Algebra lineare

Note

Autore

Claudio Marsan
Liceo Cantonale di Mendrisio
Via Agostino Maspoli
CH-6850 Mendrisio (Switzerland)
e-mail: claudio.marsan@liceomendrisio.ch

Versione

Versione 2.0, 16 marzo 2003
Maple V Release 6.02 for Windows 2000

```
> restart:
```

Il package ***linalg*** è necessario per trattare con gli oggetti dell'algebra lineare.

```
> with(linalg);
```

```
Warning, the protected names norm and trace have been redefined and
unprotected
```

[*BlockDiagonal*, *GramSchmidt*, *JordanBlock*, *LUdecomp*, *QRdecomp*, *Wronskian*, *addcol*, *addrow*, *adj*, *adjoint*, *angle*, *augment*, *backsub*, *band*, *basis*, *bezout*, *blockmatrix*, *charmat*, *charpoly*, *cholesky*, *col*, *coldim*, *colspace*, *colspan*, *companion*, *concat*, *cond*, *copyinto*, *crossprod*, *curl*, *definite*, *delcols*, *delrows*, *det*, *diag*, *diverge*, *dotprod*, *eigenvals*, *eigenvalues*, *eigenvectors*, *eigenvects*, *entermatrix*, *equal*, *exponential*, *extend*, *ffgausselim*, *fibonacci*, *forwardsub*, *frobenius*, *gausselim*, *gaussjord*, *geneqns*, *genmatrix*, *grad*, *hadamard*, *hermite*, *hessian*, *hilbert*, *htranspose*, *ihermite*, *indexfunc*, *innerprod*, *intbasis*, *inverse*, *ismith*, *issimilar*, *iszero*, *jacobian*, *jordan*, *kernel*, *laplacian*, *leastsqrs*, *linsolve*, *matadd*, *matrix*, *minor*, *minpoly*, *mulcol*, *mulrow*, *multiply*, *norm*, *normalize*, *nullspace*, *orthog*, *permanent*, *pivot*, *potential*, *randmatrix*, *randvector*, *rank*, *ratform*, *row*, *rowdim*, *rowspace*, *rowspan*, *rref*, *scalarmul*, *singularvals*, *smith*, *stackmatrix*, *submatrix*, *subvector*, *sumbasis*, *swapcol*, *swaprow*, *sylvester*, *toeplitz*, *trace*, *transpose*, *vandermonde*, *vecpotent*, *vectdim*, *vector*, *wronskian*]

1° modo per definire una matrice (come lista di liste)

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

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \\ 9 & 0 & 1 \end{bmatrix}$$

2° modo per definire una matrice (dimensione e lista degli elementi)

```
> B := matrix(2, 3, [1, 2, 3, 4, 5, 6]);
```

$$B := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

3° modo per definire una matrice (dimensione e funzione degli indici)

```
> C := matrix(3, 3, (i,j) -> i+j);
```

$$C := \begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{bmatrix}$$

```
> f := proc(i, j)
  if i < j then i^j else j^i; end if;
  end proc;
```

$$f := \mathbf{proc}(i, j)\ \mathbf{if}\ i < j\ \mathbf{then}\ i\hat{}j\ \mathbf{else}\ j\hat{}i\ \mathbf{end\ if\ end\ proc}$$

```
> F := matrix(4, 3, f);
```

$$F := \begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 8 \\ 1 & 8 & 27 \\ 1 & 16 & 81 \end{bmatrix}$$

```
> id := n -> matrix(n, n, proc(i, j) if i=j then 1 else 0 end if
  end);
```

$$id := n \rightarrow \mathrm{matrix}(n, n, \mathbf{proc}(i, j)\ \mathbf{if}\ i = j\ \mathbf{then}\ 1\ \mathbf{else}\ 0\ \mathbf{end\ if\ end\ proc})$$

```
> id(2);
```

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

```
> id(3);
```

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

Accesso ad un elemento di una matrice

```
> A;
```

$$A$$

```
> evalm(A); # evaluate matrix: visualizza la matrice
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \\ 9 & 0 & 1 \end{bmatrix}$$

