5.0 & 0.0 \end{pmatrix}$$

```
linalg::rank(A);
linalg::rank(B);
linalg::rank(C);
```

3

3

3

```
linalg::gaussElim(A);
linalg::gaussElim(B);
linalg::gaussElim(C)
```

$$\begin{pmatrix} 3 & 2 & 5 & 4 \\ 0 & 17 & 14 & 1 \\ 0 & 0 & -132 & -24 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 2 & 5 & 4 \\ 0 & \frac{17}{3} & \frac{14}{3} & \frac{1}{3} \\ 0 & 0 & -\frac{132}{17} & -\frac{24}{17} \end{pmatrix}$$

$$\begin{pmatrix} 3.0 & 2.0 & 5.0 & 4.0 \\ 0.0 & 5.666666667 & 4.666666667 & 0.3333333333 \\ 0.0 & 0.0 & -7.764705882 & -1.411764706 \end{pmatrix}$$

```
use(linalg, addRow, swapRow, multRow):
```

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

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

```
addRow(A, 3, 2, 1) // addRow(A, r1, r2, s1) adds s1 times row r1 to row r2, in the matrix A.
```

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

```
addRow(%, 3, 1, -3)
```

$$\begin{pmatrix} 0 & -4 & 20 & 4 \\ 0 & 7 & -2 & -1 \\ 1 & 2 & -5 & 0 \end{pmatrix}$$

```
swapRow(%, 1, 3)
```

$$\begin{pmatrix} 1 & 2 & -5 & 0 \\ 0 & 7 & -2 & -1 \\ 0 & -4 & 20 & 4 \end{pmatrix}$$

```
multRow(%, 3, 1/4)
```

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

```
addRow(%, 3, 2, 7)
```

$$\begin{pmatrix} 1 & 2 & -5 & 0 \\ 0 & 0 & 33 & 6 \\ 0 & -1 & 5 & 1 \end{pmatrix}$$

```
swapRow(%, 2, 3)
```

$$\begin{pmatrix} 1 & 2 & -5 & 0 \\ 0 & -1 & 5 & 1 \\ 0 & 0 & 33 & 6 \end{pmatrix}$$

```
multRow(%,2, -1)   // remove -1
```

$$\begin{pmatrix} 1 & 2 & -5 & 0 \\ 0 & 1 & -5 & -1 \\ 0 & 0 & 33 & 6 \end{pmatrix}$$

```
multRow(%, 3, 1/33)   // remove 33
```

$$\begin{pmatrix} 1 & 2 & -5 & 0 \\ 0 & 1 & -5 & -1 \\ 0 & 0 & 1 & \frac{2}{11} \end{pmatrix}$$

Minor expansion

```
reset():
```

```
use(linalg, nrows, ncols, delRow, delCol):
```

```
A := matrix([[3,2,5], [-1,5,3], [1,2,-5]])
```

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

```
minor := proc(A, i, j)
begin
   B := delRow(A,i);
   B := delCol(B,j);
   return(hold(Det)(B))
end:
```

```
Det := proc(A)
local r, j;
begin
   r := 0;
   if (nrows(A) <> ncols(A)) then
      error("Wrong input matrix")
   else
      if nrows(A)=1 then
         return(A[1,1])
      else
         for j from 1 to ncols(A) do
            r:=r+(-1)^(1+j)*A[1,j]*minor(A,1,j)
         end;
      end;
   end;
   return(r)
end:
```

```
Det(A)
```

$$5\,\mathrm{Det}\left(\begin{pmatrix} -1 & 5 \\ 1 & 2 \end{pmatrix}\right) - 2\,\mathrm{Det}\left(\begin{pmatrix} -1 & 3 \\ 1 & -5 \end{pmatrix}\right) + 3\,\mathrm{Det}\left(\begin{pmatrix} 5 & 3 \\ 2 & -5 \end{pmatrix}\right)$$

```
eval(%)
```

$$17\,\mathrm{Det}((-5)) - 14\,\mathrm{Det}((2)) - 19\,\mathrm{Det}((1))$$

```
eval(%)
```

