Maxima 在線性代數上之應用

向量空間

國立屏東教育大學 應用數學系 研究助理 徐偉玲

weilinghsu@mail.npue.edu.tw

日期：2009/8/4

除另有說明外，本文件採用創用 CC「姓名標示、非商業性」

2.5 台灣條款

下列出 Maxima 有關矩陣和線性代數的定義或函數，可自行參考：

Definitions for Matrices and Linear Algebra

Function: **ADDCOL** (M, list 1, list 2, ... , list n)　//增加行數於一矩陣 M 中

→addcol(矩陣 M，增加行 1，增加行 2，…，增加行 n)

appends the column(s) given by the one or more lists (or matrices) onto the matrix M.

Function: **ADDROW** (M, list 1, list 2, ..., list n)　//增加列數於一矩陣 M 中

→addrow(矩陣 M，增加列 1，增加列 2，…，增加列 n)

appends the row(s) given by the one or more lists (or matrices) onto the matrix M.

Function: **ADJOINT** (matrix)　//求一矩陣的伴隨矩陣

→adjoint(矩陣)

computes the adjoint of a matrix.

Function: **AUGCOEFMATRIX** ([eq 1, ...], [var 1, ...])　//將一線性方程組之係數轉換

爲擴增矩陣

→augcoefmatrix([方程式 1，方程式 2，…]，[變數 1，變數 2，…])

the augmented coefficient matrix for the variables var1,... of the system of linear equations eq1,.... This is the coefficient matrix with a column adjoined for the constant terms in each equation (i.e. those not dependent upon var1,...).

Function: **CHARPOLY** (M, var)　//求一矩陣 M 的特徵多項式

→charpoly(矩陣 M，變數)

computes the characteristic polynomial for Matrix M with respect to var. That is, DETERMINANT(M - DIAGMATRIX(LENGTH(M),var)).

Function: **COEFMATRIX** ([eq 1, ...], [var 1, ...])　//將線性方程組之係數轉換成矩陣

→coefmatrix([方程式 1，方程式 2，…]，[變數 1，變數 2，…])

the coefficient matrix for the variables var1,... of the system of linear equations eq1,...

Function: **COL** (M, i)　//取出矩陣 A 的第 i 行

→col(矩陣 M，第 i 行)

gives a matrix of the ith column of the matrix M.

Function: **CONJUGATE** (X)　//求矩陣 X 的共軛

→conjugate(矩陣 X)

a function in the EIGEN package on the SHARE directory. It returns the complex conjugate of its argument. This package may be loaded by LOAD(EIGEN); . For a complete description of this package, do PRINTFILE("eigen.usg"); .

Function: **COPYMATRIX** (M)　// 複製一矩陣 M

→copymatrix(矩陣 M)

creates a copy of the matrix M. This is the only way to make a copy aside from recreating M elementwise. Copying a matrix may be useful when SETELMX is used.

Function: **DETERMINANT** (M)　//計算一矩陣的行列式

→determinant(矩陣 M)

computes the determinant of M by a method similar to Gaussian elimination. The form of the result depends upon the setting of the switch RATMX. There is a special routine for dealing with sparse determininants which can be used by setting the switches RATMX:TRUE and SPARSE:TRUE.

Function: **ECHELON** (M)　//將矩陣 M 化成列化簡矩陣

produces the echelon form of the matrix M. That is, M with elementary row operations performed on it such that the first non-zero element in each row in the resulting matrix is a one and the column elements under the first one in each row are all zero.

Function: **EIGENVALUES** (mat)　//計算矩陣的特徵根

→eigenvalues(矩陣 M)

There is a package on the SHARE; directory which contains functions for computing EIGENVALUES and EIGENVECTORS and related matrix computations. For information on it do PRINTFILE(EIGEN,USAGE,SHARE); . EIGENVALUES(mat) takes a MATRIX as its argument and returns a list of lists the first sublist of which is the list of eigenvalues of the matrix and the other sublist of which is the list of the multiplicities of the eigenvalues in the corresponding order. [The MACSYMA function SOLVE is used here to find the roots of the characteristic polynomial of the matrix. Sometimes SOLVE may not be able to find the roots of the polynomial;in that case nothing in this package except CONJUGATE, INNERPRODUCT, UNITVECTOR, COLUMNVECTOR and GRAMSCHMIDT will work unless you know the eigenvalues. In some cases SOLVE may generate very messy eigenvalues. You may want to simplify the answers yourself before you go on. There are provisions for this and they will be explained below. (This usually happens when SOLVE returns a not-so-obviously real expression for an eigenvalue which is supposed to be real...)] The EIGENVALUES command is available directly from MACSYMA. To use the other functions you must have loaded in the EIGEN package, either by a previous call to EIGENVALUES, or by doing LOADFILE("eigen"); .

Function: **EIGENVECTORS** (MAT) //計算矩陣之特徵向量

→eigenvectors(矩陣 M)

takes a MATRIX as its argument and returns a list of lists the first sublist of which is the output of the EIGENVALUES command and the other sublists of which are the eigenvectors of the matrix corresponding to those eigenvalues respectively. This function will work directly from MACSYMA, but if you wish to take advantage of the flags for controlling it (see below), you must first load in the EIGEN package from the SHARE; directory. You may do that by LOADFILE("eigen");. The flags that affect this function are: NONDIAGONALIZABLE[FALSE] will be set to TRUE or FALSE depending on whether the matrix is nondiagonalizable or diagonalizable after an EIGENVECTORS command is executed. HERMITIANMATRIX[FALSE] If set to TRUE will cause the degenerate eigenvectors of the hermitian matrix to be orthogonalized using the Gram-Schmidt algorithm. KNOWNEIGVALS[FALSE] If set to TRUE the EIGEN package will assume the eigenvalues of the matrix are known to the user and stored under the global name LISTEIGVALS. LISTEIGVALS should be set to a list similar to the output of the EIGENVALUES command. (The MACSYMA function ALGSYS is used here to solve for the eigenvectors. Sometimes if the eigenvalues are messy, ALGSYS may not be able to produce a solution. In that case you are advised to try to simplify the eigenvalues by first finding them using EIGENVALUES command and then using whatever marvelous tricks you might have to reduce them to something simpler. You can then use the KNOWNEIGVALS flag to proceed further.)