```
> A[2,3]; # accesso all'elemento della 2a riga e 3a colonna
```

$$5$$

```
> A[4,1] := -3;
```

$$A_{4,1} := -3$$

```
> evalm(A);
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \\ -3 & 0 & 1 \end{bmatrix}$$

Operazioni con le matrici

```
> restart:
> with(linalg):
```

```
Warning, the protected names norm and trace have been redefined and
unprotected
```

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

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

```
> B := matrix(3, 3, ilcm);
```

$$B := \begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 6 \\ 3 & 6 & 3 \end{bmatrix}$$

```
> A + B;
```

$$A + B$$

```
> evalm(%);
```

$$\begin{bmatrix} 2 & 4 & 6 \\ 6 & 7 & 12 \\ 10 & 14 & 12 \end{bmatrix}$$

```
> A - B;
```

$$A - B$$

```
> evalm(%);
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 3 & 0 \\ 4 & 2 & 6 \end{bmatrix}$$

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

$$A \,\&*\, B$$

```
> evalm(%);
```

$$\begin{bmatrix} 14 & 24 & 24 \\ 32 & 54 & 60 \\ 50 & 84 & 96 \end{bmatrix}$$

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

$$B \,\&*\, A$$

```
> evalm(%);
```

$$\begin{bmatrix} 30 & 36 & 42 \\ 52 & 62 & 72 \\ 48 & 60 & 72 \end{bmatrix}$$

```
> 3 * A;
```

$$3\,A$$

```
> evalm(%);
```

$$\begin{bmatrix} 3 & 6 & 9 \\ 12 & 15 & 18 \\ 21 & 24 & 27 \end{bmatrix}$$

```
> A^3;
```

$$A^3$$

```
> evalm(%);
```

$$\begin{bmatrix} 468 & 576 & 684 \\ 1062 & 1305 & 1548 \\ 1656 & 2034 & 2412 \end{bmatrix}$$

```
> B^(-1); # inversa di B
```

$$\frac{1}{B}$$

```
> evalm(%);
```

$$\begin{bmatrix} \frac{-5}{2} & 1 & \frac{1}{2} \\ 1 & \frac{-1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{-1}{6} \end{bmatrix}$$

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

$$\frac{1}{A}$$

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

$$0$$

```
> rank(A); # rango
```

$$2$$

```
> rank(B);
```

$$3$$

```
> evalm(A); transpose(A); # trasposta
```

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

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

```
> adjoint(A); # aggiunta
```

$$\begin{bmatrix} -3 & 6 & -3 \\ 6 & -12 & 6 \\ -3 & 6 & -3 \end{bmatrix}$$

```
> submatrix(A, 2..3, 2..3); # sottomatrice
```

$$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$$

```
> submatrix(A, 1..3, 2..3);
```

$$\begin{bmatrix} 2 & 3 \\ 5 & 6 \\ 8 & 9 \end{bmatrix}$$

Matrici particolari

```
> restart:
> with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected
> A := randmatrix(5, 4); # matrice casuale
```

$$A := \begin{bmatrix} -85 & -55 & -37 & -35 \\ 97 & 50 & 79 & 56 \\ 49 & 63 & 57 & -59 \\ 45 & -8 & -93 & 92 \\ 43 & -62 & 77 & 66 \end{bmatrix}$$

```
> B := randmatrix(10, 10, sparse); # matrice casuale sparsa (quasi
  tutti gli elementi sono 0)
```

$$B := \begin{bmatrix} 0 & 0 & 0 & 0 & 41 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -62 & 0 & 53 & 0 & 0 & -91 & 0 \\ 0 & 0 & 0 & -84 & 0 & 0 & 0 & -87 & 0 & 0 \\ 0 & -61 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -98 & 0 & 9 & 0 & 0 & 0 & 0 & 72 & 0 \\ 0 & 0 & 0 & -43 & 0 & -76 & 0 & -23 & 0 & 0 \\ 0 & 0 & 0 & -18 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 94 & 0 & 0 & 0 & -66 & 0 & 0 \\ 0 & 0 & 0 & 49 & 0 & 0 & 28 & 0 & 0 & 0 \\ 83 & 22 & -34 & 0 & 0 & 88 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> M := matrix(3,3); # matrice simbolica
```

$$M := \mathrm{array}(1 .. 3, 1 .. 3, [\ ])$$

```
> evalm(M);
```

$$\begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix}$$

```
> vandermonde([1,2,3,4]); # matrice di Vandermonde
```

