Maxima 在線性代數上之應用

內積空間

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：2009/8/19

第六章 內積空間

6.1 內積與範數

2.令 x=(2, 1+i, i)及 y=(2-i, 2, 1+2i)為 C^3 上的向量，試求<x,y>，$\|x\|$，$\|y\|$,及$\|x+y\|$，

再驗證柯西不等式及三角不等式

```
(%i1) load("eigen");
(%o1) C:/PROGRA~1/MAXIMA~1.2/share/maxima/5.19.2/share/matrix/eigen.mac

(%i2) x:[2,1+%i,%i];
(%o2) [2, %i+1, %i]

(%i3) inprod(x,x);
(%o3) 7

(%i4) y:[2-%i,2,1+2*%i];
(%o4) [2-%i, 2, 2 %i+1]

(%i5) inprod(y,y);
(%o5) 14

(%i6) x+y;
(%o6) [4-%i, %i+3, 3 %i+1]

(%i7) inprod(x+y,x+y);
(%o7) 37
```

3.在 C([0, 1])上，令 f(t)=t 及 g(t)=e^t，試求<f,g>(如例 3 所定義)，$\|f\|$，$\|g\|$，及

$\|f+g\|$，再驗證柯西不等式及三角不等式

4. (b)使用 Frobenius 內積計算$\|A\|$，$\|B\|$,及<A,B>，其中 A=$\begin{pmatrix} 1 & 2+i \\ 3 & i \end{pmatrix}$及

B=$\begin{pmatrix} 1+i & 0 \\ i & -i \end{pmatrix}$

(%i1) load("eigen");

(%o1)
C:/PROGRA~1/MAXIMA~1.2/share/maxima/5.19.2/share/matrix/eigen.mac

(%i2) A:matrix([1,2+%i],[3,%i]);

(%o2) $\begin{bmatrix} 1 & \%i+2 \\ 3 & \%i \end{bmatrix}$

(%i3) mat_norm(A,frobenius);

(%o3) 4

(%i4) B:matrix([1+%i,0],[%i,-%i]);

(%o4) $\begin{bmatrix} \%i+1 & 0 \\ \%i & -\%i \end{bmatrix}$

(%i5) mat_norm(B,frobenius);

(%o5) 2

6.2 Gram-Schmidt 正交化步驟及正交補集

2.下列各題中，對內積空間 V 的已知子集 S，應用 Gram-Schmidt 步驟，求一正交基底給 span(S)，再對此基底上的每一向量做正規化，以求一組單範正交基底β給 span(S)，且對所給的向量，求相對於β的傅笠爾係數，最後，利用定理 6.5 驗證你的結果

(a) V=R^3, S={(1, 0, 1),(0, 1, 1),(1, 3, 3)}，及 x={(1, 1, 2)}

(%i1) load(eigen)$ Maxima 本身是不會是不會算 gramschmidt 和 innerproduct 的，我們可以使用適當的模組來做這件事，所謂模組就是一小段程式，通常是增加一些指令供你使用，爲何 Maxima 不一開始就把這些模組都加進來，那是因爲如此一來太佔用記憶體，於是我們要算一向量的 innerproduct 和 gramschmidt，

要使用 eigen 這個模組　//讀取 eigen 這個 package，這個模組提供了 inprod 指令去算 innerproduct 及 gramschmidt 指令去算 gramschmidt 正交化

(%i2) x:matrix([1,0,1],[0,1,1],[1,3,3]);　//定義一矩陣，矩陣名稱叫做 x，第一列元素有 1, 0, 1，第二列元素有 0, 1, 1，第三列元素有 1, 3, 3

$$(\%o2)\quad \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 3 & 3 \end{bmatrix}$$

(%i3) y:gramschmidt(x);　Gram-Schmidt 單範正交化指令：gramschmidt(矩陣)，取出 x 矩陣中的列向量去做 Gram-Schmidt 步驟來計算正交向量在將這些向量正規化以得一單範正交基底 //對矩陣 x 作 Gram-Schmidt 正交化得出單範正交集名稱叫做 y

$$(\%o3)\quad [[1,0,1],[-\frac{1}{2},1,\frac{1}{2}],[\frac{1}{3},\frac{1}{3},-\frac{1}{3}]]$$