Function: **EMATRIX** (m, n, x, i, j) //建立 m×n 矩陣中的元素除了 $M_{i,j}$ 是 x 外，其餘元素都是零

will create an m by n matrix all of whose elements are zero except for the i,j element which is x.

Function: **ENTERMATRIX** (m, n)

allows one to enter a matrix element by element with MACSYMA requesting values for each of the m*n entries.

Function: **GENMATRIX** (array, i2, j2, i1, j1)

generates a matrix from the array using array(i1,j1) for the first (upper-left) element and array(i2,j2) for the last (lower-right) element of the matrix. If j1=i1 then j1 may be omitted. If j1=i1=1 then i1 and j1 may both be omitted. If a selected element of the array doesn't exist a symbolic one will be used.

Function: **GRAMSCHMIDT** (X) //先讀取函數 eigen，用 gramschimidt 方法解矩陣 X

a function in the EIGEN package. Do LOAD(EIGEN) to use it. GRAMSCHMIDT takes a LIST of lists the sublists of which are of equal length and not necessarily orthogonal (with respect to the innerproduct defined above) as its argument and returns a similar list each sublist of which is orthogonal to all others. (Returned results may contain integers that are factored. This is due to the fact that the MACSYMA function FACTOR is used to simplify each substage of the Gram-Schmidt algorithm. This prevents the expressions from getting very messy and helps to reduce the sizes of the numbers that are produced along the way.)

Function: **IDENT** (n) //建立一 n×n 的單位矩陣

produces an n by n identity matrix

Function: **INNERPRODUCT** (X,Y) //先讀取函數 eigen，再計算兩向量的內積

a function in the EIGEN package. Do LOAD(EIGEN) to use it. INNERPRODUCT takes two LISTS of equal length as its arguments and returns their inner (scalar) product defined by (Complex Conjugate of X).Y (The "dot" operation is the same as the usual one defined for vectors).

Function: **INVERT** (matrix) //求一矩陣的反矩陣

finds the inverse of a matrix using the adjoint method. This allows a user to compute the inverse of a matrix with bfloat entries or polynomials with floating pt. coefficients without converting to cre-form. The DETERMINANT command is used to compute cofactors, so if RATMX is FALSE (the default) the inverse is computed without changing the representation of the elements. The current implementation is inefficient for matrices of high order. The DETOUT flag if true keeps the determinant factored out of the inverse. Note: the results are not automatically expanded. If the matrix originally had polynomial entries, better appearing output can be generated by EXPAND(INVERT(mat)),DETOUT. If it is desirable to then divide through by the determinant this can be accomplished by XTHRU(%) or alternatively from scratch by
EXPAND(ADJOINT(mat))/EXPAND(DETERMINANT(mat)).
INVERT(mat):=ADJOINT(mat)/DETERMINANT(mat). See also DESCRIBE("^^"); for another method of inverting a matrix.

Function: **MATRIX** (row 1, ..., row n) //定義一矩陣 M
→matrix(第 1 列，…，第 n 列)
defines a rectangular matrix with the indicated rows. Each row has the form of a list of expressions, e.g. [A, X**2, Y, 0] is a list of 4 elements. There are a number of MACSYMA commands which deal with matrices, for example: DETERMINANT, CHARPOLY, GENMATRIX, ADDCOL, ADDROW, COPYMATRIX, TRANSPOSE, ECHELON, and RANK. There is also a package on the SHARE directory for computing EIGENVALUES. Try DESCRIBE on these for more information. Matrix multiplication is effected by using the dot operator, ".", which is also convenient if the user wishes to represent other non-commutative algebraic operations. The exponential of the "." operation is "^^" . Thus, for a matrix A, A.A = A^^2 and, if it exists, A^^-1 is the inverse of A. The operations +,-,*,** are all element-by-element operations; all operations are normally carried out in full, including the . (dot) operation. Many switches exist for controlling simplification rules involving dot and matrix-list operations. Options Relating to Matrices: LMXCHAR, RMXCHAR, RATMX, LISTARITH, DETOUT, DOALLMXOPS, DOMXEXPT DOMXMXOPS, DOSCMXOPS, DOSCMXPLUS, SCALARMATRIX, and SPARSE. Do DESCRIBE(option) for details on them.

Function: **MATRIXMAP** (fn, M)

will map the function fn onto each element of the matrix M.

Function: **MATRIXP** (exp) //判斷 exp 是否爲一矩陣，是爲 TRUE 否爲 FALSE

is TRUE if exp is a matrix else FALSE

Function: **MATTRACE** (M) //求矩陣的 trace，但要先讀取模組 nchrpl

computes the trace [sum of the elements on the main diagonal] of the square matrix M. It is used by NCHARPOLY, an alternative to MACSYMA's CHARPOLY. It is used by doing LOADFILE("nchrpl");

Function: **MINOR** (M, i, j) //M 矩陣去掉第 i 列第 j 行所成的矩陣

computes the i,j minor of the matrix M. That is, M with row i and column j removed.