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{bmatrix}$$

```
> diag(-1,2,-3,4); # matrice diagonale
```

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

```
> hilbert(3); # ha 1/(i+j-x) come elemento (i,j). Se x non è
  specificato vale 1.
```

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$$

```
> hilbert(3, 9);
```

$$\begin{bmatrix} \frac{-1}{7} & \frac{-1}{6} & \frac{-1}{5} \\ \frac{-1}{6} & \frac{-1}{5} & \frac{-1}{4} \\ \frac{-1}{5} & \frac{-1}{4} & \frac{-1}{3} \end{bmatrix}$$

Vettori

```
> restart:
> with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected
> v := vector(3, [1, 2, 3]);
```

$$v := [1, 2, 3]$$

```
> w := vector(3, i -> i^2);
```

$$w := [1, 4, 9]$$

```
> u := vector(3, exp);
```

$$u := [\mathbf{e}, \mathbf{e}^2, \mathbf{e}^3]$$

```
> v;
```

$$v$$

```
> evalm(v);
```

$$[1, 2, 3]$$

```
> w[2];
```

$$4$$

```
> w[2] := 13;
```

$$w_2 := 13$$

```
> evalm(w);
```

$$[1, 13, 9]$$

```
> v + w;
```

$$v + w$$

```
> evalm(%);
```

$$[2, 15, 12]$$

```
> v - w;
```

$$v - w$$

```
> evalm(%);
```

$$[0, -11, -6]$$

```
> 5*v;
```

$$5\,v$$

```
> evalm(%);
```

$$[5, 10, 15]$$

```
> innerprod(v, w); # prodotto scalare
```

$$54$$

```
> A := matrix(5, 3, max);
```

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 3 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \\ 5 & 5 & 5 \end{bmatrix}$$

```
> A &* v;
```

$$A \,\&*\, v$$

```
> evalm(%);
```

$$[14, 15, 18, 24, 30]$$

```
> norm(v, 2); # norma
```

$$\sqrt{14}$$

```
> angle(v, w);
```

$$\arccos\left(\frac{27}{1757}\sqrt{14}\,\sqrt{251}\right)$$

```
> evalf(convert(%, degrees));
```

$$24.36351687\ degrees$$

```
> crossprod(v, w); # prodotto vettoriale
```

$$[-21, -6, 11]$$

Ancora sui vettori

```
> restart:
> with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected
```

Definizione di vettore senza indicare la dimensione

```
> v := vector([1,2,3]);
```

$$v := [1, 2, 3]$$

```
> w := vector(3, [3,2,4]);
```

$$w := [3, 2, 4]$$

```
> dotprod(v, w); # prodotto scalare
```

$$19$$

```
> v + w;
```

$$v + w$$

```
> evalm(%);
```

$$[4, 4, 7]$$

```
> matadd(v, w); # come evalm(v + w);
```

$$[4, 4, 7]$$

```
> scalarmul(v, 5); # come evalm(5*v);
```

$$[5, 10, 15]$$

```
> normalize(v); # vettore  parallelo e concorde a v, di lunghezza
  unitaria
```

$$\left[\frac{1}{14}\sqrt{14}, \frac{1}{7}\sqrt{14}, \frac{3}{14}\sqrt{14}\right]$$

Ancora sulle matrici

```
> restart:
> with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected
> A := randmatrix(3, 3);
```

$$A := \begin{bmatrix} -85 & -55 & -37 \\ -35 & 97 & 50 \\ 79 & 56 & 49 \end{bmatrix}$$

```
> B := randmatrix(3, 3);
```

$$B := \begin{bmatrix} 63 & 57 & -59 \\ 45 & -8 & -93 \\ 92 & 43 & -62 \end{bmatrix}$$

```
> A + B; evalm(%);
```

$$A + B$$

$$\begin{bmatrix} -22 & 2 & -96 \\ 10 & 89 & -43 \\ 171 & 99 & -13 \end{bmatrix}$$

```
> matadd(A, B); # equivale alla coppia di comandi sopra
```