(%i4) map(innerproduct,[y[1],y[2],y[3]],[y[2],y[3],y[1]]);　//驗證上列結果是否正確，於是驗證矩陣 y 中的列向量內積爲 0，表各自垂直，y[1]代表矩陣 y 中的第一列，以此類推

(%o4) [0, 0, 0]

(%i5) inprod(y[1],y[1]);　內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字　//矩陣 y 中第一列[1,0,1]和矩陣 y 中第一列[1,0,1]自己的內積

(%o5) 2

(%i6) inprod(y[2],y[2]);　內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字　//矩陣 y 中第二列$[-\frac{1}{2},1,\frac{1}{2}]$和矩陣 y 中第二列$[-\frac{1}{2},1,\frac{1}{2}]$自己的內積

$$(\%o6)\quad \frac{3}{2}$$

(%i7) inprod(y[3],y[3]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //矩陣 y 中第三列$[\frac{1}{3},\frac{1}{3},-\frac{1}{3}]$和矩陣 y 中第三列$[\frac{1}{3},\frac{1}{3},-\frac{1}{3}]$自己的內積

(%o7) $\frac{1}{3}$

(%i8) a[1]:inprod([1,1,2],1/sqrt(2)*[1,0,1]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //[1,1,2]和$\frac{1}{\sqrt{2}}[1,0,1]$的內積，名稱給定為 a[1]

(%o8) $\frac{3}{\sqrt{2}}$

(%i9) a[2]:inprod([1,1,2],sqrt(2)/sqrt(3)*[-1/2,1,1/2]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //[1,1,2]和$\frac{\frac{1}{\sqrt{3}}}{\sqrt{2}}[-\frac{1}{2},1,\frac{1}{2}]$的內積，名稱給定為 a[2]

(%o9) $\frac{\sqrt{3}}{\sqrt{2}}$

(%i10) a[3]:inprod([1,1,2],sqrt(3)*[1/3,1/3,-1/3]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //[1,1,2]和$\frac{1}{\frac{1}{\sqrt{3}}}[\frac{1}{3},\frac{1}{3},-\frac{1}{3}]$的內積，名稱給定為 a[3]

(%o10) 0

(%i11)
a[1]*1/sqrt(2)*[1,0,1]+a[2]*sqrt(2)/sqrt(3)*[-1/2,1,1/2]+a[3]*sqrt(3)*[1/3,1/3,1/3]; //用課本定理 6.5 去驗證結果 (定理 6.5：$T(v_j)=\sum_{i=1}^{n}<T(v_j),v_i>v_i$)

(%o11) [1 , 1 , 2]

(b) $V=R^3$，S={(1,1,1),(0,1,1),(0,0,1)}，及 x=(1,0,1)

(%i1) load(eigen)$ Maxima 本身是不會是不會算 gramschmidt 和 innerproduct 的，我們可以使用適當的模組來做這件事，所謂模組就是一小段程式，通常是增加一些指令供你使用，爲何 Maxima 不一開始就把這些模組都加進來，那是因爲如此一來太佔用記憶體，於是我們要算一向量的 innerproduct 和 gramschmidt，要使用 eigen 這個模組 //讀取 eigen 這個 package，這個模組提供了 inprod 指令去算 innerproduct 及 gramschmidt 指令去算 gramschmidt 正交化

(%i2) x:matrix([1,1,1],[0,1,1],[0,0,1]); //定義一矩陣，矩陣名稱叫做 x，第一列元素有 1, 1, 1，第二列元素有 0, 1, 1，第三列元素有 0, 0, 1

(%o2) $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

(%i3) y:gramschmidt(x); Gram-Schmidt 單範正交化指令：gramschmidt(矩陣)，取出 x 矩陣中的列向量去做 Gram-Schmidt 步驟來計算正交向量在將這些向量正規化以得一單範正交基底 //對矩陣 x 作 Gram-Schmidt 正交化得出單範正交集名稱叫做 y

(%o3) $[[1,1,1],[-\frac{2}{3},\frac{1}{3},\frac{1}{3}],[0,-\frac{1}{2},\frac{1}{2}]]$

(%i4) map(innerproduct,[y[1],y[2],y[3]],[y[2],y[3],y[1]]); //驗證上列結果是否正確，於是驗證矩陣 y 中的列向量內積爲 0，表各自垂直，y[1]代表矩陣 y 中的第一列，以此類推

(%o4) [0, 0, 0]