Function: **RANK** (M) //計算矩陣 M 的秩

computes the rank of the matrix M. That is, the order of the largest non-singular subdeterminant of M. Caveat: RANK may return the wrong answer if it cannot determine that a matrix element that is equivalent to zero is indeed so.

Variable: **RATMX**

default: [FALSE] - if FALSE will cause determinant and matrix addition, subtraction, and multiplication to be performed in the representation of the matrix elements and will cause the result of matrix inversion to be left in general representation. If it is TRUE, the 4 operations mentioned above will be performed in CRE form and the result of matrix inverse will be in CRE form. Note that this may cause the elements to be expanded (depending on the setting of RATFAC) which might not always be desired.

Function: **ROW** (M, i) //取出矩陣 M 中的第 i 列

gives a matrix of the ith row of matrix M.

Variable: **SCALARMATRIXP**

default: [TRUE] - if TRUE, then whenever a 1 x 1 matrix is produced as a result of computing the dot product of matrices it will be converted to a scalar, namely the only element of the matrix. If set to ALL, then this conversion occurs whenever a 1 x 1 matrix is simplified. If set to FALSE, no conversion will be done.

Function: **SETELMX** (x, i, j, M)

changes the i,j element of M to x. The altered matrix is returned as the value. The notation M[i,j]:x may also be used, altering M in a similar manner, but returning x as the value

Function: **SIMILARITYTRANSFORM** (MAT)

a function in the EIGEN package. Do LOAD(EIGEN) to use it. SIMILARITYTRANSFORM takes a MATRIX as its argument and returns a list which is the output of the UNITEIGENVECTORS command. In addition if the flag NONDIAGONALIZABLE is FALSE two global matrices LEFTMATRIX and RIGHTMATRIX will be generated. These matrices have the property that LEFTMATRIX.MAT.RIGHTMATRIX is a diagonal matrix with the eigenvalues of MAT on the diagonal. If NONDIAGONALIZABLE is TRUE these two matrices will not be generated. If the flag HERMITIANMATRIX is TRUE then LEFTMATRIX is the complex conjugate of the transpose of RIGHTMATRIX. Otherwise LEFTMATRIX is the inverse of RIGHTMATRIX. RIGHTMATRIX is the matrix the columns of which are the unit eigenvectors of MAT. The other flags (see DESCRIBE(EIGENVALUES); and DESCRIBE(EIGENVECTORS);) have the same effects since SIMILARITYTRANSFORM calls the other functions in the package in order to be able to form RIGHTMATRIX.

Variable: **SPARSE**

default: [FALSE] - if TRUE and if RATMX:TRUE then DETERMINANT will use special routines for computing sparse determinants.

Function: **SUBMATRIX** (m1, ..., M, n1, ...)

creates a new matrix composed of the matrix M with rows mi deleted, and columns ni deleted.

Function: **TRANSPOSE** (M) //矩陣 M 的轉置

produces the transpose of the matrix M.

Function: **TRIANGULARIZE** (M) //將矩陣 M 上三角化且矩陣 M 不需爲方正矩陣

produces the upper triangular form of the matrix M which needn't be square.

Function: **ZEROMATRIX** (m,n)   //建立一 m×n 的零矩陣

takes integers m,n as arguments and returns an m by n matrix of 0's.

第一章 向量空間

1.1 引言

1.判斷由原點引出而終點在下列指定二點所形成的向量是否平行

(a) (3,1,2)及(6,4,2)

(b) (-3,1,7)及(9,-3,-21)

(c) (5,-6,7)及(-5,6,-7)

(d) (2,0,-5)及(5,0,-2)

2.求通過下列空間上的二點所形成的直線方程式

(a) (3,-2,4)及(-5,7,1)

(b) (2,4,0)及(-3,-6,0)

(c) (3,7,2)及(3,7,-8)

(d) (-2,-1,5)及(3,9,7)

3.求包含下列空間上的三點所形成的平面方程式

(a) (2,-5,-1), (0,4,6)及(-3,7,1)

(b) (3,-6,7), (-2,0,-4)及(5,-9,-2)

(c) (-8,2,0), (1,3,0)及(6,-5,0)

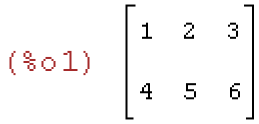
(d) (1,1,1), (5,5,5)及(-6,4,2)

1.2 向量空間

3. 若 M=$\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{pmatrix}$, 則 M_{13}, M_{21}, M_{22} 爲何?

```
(%i1) M:matrix([1,2,3],[4,5,6]);    //定義一矩陣，矩陣名稱定作 M，第一列爲 1,2,3，第二列爲 4,5,6
```

此矩陣取名爲 M，matrix 是「矩陣」的意思，以中括號表示「列」，將矩陣取名是爲了方便之後的讀取



```
(%o1) [ 1  2  3 ]
      [         ]
      [ 4  5  6 ]
```

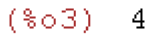
```
(%i2) M[1,3];    //取矩陣 M 中第 1 列第 3 行的元素
```

讀取上面所定義的矩陣名稱 M，1 代表是第 1 列，3 代表是第 3 行

```
(%o2) 3
```

(%i3) M[2,1];　//取矩陣 M 中第 2 列第 1 行的元素

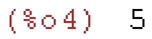
＊讀取上面所定義的矩陣名稱 M，2 代表是第 2 列，1 代表是第 1 行



```
(%o3)  4
```

(%i4) M[2,2];　//矩陣 M 中第 2 列第 2 行的元素

＊讀取上面所定義的矩陣名稱 M，2 代表是第 2 列，2 代表是第 2 行

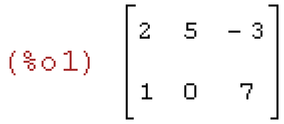


```
(%o4)  5
```

4.完成下列指定的運算

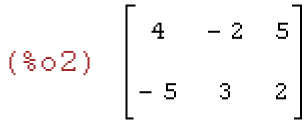
(a) $\begin{pmatrix} 2 & 5 & -3 \\ 1 & 0 & 7 \end{pmatrix} + \begin{pmatrix} 4 & -2 & 5 \\ -5 & 3 & 2 \end{pmatrix}$