$$\begin{bmatrix} -22 & 2 & -96 \\ 10 & 89 & -43 \\ 171 & 99 & -13 \end{bmatrix}$$

```
> evalm(A &* B);
```

$$\begin{bmatrix} -11234 & -5996 & 12424 \\ 6760 & -621 & -10056 \\ 12005 & 6162 & -12907 \end{bmatrix}$$

```
> multiply(A, B); # equivale a sopra
```

$$\begin{bmatrix} -11234 & -5996 & 12424 \\ 6760 & -621 & -10056 \\ 12005 & 6162 & -12907 \end{bmatrix}$$

```
> evalm(A * 5);
```

$$\begin{bmatrix} -425 & -275 & -185 \\ -175 & 485 & 250 \\ 395 & 280 & 245 \end{bmatrix}$$

```
> scalarmul(A, 5); # equivale a sopra
```

$$\begin{bmatrix} -425 & -275 & -185 \\ -175 & 485 & 250 \\ 395 & 280 & 245 \end{bmatrix}$$

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

$$\frac{1}{A}$$

$$\begin{bmatrix} \frac{-1953}{121529} & \frac{-623}{121529} & \frac{-839}{121529} \\ \frac{-5665}{121529} & \frac{1242}{121529} & \frac{-5545}{121529} \\ \frac{9623}{121529} & \frac{-415}{121529} & \frac{10170}{121529} \end{bmatrix}$$

```
> inverse(A);  # equivale a sopra
```

$$\begin{bmatrix} \frac{-1953}{121529} & \frac{-623}{121529} & \frac{-839}{121529} \\ \frac{-5665}{121529} & \frac{1242}{121529} & \frac{-5545}{121529} \\ \frac{9623}{121529} & \frac{-415}{121529} & \frac{10170}{121529} \end{bmatrix}$$

Matrice identità

```
> array(identity, 1..3, 1..3);
```

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

Manipolazioni con matrici e vettori

```
> restart:
> with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected
> a := vector(3);
```

$$a := \text{array}(1 .. 3, [\ ])$$

```
> b := vector(3);
```

$$b := \text{array}(1 .. 3, [\ ])$$

```
> c := vector(3);
```

$$c := \text{array}(1 .. 3, [\ ])$$

```
> evalm(a);
```

$$[a_1, a_2, a_3]$$

```
> A := augment(a, b);
```

$$A := \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}$$

```
> B := augment(A, c);
```

$$B := \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}$$

```
> C := augment(A, B);
```

$$C := \begin{bmatrix} a_1 & b_1 & a_1 & b_1 & c_1 \\ a_2 & b_2 & a_2 & b_2 & c_2 \\ a_3 & b_3 & a_3 & b_3 & c_3 \end{bmatrix}$$

```
> X := stackmatrix(a, b);
```

$$X := \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix}$$

```
> Y := stackmatrix(X, c);
```

$$Y := \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}$$

```
> Z := stackmatrix(X, Y);
```

$$Z := \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}$$

```
> d := vector(3);
```

$$d := \mathrm{array}(1 .. 3, [\ ])$$

```
> A := augment(B, d);
```

$$A := \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{bmatrix}$$

```
> addrow(A, 1, 2, 3); # moltiplica la 1a riga di A per 3 e somma
  il risultato alla 2a riga
```

$$\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ 3\,a_1 + a_2 & 3\,b_1 + b_2 & 3\,c_1 + c_2 & 3\,d_1 + d_2 \\ a_3 & b_3 & c_3 & d_3 \end{bmatrix}$$

```
> addcol(A, 1, 2, 3); # moltiplica la 1a colonna di A per 3 e
  somma il risultato alla 2a colonna
```

$$\begin{bmatrix} a_1 & 3\,a_1+b_1 & c_1 & d_1 \\ a_2 & 3\,a_2+b_2 & c_2 & d_2 \\ a_3 & 3\,a_3+b_3 & c_3 & d_3 \end{bmatrix}$$

```
> mulrow(A, 1, 2); # moltiplica la 1a riga di A per 2
```

$$\begin{bmatrix} 2\,a_1 & 2\,b_1 & 2\,c_1 & 2\,d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{bmatrix}$$

```
> mulcol(A, 1, 2); # moltiplica la 1a colonna di A per 2
```