(%i5) inprod(y[1],y[1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第一列[1, 1, 1]和矩陣 y 中第一列[1, 1, 1]自己的內積

(%o5) 3

(%i6) inprod(y[2],y[2]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第二列[$-\frac{2}{3},\frac{1}{3},\frac{1}{3}$]和矩陣 y 中第二列[$-\frac{2}{3},\frac{1}{3},\frac{1}{3}$]自己的內積

(%o6) $\frac{2}{3}$

(%i7) inprod(y[3],y[3]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第三列$[0,-\frac{1}{2},\frac{1}{2}]$和矩陣 y 中第三列$[0,-\frac{1}{2},\frac{1}{2}]$自己的內積

(%o7) $\frac{1}{2}$

(%i8) a[1]:inprod([1,0,1],1/sqrt(3)*[1,1,1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[1, 0, 1]和$\frac{1}{\sqrt{3}}$[1, 1, 1]的內積，名稱給定爲 a[1]

(%o8) $\frac{2}{\sqrt{3}}$

(%i9) a[2]:inprod([1,0,1],1/sqrt(6)*[-2,1,1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[1, 0, 1]和$\frac{1}{\sqrt{6}}$[-2, 1, 1]的內積，名稱給定爲 a[2]

(%o9) $-\frac{1}{\sqrt{6}}$

(%i10) a[3]:inprod([1,0,1],1/sqrt(2)*[0,-1,1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[1, 0, 2]和$\frac{1}{\sqrt{2}}$[0, -1, 1]的內積，名稱給定爲 a[3]

(%o10) $\frac{1}{\sqrt{2}}$

(c) $V=P_2(R)$，其中內積<f(x), g(x)>=$\int_0^1 f(t)g(t)\,dt$，S={1, x, x^2}，及 f(x)=1+x

(d) V=span(S)，其中 S={(1, i, 0), (1-i, 2, 4i)}，及 x=(3+i, 4i, -4)

(e) $V=R^4$，S={(2, -1, -2, 4), (-2, 1, -5, 5), (-1, 3, 7, 11)}，且 x=(-11, 8, -4, 18)

(%i1) load(eigen)$ Maxima 本身是不會是不會算 gramschmidt 和 innerproduct 的，我們可以使用適當的模組來做這件事，所謂模組就是一小段程式，通常是增加一些指令供你使用，爲何 Maxima 不一開始就把這些模組都加進來，那是因爲

如此一來太佔用記憶體，於是我們要算一向量的 innerproduct 和 gramschmidt，要使用 eigen 這個模組 //讀取 eigen 這個 package，這個模組提供了 inprod 指令去算 innerproduct 及 gramschmidt 指令去算 gramschmidt 正交化

(%i2) x:matrix([2,-1,-2,4],[-2,1,-5,5],[-1,3,7,11]); //定義一矩陣，矩陣名稱叫做 x，第一列元素有 2, -1, -2, 4，第二列元素有-2, 1, -5, 5，第三列元素有-1, 3, 7, 11

$$(\%o2)\quad \begin{bmatrix} 2 & -1 & -2 & 4 \\ -2 & 1 & -5 & 5 \\ -1 & 3 & 7 & 11 \end{bmatrix}$$

(%i3) y:gramschmidt(x); Gram-Schmidt 單範正交化指令：gramschmidt(矩陣)，取出 x 矩陣中的列向量去做 Gram-Schmidt 步驟來計算正交向量在將這些向量正規化以得一單範正交基底 //對矩陣 x 作 Gram-Schmidt 正交化得出單範正交集名稱叫做 y

$$(\%o3)\quad [[2,-1,-2,4],[-2^2,2,-3,1],[-3,2^2,3^2,7]]$$

(%i4) map(innerproduct,[y[1],y[2],y[3]],[y[2],y[3],y[1]]); //驗證上列結果是否正確，於是驗證矩陣 y 中的列向量內積爲 0，表各自垂直，y[1]代表矩陣 y 中的第一列，以此類推

(%o4) [0, 0, 0]

(%i5) inprod(y[1],y[1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第一列[2, -1, -2, 4]和矩陣 y 中第一列[2, -1, -2, 4]自己的內積

(%o5) 25

(%i6) inprod(y[2],y[2]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第二列[-4, 2, -3, 1]和矩陣 y 中第二列[-4, 2, -3, 1]自己的內積