(%i1) A:matrix([2,5,-3],[1,0,7]);　//矩陣名稱取為 A，第一列的元素有 2, 5, -3，第二列的元素為 1, 0, 7



```
        [ 2  5  - 3 ]
(%o1)   [           ]
        [ 1  0    7 ]
```

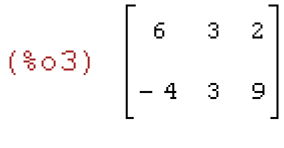
(%i2) B:matrix([4,-2,5],[-5,3,2]);　//矩陣名稱取為 B，第一列的元素有 4, -2, 5，第二列的元素有-5, 3, 2



```
        [  4   - 2  5 ]
(%o2)   [             ]
        [ - 5    3  2 ]
```

(%i3) A+B;　//將上面所定義的矩陣 A 和矩陣 B 相加

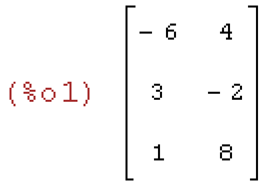
將矩陣制定名稱的好處是作運算時只要呼叫函數的名稱即可，不須把所有指令打上去



```
        [  6   3  2 ]
(%o3)   [           ]
        [ - 4  3  9 ]
```

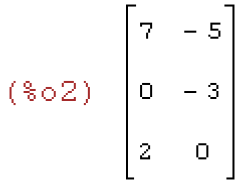
(b) $\begin{pmatrix} -6 & 4 \\ 3 & -2 \\ 1 & 8 \end{pmatrix} + \begin{pmatrix} 7 & -5 \\ 0 & -3 \\ 2 & 0 \end{pmatrix}$

(%i1) A:matrix([-6,4],[3,-2],[1,8]); //矩陣名稱取爲 A，第一列的元素有-6, 4，第二列的元素爲 3, -2，第 3 列的元素爲 1, 8



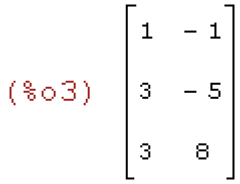
```
       [ - 6   4  ]
       [          ]
(%o1)  [  3   - 2 ]
       [          ]
       [  1    8  ]
```

(%i2) B:matrix([7,-5],[0,-3],[2,0]); //矩陣名稱取爲 B，第一列的元素有 7, -5，第二列的元素爲 0, -3，第 3 列的元素爲 2, 0



```
       [ 7  - 5 ]
       [        ]
(%o2)  [ 0  - 3 ]
       [        ]
       [ 2   0  ]
```

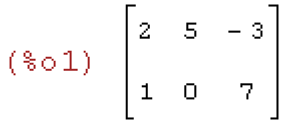
(%i3) A+B; //將上面所定義的矩陣 A 和矩陣 B 相加



```
       [ 1  - 1 ]
       [        ]
(%o3)  [ 3  - 5 ]
       [        ]
       [ 3   8  ]
```

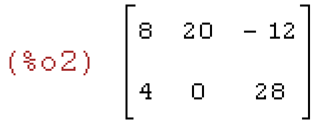
(c) $4\begin{pmatrix} 2 & 5 & -3 \\ 1 & 0 & 7 \end{pmatrix}$

(%i1) A:matrix([2,5,-3],[1,0,7]);　 //矩陣名稱取爲 A，第一列的元素有 2, 5, -3，第二列的元素爲 1, 0, 7



```
       [ 2  5  - 3 ]
(%o1)  [           ]
       [ 1  0   7  ]
```

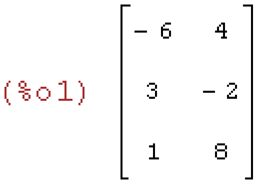
(%i2) 4*A;　 //矩陣 A 乘上純量 4 矩陣的純量乘法，純量乘法用「*」符號，若矩陣相乘則用「.」的符號



```
       [ 8  20  - 12 ]
(%o2)  [             ]
       [ 4   0   28  ]
```

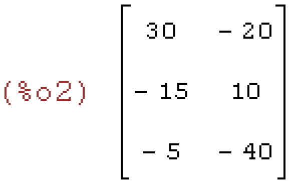
(d) $-5\begin{pmatrix} -6 & 4 \\ 3 & -2 \\ 1 & 8 \end{pmatrix}$

(%i1) A:matrix([-6,4],[3,-2],[1,8]); //矩陣名稱取爲 A，第一列的元素有-6, 4，第二列的元素爲 3, -2，第三列的元素爲 1, 8



```
         [ - 6   4  ]
         [          ]
(%o1)    [  3   - 2 ]
         [          ]
         [  1    8  ]
```

(%i2) -5*A;　//矩陣的純量乘法，矩陣 A 乘上純量-5，純量乘法用「*」符號，若矩陣相乘則用「.」的符號

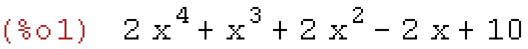


```
         [  30    - 20 ]
         [             ]
(%o2)    [ - 15    10  ]
         [             ]
         [ - 5    - 40 ]
```