$$-132$$

```
B := Dom::Matrix(Dom::Integer)(   [[3,2,5,4], [2,5,3,8], [1,2,3,0], [2,3,4,5]])
```

$$\begin{pmatrix} 3 & 2 & 5 & 4 \\ 2 & 5 & 3 & 8 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 4 & 5 \end{pmatrix}$$

```
Det(B)
```

$$5\,\mathrm{Det}\left(\begin{pmatrix} 2 & 5 & 8 \\ 1 & 2 & 0 \\ 2 & 3 & 5 \end{pmatrix}\right) - 4\,\mathrm{Det}\left(\begin{pmatrix} 2 & 5 & 3 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix}\right) - 2\,\mathrm{Det}\left(\begin{pmatrix} 2 & 3 & 8 \\ 1 & 3 & 0 \\ 2 & 4 & 5 \end{pmatrix}\right) + 3\,\mathrm{Det}\left(\begin{pmatrix} 5 & 3 & 8 \\ 2 & 3 & 0 \\ 3 & 4 & 5 \end{pmatrix}\right)$$

```
eval(%)
```

$$28\,\mathrm{Det}\left(\begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}\right) - 19\,\mathrm{Det}\left(\begin{pmatrix} 1 & 0 \\ 2 & 5 \end{pmatrix}\right) + \mathrm{Det}\left(\begin{pmatrix} 2 & 0 \\ 3 & 5 \end{pmatrix}\right) + 4\,\mathrm{Det}\left(\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}\right) + 11\,\mathrm{Det}\left(\begin{pmatrix} 3 & 0 \\ 4 & 5 \end{pmatrix}\right) + 16\,\mathrm{Det}\left(\begin{pmatrix} 2 & 3 \\ 3 & 4 \end{pmatrix}\right)$$

```
eval(%)
```

$$36\,\mathrm{Det}((4)) - 20\,\mathrm{Det}((3)) - 68\,\mathrm{Det}((2)) + 16\,\mathrm{Det}((5))$$

```
eval(%)
```

$$28$$

## Angle

We compute the angle between the two vectors $\begin{pmatrix} 2 \\ 5 \end{pmatrix}$ and $\begin{pmatrix} -3 \\ 3 \end{pmatrix}$:

```
phi := linalg::angle(
  matrix([2, 5]), matrix([-3, 3])
)
```

$$\arccos\left(\frac{\sqrt{18}\,\sqrt{29}}{58}\right)$$

We give two further examples:

```
linalg::angle(
  matrix([1, -1]), matrix([1, 1])
)
```

$$\frac{\pi}{2}$$

```
linalg::angle(
  matrix([1, 1]), matrix([-1, -1])
)
```

$$\pi$$

## Trace

```
A := Dom::Matrix(Dom::Integer)
  (3, 3, (i, j) -> 3*(i - 1) + j)
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

```
linalg::tr(A)
```

$$15$$

## Determinant

```
A := matrix([[a11, a12], [a21, a22]])
```

$$\begin{pmatrix} a11 & a12 \\ a21 & a22 \end{pmatrix}$$

which gives us the well-known formula for the determinant of an arbitrary $2 \times 2$ matrix:

```
linalg::det(A)
```

$$a11\,a22 - a12\,a21$$

The standard algorithms for computing determinants suffer from extreme internal expression swell when many symbolic entries are involved. For this reason, the following computation takes some time:

```
A := matrix([[x, y, z, x, y, z, 0, 0, 0],
              [0, x, y, z, x, y, z, 0, 0],
              [0, 0, x, y, z, x, y, z, 0],
              [0, 0, 0, x, y, z, x, y, z],
              [z, 0, 0, 0, x, y, z, x, y],
              [y, z, 0, 0, 0, x, y, z, x],
              [x, y, z, 0, 0, 0, x, y, z],
              [z, x, y, z, 0, 0, 0, x, y],
              [y, z, x, y, z, 0, 0, 0, x]
             ]):
t1 := time((d1 := linalg::det(A)))*msec:
t1, d1
```

$$218.4014\ \mathrm{msec},\ 8\,x^9 + 72\,x^4\,y\,z^4 - 240\,x^3\,y^3\,z^3 + 216\,x^2\,y^5\,z^2 - 72\,x\,y^7\,z + 8\,y^9 + 8\,z^9$$

## Eigenvalues

We compute the eigenvalues of the matrix $A = \begin{pmatrix} 1 & 4 & 2 \\ 1 & 4 & 2 \\ 2 & 5 & 3 \end{pmatrix}$:

```
A := matrix([[1, 4, 2], [1, 4, 2], [2, 5, 3]]):
linalg::eigenvalues(A)
```

$$\{0,\ 4 - \sqrt{15},\ \sqrt{15} + 4\}$$

If we consider the matrix over the domain `Dom::Float`, then the call of `linalg::eigenvalues(A)` results in a numerical computation of the eigenvalues of `A` via numeric::eigenvalues:

```
B := Dom::Matrix(Dom::Float)(A):
linalg::eigenvalues(B)
```

$$\{-1.370431546\ 10^{-15},\ 0.1270166538,\ 7.872983346\}$$

With the option *`Multiple`* we get the information about the algebraic multiplicity of each eigenvalue:

```
C := Dom::Matrix(Dom::Rational)(4, 4, [[-3], [0, 6]])
```

$$\begin{pmatrix} -3 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

```
linalg::eigenvalues(C, Multiple)
```

$[[-3, 1], [0, 2], [6, 1]]$

## Eigenvectors

We compute the eigenvalues and the eigenvectors of the matrix $A = \begin{pmatrix} 1 & -3 & 3 \\ 6 & -10 & 6 \\ 6 & 6 & 4 \end{pmatrix}$ :

```
A := Dom::Matrix(Dom::Rational)(
  [[1, -3, 3], [6, -10, 6], [6, 6, 4]]
):
Ev:= linalg::eigenvectors(A)
```

$$\left[\left[-11, 1, \left[\begin{pmatrix} -\frac{7}{10} \\ -\frac{9}{5} \\ 1 \end{pmatrix}\right]\right], \left[-2, 1, \left[\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}\right]\right], \left[8, 1, \left[\begin{pmatrix} \frac{1}{4} \\ \frac{5}{12} \\ 1 \end{pmatrix}\right]\right]\right]$$

The matrix `A` is diagonalizable. Hence, we extract the eigenvectors and combine them to a matrix `P` such that `P^-1 * A * P` is the diagonal matrix whose diagonal entries are given by the corresponding eigenvalues:

```
Eigenvectors:= Ev[1][3][1], Ev[2][3][1], Ev[3][3][1]
```

$$\begin{pmatrix} -\frac{7}{10} \\ -\frac{9}{5} \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} \frac{1}{4} \\ \frac{5}{12} \\ 1 \end{pmatrix}$$

```
P:= Eigenvectors[1].Eigenvectors[2].Eigenvectors[3]
```

$$\begin{pmatrix} -\frac{7}{10} & -1 & \frac{1}{4} \\ -\frac{9}{5} & 0 & \frac{5}{12} \\ 1 & 1 & 1 \end{pmatrix}$$

```
P^-1 * A * P
```

$$\begin{pmatrix} -11 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 8 \end{pmatrix}$$

A more skillful way of extracting the above eigenvectors from the output generated by `linalg::eigenvectors` is the following:

```
map(Ev, op@op, 3)
```

$$\left[\begin{pmatrix} -\frac{7}{10} \\ -\frac{9}{5} \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} \frac{1}{4} \\ \frac{5}{12} \\ 1 \end{pmatrix}\right]$$

If we consider the matrix `A` over the domain `Dom::Float`, the call of `linalg::eigenvectors(A)` results in a numerical computation of the eigenvalues and the eigenvectors of `A` via the function `numeric::eigenvectors`:

```
B := Dom::Matrix(Dom::Float)(A):
linalg::eigenvectors(B)
```

$$\left[\left[8.0, 1, \left[\begin{pmatrix} 0.2248595067 \\ 0.3747658445 \\ 0.8994380268 \end{pmatrix}\right]\right], \left[-2.0, 1, \left[\begin{pmatrix} 0.7071067812 \\ 0.0 \\ -0.7071067812 \end{pmatrix}\right]\right], \left[-11.0, 1, \left[\begin{pmatrix} 0.3218603429 \\ 0.8276408818 \\ -0.4598004899 \end{pmatrix}\right]\right]\right]$$

## Characteristic polynomial

We define a matrix over the rational numbers:

```
A := Dom::Matrix(Dom::Real)([[1, 2], [3, 4]])
```

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

Then the characteristic polynomial $p_A(x)$ is given by:

```
linalg::charpoly(A, x)
```

$x^2 - 5x - 2$

```
B := matrix(3,3,[[1, 2, -1], [-1,-1,4],[3, 4, -5]])
```

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

```
solve(linalg::charpoly(B, x),x)
```

$$\left\{2, -\frac{\sqrt{41}}{2} - \frac{7}{2}, \frac{\sqrt{41}}{2} - \frac{7}{2}\right\}$$

```
linalg::eigenvalues(B)
```

$$\left\{2, -\frac{\sqrt{41}}{2} - \frac{7}{2}, \frac{\sqrt{41}}{2} - \frac{7}{2}\right\}$$

### Norm

```
M := matrix([[a, b], [c, d]]):
norm(M)=norm(M, Infinity);
norm(M, 1);
norm(M, 2)=norm(M, Spectral);
norm(M, Frobenius);
```

$\max(|a| + |b|, |c| + |d|) = \max(|a| + |b|, |c| + |d|)$

$\max(|a| + |c|, |b| + |d|)$

$\sigma_1 = \sigma_1$

where

$$\sigma_1 = \sqrt{\max(|\sigma_6 + \sigma_5 + \sigma_4 + \sigma_3 - \sigma_2|, |\sigma_6 + \sigma_5 + \sigma_4 + \sigma_3 + \sigma_2|)}$$

$$\sigma_2 = \frac{\sqrt{\frac{4 a^2 d^2 |b|^2 |c|^2 + a b c d |a|^4 + 2 a b c d |a|^2 |b|^2 + 2 a b c d |a|^2 |c|^2 - 2 a b c d |a|^2 |d|^2 + a b c d |b|^4 - 2 a b c d |b|^2 |c|^2 + 2 a b c d |b|^2 |d|^2 + a b c d |c|^4 + 2 a b c d |c|^2 |d|^2 + a b c d |d|^4 + 4 b^2 c^2 |a|^2 |d|^2}{a b c d}}}{2}$$

$$\sigma_3 = \frac{|d|^2}{2}$$

$$\sigma_4 = \frac{|c|^2}{2}$$

$$\sigma_5 = \frac{|b|^2}{2}$$

$$\sigma_6 = \frac{|a|^2}{2}$$

$$\sqrt{|a|^2 + |b|^2 + |c|^2 + |d|^2}$$

```
M := matrix([a, b,c, d]):
norm(M, Infinity);
norm(M, 1);
Simplify(norm(M, 2));
norm(M, Spectral)
```

$\max(|a|, |b|, |c|, |d|)$

$|a| + |b| + |c| + |d|$

$\sqrt{|a|^2 + |b|^2 + |c|^2 + |d|^2}$

$\sqrt{|a|^2 + |b|^2 + |c|^2 + |d|^2}$

עבור המטריצה $\begin{pmatrix} 3 & -t & 4 \\ -2 & 6 & t \\ 2t & t & 3 \end{pmatrix}$

1. חשב את הפולינום האופייני.
2. מצא את הערך של הפרמטר t כך ש- 7 יהיה ערך עצמי של A וכן המטריצה תהיה סימטרית.
3. חשב את שאר הערכים העצמיים.
4. קבע עפ"י הערכים העצמיים אם המטריצה הפיכה.
5. מצא וקטורים עצמיים.
6. אם ניתן, מצא מטריצה אלכסונית הדומה למטריצה.