(%o6) 30

(%i7) inprod(y[3],y[3]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第三列[-3, 4, 9, 7]和矩陣 y 中第三列[-3, 4, 9, 7]自己的內積

(%o7) 155

(%i8) a[1]:inprod([-11,8,-4,18],1/5*[2,-1,-2,4]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-11, 8, -4, 18]和$\frac{1}{5}$[2, -1, -2, 4]的內積，名稱給定爲 a[1]

(%o8) 10

(%i9) a[2]:inprod([-11,8,-4,18],1/sqrt(30)*[-4,2,-3,1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-11, 8, -4, 18]和$\frac{1}{\sqrt{30}}$[-4, 2, -3, 1]的內積，名稱給定爲 a[2]

(%o9) $\frac{90}{\sqrt{30}}$

(%i10) a[3]:inprod([-11,8,-4,18],1/sqrt(155)*[-3,4,9,7]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-11, 8, -4, 18]和$\frac{1}{\sqrt{155}}$[-3, 4, 9, 7]的內積，名稱給定爲 a[3]

(%o10) $\frac{155}{\sqrt{155}}$

(f) V=R^4 ，S={(1, -2, -1, 3), (3, 6, 3, -1), (1, 4, 2, 8)}，且 x=(-1, 2, 1, 1)

(%i1) load(eigen)$ Maxima 本身是不會是不會算 gramschmidt 和 innerproduct 的，我們可以使用適當的模組來做這件事，所謂模組就是一小段程式，通常是增加一些指令供你使用，爲何 Maxima 不一開始就把這些模組都加進來，那是因爲如此一來太佔用記憶體，於是我們要算一向量的 innerproduct 和 gramschmidt，要使用 eigen 這個模組 //讀取 eigen 這個 package，這個模組提供了 inprod 指令去算 innerproduct 及 gramschmidt 指令去算 gramschmidt 正交化

(%i2) x:matrix([1,-2,-1,3],[3,6,3,-1],[1,4,2,8]); //定義一矩陣，矩陣名稱叫做 x，第一列元素有 1, -2, -1, 3，第二列元素有 3, 6, 3, -1，第三列元素有 1, 4, 2, 8

$$(\%o2)\quad \begin{bmatrix} 1 & -2 & -1 & 3 \\ 3 & 6 & 3 & -1 \\ 1 & 4 & 2 & 8 \end{bmatrix}$$

(%i3) y:gramschmidt(x); Gram-Schmidt 單範正交化指令：gramschmidt(矩陣)，取出 x 矩陣中的列向量去做 Gram-Schmidt 步驟來計算正交向量在將這些向量正規化以得一單範正交基底 //對矩陣 x 作 Gram-Schmidt 正交化得出單範正交集名稱叫做 y

$$(\%o3)\quad [[1,-2,-1,3],[2^2,2^2,2,2],[-2^2,2,1,3]]$$

(%i4) map(innerproduct,[y[1],y[2],y[3]],[y[2],y[3],y[1]]); //驗證上列結果是否正確，於是驗證矩陣 y 中的列向量內積為 0，表各自垂直，y[1]代表矩陣 y 中的第一列，以此類推

(%o4) [0, 0, 0]

(%i5) inprod(y[1],y[1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第一列[1, -2, -1, 3]和矩陣 y 中第一列[1, -2, -1, 3]自己的內積

(%o5) 15

(%i6) inprod(y[2],y[2]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第二列[4, 4, 2, 2]和矩陣 y 中第二列[4, 4, 2, 2]自己的內積

(%o6) 40

(%i7) inprod(y[3],y[3]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第三列[-4, 2, 1, 3]和矩陣 y 中第三列[-4, 2, 1, 3]自己的內積

(%o7) 30

(%i8) a[1]:inprod([-1,2,1,1],1/sqrt(15)*[1,-2,-1,3]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-1, 2, 1, 1]和 $\frac{1}{\sqrt{15}}$[1, -2, -1, 3]的內積，名

稱給定爲 a[1]

(%o8) $-\frac{3}{\sqrt{15}}$

(%i9) a[2]:inprod([-1,2,1,1],1/sqrt(40)*[4,4,2,2]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-1, 2, 1, 1]和$\frac{1}{\sqrt{40}}$[4, 4, 2, 2]的內積，名稱給定爲 a[2]