(e) $(2x^4-7x^3+4x+3)+(8x^3+2x^2-6x+7)$

多項式的相加，乘法的符號爲「*」而次方的符號爲「^」，例如 2 乘上 x 以 2*x 表示，而 x^4 是以 x^4 表示，以此類推。

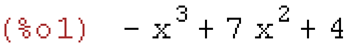
(%i1) (2*x^4-7*x^3+4*x+3)+(8*x^3+2*x^2-6*x+7);



```
            4    3      2
(%o1)   2 x  + x  + 2 x  - 2 x + 10
```

(f) $(-3x^3+7x^2+8x-6)+(2x^3-8x+10)$　//多項式-3*x^3+7*x^2+8*x-6 加上多項式 2*x^3-8*x+10

(%i1) (-3*x^3+7*x^2+8*x-6)+(2*x^3-8*x+10);



```
             3      2
(%o1)   - x  + 7 x  + 4
```

(g) $5(2x^7-6x^4+8x^2-3x)$ //多項式的純量乘法，多項式 2*x^7-6*x^4+8*x^2-3*x 乘上 5，也是用「*」代表乘法

(%i1) 5*(2*x^7-6*x^4+8*x^2-3*x);

(%o1) $5\left(2x^7-6x^4+8x^2-3x\right)$

(%i2) expand((5*(2*x^7-6*x^4+8*x^2-3*x))); //多項式的展開 因為所做的乘法運算 maxima 不會自動展開，它只會做整理的動作，所以用 expand 展開運算式就是所要的答案了

(%o2) $10x^7-30x^4+40x^2-15x$

(h) $3(x^5-2x^3+4x+2)$ //多項式 x^5-2*x^3+4*x+2 乘上純量 3

(%i1) 3*(x^5-2*x^3+4*x+2);

(%o1) $3\left(x^5-2x^3+4x+2\right)$

(%i2) expand(3*(x^5-2*x^3+4*x+2)); //用 expand 展開運算式

(%o2) $3x^5-6x^3+12x+6$

1.3 子空間

2.決定下列矩陣的轉置矩陣，此外，若它為方陣，試求其跡數

(a) $\begin{pmatrix}-4 & 2\\ 5 & -1\end{pmatrix}$

(%i1) load("nchrpl"); //Maxima 本身是「不會」算 trace 的，我們可以用適當的模組來做這件事，所謂模組就是一段小程式，通常是增加一些指令，為什麼 Maxima 一開始就把這些模組都加進來？那是因為一來太佔記憶體，一來是很多對某些人很重要的指令也許你永遠也沒用到。我們要算一矩陣的 trace 要使用 nchrpl 這個模組，此模組提供了 mattrace 指令去計算 trace。讀入 nchrpl 模組，接著就可以使用這個模組提供的指令。

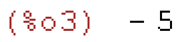
(%o1)

C:/PROGRA~1/MAXIMA~1.1/share/maxima/5.18.1/share/matrix/nchrpl.mac

(%i2) A: matrix([-4,2],[5,-1]); //矩陣名稱為 A，第一列的元素有-4, 2，第二列的元素有 5, -1

(%o2) $\begin{bmatrix}-4 & 2\\ 5 & -1\end{bmatrix}$

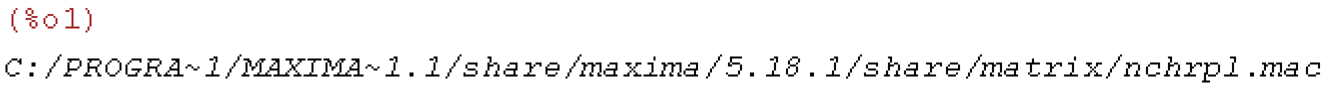
(%i3) mattrace(A); 由於已經讀取 nchrpl 模組，因此我們可以使用這個模組所提供的指令打上 mattrace(矩陣)來計算矩陣 A 的跡數



```
(%o3)  - 5
```

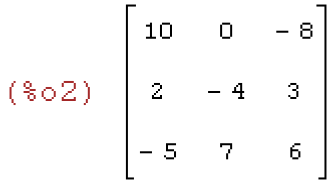
(d) $\begin{pmatrix} 10 & 0 & -8 \\ 2 & -4 & 3 \\ -5 & 7 & 6 \end{pmatrix}$

(%i1) load("nchrpl"); //讀入 nchrpl 模組，接著就可以使用這個模組提供的指令。



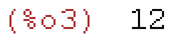
```
(%o1)
C:/PROGRA~1/MAXIMA~1.1/share/maxima/5.18.1/share/matrix/nchrpl.mac
```

(%i2) A: matrix([10,0,-8],[2,-4,3],[-5,7,6]); //矩陣名稱爲 A，第一列的元素有 10, 0, -8，第二列的元素有 2, -4, 3，第三列的元素有-5, 7, 6



```
        [ 10   0   - 8 ]
        [              ]
(%o2)   [ 2   - 4   3  ]
        [              ]
        [ - 5  7    6  ]
```

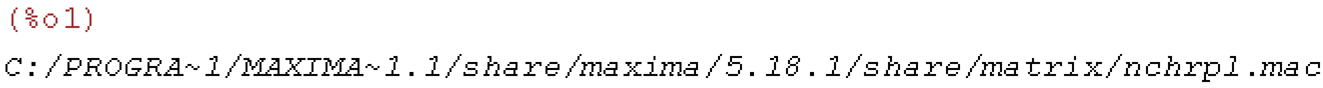
(%i3) mattrace(A); //由於已經讀取 nchrpl 模組，因此我們可以使用這個模組所提供的指令打上 mattrace(矩陣)來計算矩陣 A 的跡數



```
(%o3)  12
```

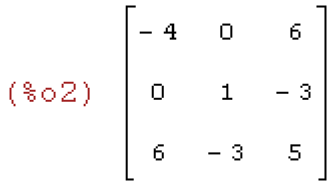
(h) $\begin{pmatrix} -4 & 0 & 6 \\ 0 & 1 & -3 \\ 6 & -3 & 5 \end{pmatrix}$