$$\begin{bmatrix} 2\,a_1 & b_1 & c_1 & d_1 \\ 2\,a_2 & b_2 & c_2 & d_2 \\ 2\,a_3 & b_3 & c_3 & d_3 \end{bmatrix}$$

```
> swaprow(A, 1, 2); # scambia la 1a riga con la 2a
```

$$\begin{bmatrix} a_2 & b_2 & c_2 & d_2 \\ a_1 & b_1 & c_1 & d_1 \\ a_3 & b_3 & c_3 & d_3 \end{bmatrix}$$

```
> swapcol(A, 1, 2); # scambia la 1a colonna con la 2a
```

$$\begin{bmatrix} b_1 & a_1 & c_1 & d_1 \\ b_2 & a_2 & c_2 & d_2 \\ b_3 & a_3 & c_3 & d_3 \end{bmatrix}$$

Sistemi di equazioni lineari in forma matriciale

```
> restart:
> with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected
```

```
> A := randmatrix(4, 4, entries = rand(10));
```

$$A := \begin{bmatrix} 1 & 0 & 7 & 3 \\ 6 & 8 & 5 & 8 \\ 1 & 9 & 5 & 3 \\ 7 & 0 & 4 & 5 \end{bmatrix}$$

```
> b := randvector(4, entries = rand(10));
```

$$b := [6, 1, 7, 9]$$

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

$$\left[\frac{212}{73}, \frac{43}{73}, \frac{157}{73}, \frac{-291}{73}\right]$$

```
> sistema := {
  -2*x + 2*y + 7*z = 0,
  x - y - 3*z = 1,
  3*x + 2*y + 2*z = 5};
```

$$sistema := \{-2\,x+2\,y+7\,z=0,\ x-y-3\,z=1,\ 3\,x+2\,y+2\,z=5\}$$

```
> M := genmatrix(sistema, [x, y, z], flag); # genera la matrice
```

del sistema (nota: senza "flag" non viene scritta la colonna dei termini noti)

$$M := \begin{bmatrix} -2 & 2 & 7 & 0 \\ 1 & -1 & -3 & 1 \\ 3 & 2 & 2 & 5 \end{bmatrix}$$

```
> gausselim(M);
```

$$\begin{bmatrix} -2 & 2 & 7 & 0 \\ 0 & 5 & \frac{25}{2} & 5 \\ 0 & 0 & \frac{1}{2} & 1 \end{bmatrix}$$

```
> backsub(%);
```

$$[3, -4, 2]$$

```
> rref(M);
```

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

```
> N := randmatrix(3, 3, entries=rand(10));
```

$$N := \begin{bmatrix} 6 & 8 & 6 \\ 9 & 3 & 1 \\ 4 & 6 & 3 \end{bmatrix}$$

```
> geneqns(N, [x, y, z]); # genera equazioni dalla matrice dei coefficienti
```

$$\{6x + 8y + 6z = 0, 9x + 3y + z = 0, 4x + 6y + 3z = 0\}$$

```
> c := randvector(3, entries=rand(10));
```

$$c := [2, 1, 3]$$

```
> geneqns(N, [x, y, z], c); # genera equazioni dalla matrice dei coefficienti e dal vettore dei termini noti c
```

$$\{6x + 8y + 6z = 2, 9x + 3y + z = 1, 4x + 6y + 3z = 3\}$$

Nucleo, autovalori e autovettori

```
> restart:
> with(linalg):
```

```
Warning, the protected names norm and trace have been redefined and unprotected
```

```
> A := matrix(5, 7, (i, j) -> i+j);
```

$$A := \begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

```
> kernel(A); # restituisce una base del nucleo di A
```

$$\{[1, -2, 1, 0, 0, 0, 0], [2, -3, 0, 1, 0, 0, 0], [4, -5, 0, 0, 0, 1, 0], [5, -6, 0, 0, 0, 0, 1],$$

```
    [3, -4, 0, 0, 1, 0, 0]}
> nullspace(A); # come sopra
{[1, -2, 1, 0, 0, 0, 0], [2, -3, 0, 1, 0, 0, 0], [4, -5, 0, 0, 0, 1, 0], [5, -6, 0, 0, 0, 0, 1],
    [3, -4, 0, 0, 1, 0, 0]}
```