(%o9) $\frac{4}{\sqrt{10}}$

(%i10) a[3]:inprod([-1,2,1,1],1/sqrt(30)*[-4,2,1,3]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-1, 2, 1, 1]和$\frac{1}{\sqrt{30}}$[-4, 2, 1, 3]的內積，名稱給定爲 a[3]

(%o10) $\frac{12}{\sqrt{30}}$

(g) V=$M_{2\times 2}$(R)，S={$\begin{pmatrix} 3 & 5 \\ -1 & 1 \end{pmatrix}$, $\begin{pmatrix} -1 & 9 \\ 5 & -1 \end{pmatrix}$, $\begin{pmatrix} 7 & -17 \\ 2 & -6 \end{pmatrix}$}，且 A=$\begin{pmatrix} -1 & 27 \\ -4 & 8 \end{pmatrix}$

(%i1) load(eigen)$ Maxima 本身是不會是不會算 gramschmidt 和 innerproduct 的，我們可以使用適當的模組來做這件事，所謂模組就是一小段程式，通常是增加一些指令供你使用，爲何 Maxima 不一開始就把這些模組都加進來，那是因爲如此一來太佔用記憶體，於是我們要算一向量的 innerproduct 和 gramschmidt，要使用 eigen 這個模組 //讀取 eigen 這個 package，這個模組提供了 inprod 指令去算 innerproduct 及 gramschmidt 指令去算 gramschmidt 正交化

(%i2) x:matrix([3,-1,5,1],[-1,5,9,-1],[7,2,-17,-6]); //定義一矩陣，矩陣名稱叫做 x，第一列元素有 3, -1, 5, 1，第二列元素有-1, 5, 9, -1，第三列元素有 7, 2, -17, -6

(%o2) $\begin{bmatrix} 3 & -1 & 5 & 1 \\ -1 & 5 & 9 & -1 \\ 7 & 2 & -17 & -6 \end{bmatrix}$

(%i3) y:gramschmidt(x); Gram-Schmidt 單範正交化指令：gramschmidt(矩陣)，取出 x 矩陣中的列向量去做 Gram-Schmidt 步驟來計算正交向量在將這些向量正規化以得一單範正交基底 //對矩陣 x 作 Gram-Schmidt 正交化得出單範正交集名稱叫做 y

(%o3) $[[3,-1,5,1],[-2^2,23,2^2,-2],[3^2,23,-3,-23]]$

(%i4) map(innerproduct,[y[1],y[2],y[3]],[y[2],y[3],y[1]]); //驗證上列結果是否正確，於是驗證矩陣 y 中的列向量內積爲 0，表各自垂直，y[1]代表矩陣 y 中的第一列，以此類推

(%o4) [0, 0, 0]

(%i5) inprod(y[1],y[1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第一列[3, -1, 5, 1]和矩陣 y 中第一列[3, -1, 5, 1]自己的內積

(%o5) 36

(%i6) inprod(y[2],y[2]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第二列[-4, 23, 4, -2]和矩陣 y 中第二列[-4, 23, 4, -2]自己的內積

(%o6) 72

(%i7) inprod(y[3],y[3]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第三列[9, 23, -3, -23]和矩陣 y 中第三列[9, 23, -3, -23]自己的內積

(%o7) 162

(%i8) a[1]:inprod([-1,-4,27,8],1/6*[3,-1,5,1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-1, -4, 27, 8]和$\frac{1}{6}$[3, -1, 5, 1]的內積，名稱給定爲 a[1]

(%o8) 24

(%i9) a[2]:inprod([-1,-4,27,8],1/sqrt(72)*[-4,23,4,-2]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-1, -4, 27, 8]和$\frac{1}{\sqrt{72}}$ [-4, 23, 4, -2]的內積，名稱給定爲 a[2]

(%o9) $\frac{2}{3\sqrt{2}}$

(%i10) a[3]:inprod([-1,-4,27,8],1/sqrt(162)*[9,23,-3,-23]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-1, -4, 27, 8]和$\frac{1}{\sqrt{162}}$ [9, 23, -3, -23]的內積，名稱給定爲 a[3]