(%i1) load("nchrpl"); //讀入 nchrpl 模組，接著就可以使用這個模組提供的指令。



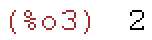
```
(%o1)
C:/PROGRA~1/MAXIMA~1.1/share/maxima/5.18.1/share/matrix/nchrpl.mac
```

(%i2) A: matrix([-4,0,6],[0,1,-3],[6,-3,5]);　//矩陣名稱為 A，第一列的元素有-4, 0, 6，第二列的元素有 0, 1, -3，第三列的元素有 6, -3, 5



```
(%o2) [ -4   0   6 ]
      [  0   1  -3 ]
      [  6  -3   5 ]
```

(%i3) mattrace(A);　//由於已經讀取 nchrpl 模組，因此我們可以使用這個模組所提供的指令打上 mattrace(矩陣)來計算矩陣 A 的跡數



```
(%o3)  2
```

1.4 線性組合與線性方程組

2. 解下列線性方程組

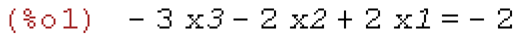
(a) $2x_1$-$2x_2$-$3x_3$=-2

$3x_1$-$3x_2$-$2x_3$+$5x_4$=7

x_1-x_2-$2x_3$-x_4=-3

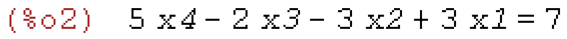
(%i1) eq1:2*x1-2*x2-3*x3=-2;　解線性方程組有 2 種表示方式，第一種為先建立好方程式 1，方程式 2，…，方程式 n，再用 solve([方程式 1 名稱，方程式 2 名稱，…，方程式 n 名稱]，[未知數 1，未知數 2，…，未知數 n])即可解線性方程組;而第二種方式為將方程式和未知數一起併入,而不分開表示,指令為 solve([方程式 1，方程式 2，…，方程式 n]，[未知數 1，未知數 2，…，未知數 n])

//方程式名稱定為 eq1，方程式 1 為 $2x_1$-$2x_2$-$3x_3$=-2



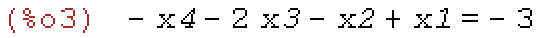
```
(%o1)  - 3 x3 - 2 x2 + 2 x1 = - 2
```

(%i2) eq2:3*x1-3*x2-2*x3+5*x4=7;　//方程式名稱定為 eq2，方程式 2 為 $3x_1$-$3x_2$-$2x_3$+$5x_4$=7



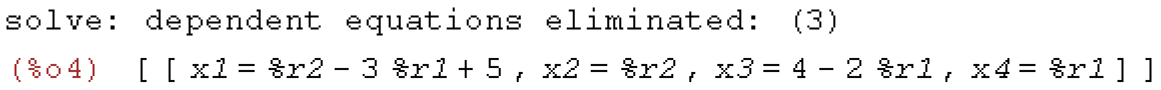
```
(%o2)  5 x4 - 2 x3 - 3 x2 + 3 x1 = 7
```

(%i3) eq3:x1-x2-2*x3-x4=-3;　//方程式名稱定為 eq3，方程式 3 為 x_1-x_2-$2x_3$-x_4=-3



```
(%o3)  - x4 - 2 x3 - x2 + x1 = - 3
```

(%i4) solve([eq1,eq2,eq3],[x1,x2,x3,x4]); //用 slove 解線性方程組 eq1，eq2，eq3 和未知數 x1，未知數 x2，未知數 x3，未知數 x4，求得的解由於方程式爲線性相依的，%r1、%r2 可以看成是任意數，因此 x4=a，則 x3=4-2a，又 x2=b，x1=b-3a+5，x3 和 x1 的值是會隨著 x4 和 x2 變動的



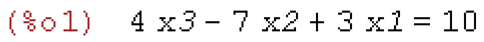
```
solve: dependent equations eliminated: (3)
(%o4)  [ [ x1 = %r2 - 3 %r1 + 5 , x2 = %r2 , x3 = 4 - 2 %r1 , x4 = %r1 ] ]
```

(b) $3x_1-7x_2+4x_3=10$

$x_1-2x_2+x_3=3$

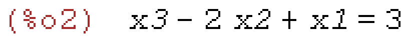
$2x_1-x_2-2x_3=6$

(%i1) eq1:3*x1-7*x2+4*x3=10; //方程式名稱定爲 eq1，方程式 1 爲 $3x_1-7x_2+4x_3=10$



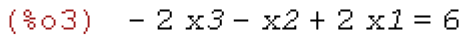
```
(%o1)  4 x3 - 7 x2 + 3 x1 = 10
```

(%i2) eq2:x1-2*x2+x3=3; //方程式名稱定爲 eq2，方程式 2 爲 $x_1-2x_2+x_3=3$



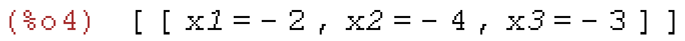
```
(%o2)  x3 - 2 x2 + x1 = 3
```

(%i3) eq3:2*x1-x2-2*x3=6; //方程式名稱定爲 eq3，方程式 3 爲 $2x_1-x_2-2x_3=6$



```
(%o3)  - 2 x3 - x2 + 2 x1 = 6
```

(%i4) solve([eq1,eq2,eq3],[x1,x2,x3]); //用 slove 解線性方程組 eq1，eq2，eq3 和未知數 x1，未知數 x2，未知數 x3，求得的解爲 x1=-2，x2=-4，x3=-3 是唯一解



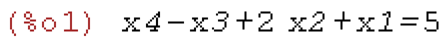
```
(%o4)  [ [ x1 = - 2 , x2 = - 4 , x3 = - 3 ] ]
```