```
reset():
use(linalg):
Warning: Identifier 'laplacian' already has a value. It is not exported. [use]
Warning: Identifier 'hessian' already has a value. It is not exported. [use]
Warning: Identifier 'htranspose' already has a value. It is not exported. [use]
Warning: Identifier 'potential' already has a value. It is not exported. [use]
Warning: Identifier 'jacobian' already has a value. It is not exported. [use]
Warning: Identifier 'vectorPotential' already has a value. It is not exported. [use]
Warning: Identifier 'curl' already has a value. It is not exported. [use]
Warning: Identifier 'gradient' already has a value. It is not exported. [use]
Warning: Identifier 'transpose' already has a value. It is not exported. [use]
Warning: Identifier 'det' already has a value. It is not exported. [use]
Warning: Identifier 'divergence' already has a value. It is not exported. [use]
```

```
A:=matrix([[3,-t,4],[-2,6,t],[2*t,t,3]]);
```

$$\begin{pmatrix} 3 & -t & 4 \\ -2 & 6 & t \\ 2t & t & 3 \end{pmatrix}$$

```
CharPoly:=charpoly(A, x)
```

$$x^3 - 12x^2 + (-t^2 - 10t + 45)x + 2t^3 + 3t^2 + 62t - 54$$

```
t_opt:=solve(Simplify(subs(expr(CharPoly),x=7)),t)
```

$$\{-2, 2\}$$

```
A1:=subs(A,t=-2)
```

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

We have got a non symmetric matrix

```
A2:=subs(A,t=2)
```

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

We have got a symmetric matrix, thus t=2.

```
eigenvalues(A2, Multiple)
```

$$[[-2, 1], [7, 2]]$$

אף אחד מהערכים העצמיים אינו 0 ולכן המטריצה הפיכה

```
EigenV:= Simplify(linalg::eigenvectors(A2))
```

$$\left[\left[-2, 1, \left[\begin{pmatrix} -1 \\ -\frac{1}{2} \\ 1 \end{pmatrix}\right]\right], \left[7, 2, \left[\begin{pmatrix} -\frac{1}{2} \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}\right]\right]\right]$$

```
EigenVMatrix:= EigenV[1][3][1].EigenV[2][3][1].EigenV[2][3][2]
```

$$\begin{pmatrix} -1 & -\frac{1}{2} & 1 \\ -\frac{1}{2} & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

```
det(EigenVMatrix)
```

$$-\frac{9}{4}$$

הפיכה ולכן יש מטריצה אלכסונית שדומה לה

```
EigenVMatrix^-1 * A2 * EigenVMatrix
```

$$\begin{pmatrix} -2 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 7 \end{pmatrix}$$

מצא מטריצה דומה למטריצה $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

```
reset():
use(linalg):
Warning: Identifier 'laplacian' already has a value. It is not exported. [use]
Warning: Identifier 'hessian' already has a value. It is not exported. [use]
Warning: Identifier 'htranspose' already has a value. It is not exported. [use]
Warning: Identifier 'potential' already has a value. It is not exported. [use]
Warning: Identifier 'jacobian' already has a value. It is not exported. [use]
Warning: Identifier 'vectorPotential' already has a value. It is not exported. [use]
Warning: Identifier 'curl' already has a value. It is not exported. [use]
Warning: Identifier 'gradient' already has a value. It is not exported. [use]
Warning: Identifier 'transpose' already has a value. It is not exported. [use]
Warning: Identifier 'det' already has a value. It is not exported. [use]
Warning: Identifier 'divergence' already has a value. It is not exported. [use]
```

```
A:=matrix([[1,2],[3,4]]):
B:=matrix([[a,b],[c,d]]):
```

```
Expr1:=det(A)=det(B);
Expr2:=tr(A)=tr(B);
Expr3:=subs({Expr1,Expr2},{b=2,c=1})
```

$$-2 = a d - b c$$

$$5 = a + d$$

$$\{5 = a + d, -2 = a d - 2\}$$

```
solve(Expr3,{a,d})
```

$$\{[a = 0, d = 5], [a = 5, d = 0]\}$$

```
B:=subs(B,{a=0,d=5,b=2,c=1});
```

$$\begin{pmatrix} 0 & 2 \\ 1 & 5 \end{pmatrix}$$

```
addRow(multRow(swapRow(gaussElim(A),1,2),1,-1),1,2,1.5)
```

$$\begin{pmatrix} 0 & 2 \\ 1 & 5.0 \end{pmatrix}$$