(%o10) $-\frac{122}{3\sqrt{2}}$

(h) V=$M_{2\times 2}$(R)，S={$\begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix}$, $\begin{pmatrix} 11 & 4 \\ 2 & 5 \end{pmatrix}$, $\begin{pmatrix} 4 & -12 \\ 3 & -16 \end{pmatrix}$}，且 A=$\begin{pmatrix} 8 & 6 \\ 25 & -13 \end{pmatrix}$

(%i1) load(eigen)$ Maxima 本身是不會是不會算 gramschmidt 和 innerproduct 的，我們可以使用適當的模組來做這件事，所謂模組就是一小段程式，通常是增加一些指令供你使用，爲何 Maxima 不一開始就把這些模組都加進來，那是因爲如此一來太佔用記憶體，於是我們要算一向量的 innerproduct 和 gramschmidt，要使用 eigen 這個模組 //讀取 eigen 這個 package，這個模組提供了 inprod 指令去算 innerproduct 及 gramschmidt 指令去算 gramschmidt 正交化

(%i2) x:matrix([2,2,2,1],[11,2,4,5],[4,3,-12,-16]); //定義一矩陣，矩陣名稱叫做 x，第一列元素有 2, 2, 2, 1，第二列元素有 11, 2, 4, 5，第三列元素有 4, 3, -12, -16

(%o2) $$\begin{bmatrix} 2 & 2 & 2 & 1 \\ 11 & 2 & 4 & 5 \\ 4 & 3 & -12 & -16 \end{bmatrix}$$

(%i3) y:gramschmidt(x); Gram-Schmidt 單範正交化指令：gramschmidt(矩陣)，取出 x 矩陣中的列向量去做 Gram-Schmidt 步驟來計算正交向量在將這些向量正規化以得一單範正交基底 //對矩陣 x 作 Gram-Schmidt 正交化得出單範正交集名稱叫做 y

(%o3) $[[2,2,2,1],[5,-2^2,-2,2],[2^3,7,-2^3,-27]]$

(%i4) map(innerproduct,[y[1],y[2],y[3]],[y[2],y[3],y[1]]); //驗證上列結果是否正確，於是驗證矩陣 y 中的列向量內積爲 0，表各自垂直，y[1]代表矩陣 y 中的第一列，以此類推

(%o4) [0 , 0 , 0]

(%i5) inprod(y[1],y[1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第一列[2, 2, 2, 1]和矩陣 y 中第一列[2, 2, 2, 1]自己的內積

(%o5) 13

(%i6) inprod(y[2],y[2]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第二列[5, -4, -2, 2]和矩陣 y 中第二列[5, -4, -2, 2]自己的內積

(%o6) 49

(%i7) inprod(y[3],y[3]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第三列[8, 7, -8, -27]和矩陣 y 中第三列[8, 7, -8, -27]自己的內積

(%o7) 373

(%i8) a[1]:inprod([8,25,6,-13],1/sqrt(13)*[2,2,2,1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[8, 25, 6, -13]和$\frac{1}{\sqrt{13}}$[2, 2, 2, 1]的內積，名稱給定爲 a[1]

(%o8) $\frac{65}{\sqrt{13}}$

(%i9) a[2]:inprod([8,25,6,-13],1/7*[5,-4,-2,2]); 的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[8, 25, 6, -13]和$\frac{1}{7}$[5, -4, -2, 2]的內積，名稱給定爲 a[2]

(%o9) - 14

(%i10) a[3]:inprod([8,25,6,-13],sqrt(13)/65*[8,7,-8,-27]); 的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[8, 25, 6, -13]和$\sqrt{13}$[8, 7, -8, -27]的內積，名稱給定爲 a[3]

(%o10) $\dfrac{542}{5\sqrt{13}}$

(i) V=span(S)具內積$\langle f,g\rangle=\int_0^{\pi} f(t)g(t)dt$，S={sint, cost, 1, t}，且 h(t)=2t+1

(j) V=C^4，S={(1, i, 2-i, -1), (2+3i, 3i, 1-i, 2i), (-1+7i, 6+10i, 11-4i, 3+4i)}，且 x={(-2+7i, 6+9i, 9-3i, 4+4i)}

(%i1) load(eigen)$ Maxima 本身是不會是不會算 gramschmidt 和 innerproduct 的，我們可以使用適當的模組來做這件事，所謂模組就是一小段程式，通常是增加一些指令供你使用，爲何 Maxima 不一開始就把這些模組都加進來，那是因爲如此一來太佔用記憶體，於是我們要算一向量的 innerproduct 和 gramschmidt，要使用 eigen 這個模組 //讀取 eigen 這個 package，這個模組提供了 inprod 指令去算 innerproduct 及 gramschmidt 指令去算 gramschmidt 正交化