(c) $x_1+2x_2-x_3+x_4=5$

$x_1+4x_2-3x_3-3x_4=6$

$2x_1+3x_2-x_3+4x_4=8$

(%i1) eq1:x1+2*x2-x3+x4=5; //方程式名稱定爲 eq1，方程式 1 爲 $x_1+2x_2-x_3+x_4=5$



```
(%o1)  x4 - x3 + 2 x2 + x1 = 5
```

(%i2) eq2:x1+4*x2-3*x3-3*x4=6;　//方程式名稱定爲 eq2，方程式 2 爲 $x_1+4x_2-3x_3-3x_4=6$

```
(%o2) -3 x4-3 x3+4 x2+x1=6
```

(%i3) eq3:2*x1+3*x2-x3+4*x4=8;　//方程式名稱定爲 eq3，方程式 3 爲 $2x_1+3x_2-x_3+4x_4=8$

```
(%o3) 4 x4-x3+3 x2+2 x1=8
```

(%i4) solve([eq1,eq2,eq3],[x1,x2,x3,x4]);　//此線性系統無解

```
(%o4) []
```

(d) $x_1+2x_2+2x_3=2$

$x_1+8x_3+5x_4=-6$

$x_1+x_2+5x_3+5x_4=3$

(%i1) eq1:x1+8*x3+5*x4=-6;　//方程式名稱定爲 eq1，方程式 1 爲 $x_1+8x_3+5x_4=-6$

```
(%o1) 5 x4+8 x3+x1=-6
```

(%i2) eq2:x1+x2+5*x3+5*x4=3;　//方程式名稱定爲 eq2，方程式 2 爲 $x_1+x_2+5x_3+5x_4=3$

```
(%o2) 5 x4+5 x3+x2+x1=3
```

(%i3) eq3:x1+2*x2+2*x3=2;　//方程式名稱定爲 eq3，方程式 3 爲 $x_1+2x_2+2x_3=2$

```
(%o3) 2 x3+2 x2+x1=2
```

(%i4) solve([eq1,eq2,eq3],[x1,x2,x3,x4]);　//用 slove 解線性方程組 eq1，eq2，eq3 和未知數 x1，未知數 x2，未知數 x3，未知數 x4，求得的解由於方程式爲線性相依的，%r3 可以看成是任意數，因此 x4=2，則 x3=-(%r3+16)/8，又 x2=-(3%r3-24)/8，x1=%r3

```
                      3 %r3-24        %r3+16
(%o4) [[x1=%r3, x2=-----------, x3=-----------, x4=2]]
                          8              8
```

(e) $x_1+2x_2-4x_3-x_4+x_5=7$

$-x_1+10x_3-3x_4-4x_5=-16$
$2x_1+5x_2-5x_3-4x_4-x_5=2$
$4x_1+11x_2-7x_3-10x_4-2x_5=7$

(%i1) eq1:x1+2*x2-4*x3-x4+x5=7; //方程式名稱定爲 eq1，方程式 1 爲 $x_1+2x_2-4x_3-x_4+x_5=7$

```
(%o1) x5-x4-4 x3+2 x2+x1=7
```

(%i2) eq2:-x1+10*x3-3*x4-4*x5=-16; //方程式名稱定爲 eq2，方程式 2 爲 $x_1+10x_3-3x_4-4x_5=-16$

```
(%o2) -4 x5-3 x4+10 x3-x1=-16
```

(%i3) eq3:2*x1+5*x2-5*x3-4*x4-x5=2; //方程式名稱定爲 eq3，方程式 3 爲 $2x_1+5x_2-5x_3-4x_4-x_5=2$

```
(%o3) -x5-4 x4-5 x3+5 x2+2 x1=2
```

(%i4) eq4:4*x1+11*x2-7*x3-10*x4-2*x5=7; //方程式名稱定爲 eq4，方程式 4 爲 $4x_1+11x_2-7x_3-10x_4-2x_5=7$

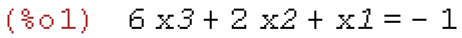
```
(%o4) -2 x5-10 x4-7 x3+11 x2+4 x1=7
```

(%i5) solve([eq1,eq2,eq3,eq4],[x1,x2,x3,x4,x5]); //用 slove 解線性方程組 eq1，eq2，eq3，eq4 和未知數 x1，未知數 x2，未知數 x3，未知數 x4，未知數 x5，求得的解由於方程式爲線性相依的，%r1、%r2 可以看成是任意數，因此 x5=5，x4=%r2，則 x3=%r1，又 x2=2%r2-3%r1+3，x1=-3%r2+10%r1-4

```
solve: dependent equations eliminated: (4)
 (%o5) [[x1=-3 %r2+10 %r1-4, x2=2 %r2-3 %r1+3, x3=%r1, x4=
%r2, x5=5]]
```

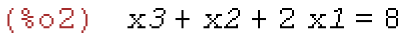
(f) $x_1+2x_2+6x_3=-1$
$2x_1+x_2+x_3=8$
$3x_1+x_2-x_3=15$
$x_1+3x_2+10x_3=-5$

(%i1) eq1:x1+2*x2+6*x3=-1;　//方程式名稱定爲 eq1，方程式 1 爲 $x_1+2x_2+6x_3=-1$



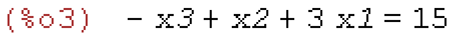
```
(%o1)   6 x3 + 2 x2 + x1 = - 1
```

(%i2) eq2:2*x1+x2+x3=8;　//方程式名稱定爲 eq2，方程式 2 爲 $2x_1+x_2+x_3=8$



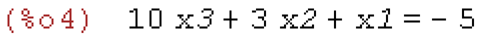
```
(%o2)   x3 + x2 + 2 x1 = 8
```

(%i3) eq3:3*x1+x2-x3=15;　//方程式名稱定爲 eq3，方程式 3 爲 $3x_1+x_2-x_3=15$



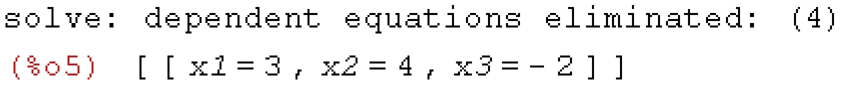
```
(%o3)   - x3 + x2 + 3 x1 = 15
```

(%i4) eq4:x1+3*x2+10*x3=-5;　//方程式名稱定爲 eq4，方程式 4 爲 $x_1+3x_2+10x_3=-5$



```
(%o4)   10 x3 + 3 x2 + x1 = - 5
```

(%i5) solve([eq1,eq2,eq3,eq4],[x1,x2,x3]); //用 slove 解線性方程組 eq1，eq2，eq3，eq4 和未知數 x1，未知數 x2，未知數 x3，由於方程組爲線性相依，但可求出一解爲 x1=3，x2=4 和 x3=-2



```
solve: dependent equations eliminated: (4)
(%o5)   [ [ x1 = 3 , x2 = 4 , x3 = - 2 ] ]
```

5.試決定下面各小題所給的向量是否在 S 的生成集裡

(a) (2,-1,1), S={(1,0,2), (-1,1,1)}

(b) (-1,2,1), S={(1,0,2), (-1,1,1)}

(c) (-1,1,1,2), S={(1,0,1,-1), (0,1,1,1)}

(d) (2,-1,1,-3), S={(1,0,1,-1), (0,1,1,1)}

(e) $-x^3+2x^2+3x+3$, S=$\{x^3+x^2+x+1, x^2+x+1, x+1\}$

(f) $2x^3-x^2+x+3$, S=$\{x^3+x^2+x+1,\ x^2+x+1,\ x+1\}$

(g) $\begin{pmatrix}1 & 2\\ -3 & 4\end{pmatrix}$, S=$\{\begin{pmatrix}1 & 0\\ -1 & 0\end{pmatrix}, \begin{pmatrix}0 & 1\\ 0 & 1\end{pmatrix}, \begin{pmatrix}1 & 1\\ 0 & 0\end{pmatrix}\}$

(h) $\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$, S=$\{\begin{pmatrix}1 & 0\\ -1 & 0\end{pmatrix}, \begin{pmatrix}0 & 1\\ 0 & 1\end{pmatrix}, \begin{pmatrix}1 & 1\\ 0 & 0\end{pmatrix}\}$

1.5　線性獨立與線性相依

2. 試決定下面集合是否爲線性相依或線性獨立

(a) $\{\begin{pmatrix} 1 & -3 \\ -2 & 4 \end{pmatrix}, \begin{pmatrix} -2 & 6 \\ 4 & -8 \end{pmatrix}\}$在 $M_{2\times2}(R)$

(b) $\{\begin{pmatrix} 1 & -2 \\ -1 & 4 \end{pmatrix}, \begin{pmatrix} -1 & 1 \\ 2 & -4 \end{pmatrix}\}$在 $M_{2\times2}(R)$

(c) $\{x^3+2x^2, -x^2+3x+1, x^3-x^2+2x-1\}$在 $P_3(R)$

(d) $\{x^3-x, 2x^2+4, -2x^3+3x^2+2x+6\}$在 $P_3(R)$

(e) {(1,-1,2), (1,-2,1), (1,1,4)}在 R^3

(f) {(1,-1,2), (2,0,1), (-1,2,-1)}在 R^3

(g) $\{\begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}, \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} -1 & 2 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 2 & 1 \\ -4 & 4 \end{pmatrix}\}$在 $M_{2\times2}(R)$

(h) $\{\begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}, \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} -1 & 2 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 2 & 1 \\ 2 & -2 \end{pmatrix}\}$在 $M_{2\times2}(R)$

(i) $\{x^4-x^3+5x^2-8x+6, -x^4+x^3-5x^2+5x-3, x^4+3x^2-3x+5, 2x^4+3x^3+4x^2-x+1, x^3-x+2\}$在 $P_4(R)$

(j) $\{x^4-x^3+5x^2-8x+6, -x^4+x^3-5x^2+5x-3, x^4+3x^2-3x+5, 2x^4+x^3+4x^2+8x\}$在 $P_4(R)$

1.6 基底與維度

2.試判斷下列集合是否爲 R^3的基底

(a) {(1,0,-1), (2,5,1), (0,-4,3)}

(b) {(2,-4,1), (0,3,-1), (6,0,-1)}

(c) {(1,2,-1), (1,0,2), (2,1,1)}

(d) {(-1,3,1), (2,-4,-3), (-3,8,2)}

(e) {(1,-3,-2), (-3,1,3), (-2,-10,-2)}

3.試判斷下列集合是否爲 $P_2(R)$的基底

(a) $\{-1-x+2x^2, 2+x-2x^2, 1-2x+4x^2\}$

(b) $\{1+2x+x^2, 3+x^2, x+x^2\}$

(c) $\{1-2x-2x^2, -2+3x-x^2, 1-x+6x^2\}$

(d) $\{-1+2x+4x^2, 3-4x-10x^2, -2-5x-6x^2\}$

(e) $\{1+2x-x^2, 4-2x+x^2, -1+18x-9x^2\}$

4.多項式 x^3-2x^2+1, $4x^2-x+3$, 及 3x-2 是否生成 $P_3(R)$?說明你的答案

1.7 極大線性獨立子集