(%i2)
x:matrix([1,%i,2-%i,-1],[2+3*%i,3*%i,1-%i,2*%i],[-1+7*%i,6+10*%i,11-4*%i,3+4*%i]); //定義一矩陣，矩陣名稱叫做 x，第一列元素有 1, i, 2-i, -1 ，第二列元素有 3i+2, 3i, 1-i, 2i，第三列元素有 7i-1, 10i+6, 11-4i, 4i+3

(%o2) $\begin{bmatrix} 1 & \%i & 2-\%i & -1 \\ 3\,\%i+2 & 3\,\%i & 1-\%i & 2\,\%i \\ 7\,\%i-1 & 10\,\%i+6 & 11-4\,\%i & 4\,\%i+3 \end{bmatrix}$

(%i3) y:gramschmidt(x); Gram-Schmidt 單範正交化指令：gramschmidt(矩陣)，取出 x 矩陣中的列向量去做 Gram-Schmidt 步驟來計算正交向量在將這些向量正規化以得一單範正交基底 //對矩陣 x 作 Gram-Schmidt 正交化得出單範正交集名

稱叫做 y

(%o3) [[1 , %i , 2 − %i , − 1] , [3 %i + 1 , 2 %i , − 1 , 2 %i + 1] , [%i − 7 , 2(%i + 3), 5 , 5]]

(%i4) map(innerproduct,[y[1],y[2],y[3]],[y[2],y[3],y[1]]); //驗證上列結果是否正確，於是驗證矩陣 y 中的列向量內積爲 0，表各自垂直，y[1]代表矩陣 y 中的第一列，以此類推

(%o4) [0 , 0 , 0]

(%i5) inprod(y[1],y[1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第一列[1, i, 2-i, -1]和矩陣 y 中第一列[1, i, 2-i, -1]自己的內積

(%o5) 8

(%i6) inprod(y[2],y[2]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第二列[3i+1, 2i, -1, 2i+1]和矩陣 y 中第二列[3i+1, 2i, -1, 2i+1]自己的內積

(%o6) 20

(%i7) inprod(y[3],y[3]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第三列[i-7, 2(i+3), 5, 5]和矩陣 y 中第三列[i-7, 2(i+3), 5, 5]自己的內積

(%o7) 140

(%i8) a[1]:inprod([-2+7*%i,6+9*%i,9-3*%i,4+4*%i],1/sqrt(8)*[1,%i,2-%i,-1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-2+7i, 6+9i, 9-3i, 4+4i]和$\frac{1}{\sqrt{8}}$[1, I, 2-i, -1]的內積，名稱給定爲 a[1]

(%o8) $6\sqrt{2}$

(%i9) a[2]:inprod([-2+7*%i,6+9*%i,9-3*%i,4+4*%i],1/sqrt(20)*[3*%i+1,2*%i,-1,2*%i+1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-2+7i, 6+9i, 9-3i, 4+4i]和$\frac{1}{\sqrt{20}}$[3i+1, 2i-1, -1, 2i+1]的內積，名稱給定爲 a[2]

(%o9) $4\sqrt{5}$

(%i10)
a[3]:inprod([-2+7*%i,6+9*%i,9-3*%i,4+4*%i],1/sqrt(140)/65*[%i-7,2*(%i+3),5,5]);

內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-2+7i, 6+9i, 9-3i, 4+4i]和$\frac{1}{65\sqrt{140}}$[i-7, 2(i+3), 5, 5]的內積，名稱給定爲 a[2]

(%o10) $\frac{2\sqrt{35}}{65}$

(a) V=C ^{4} ，S={(-4, 3-2i, i, 1-4i), (-1-5i, 5-4i, -3+5i, 7-2i), (-27-i, -7-6i, -15+25i, -7-6i)}，且 x=(-13-7i, -12+3i, -39-11i, -26+5i)

(%i1) load(eigen)$ Maxima 本身是不會是不會算 gramschmidt 和 innerproduct 的，我們可以使用適當的模組來做這件事，所謂模組就是一小段程式，通常是增加一些指令供你使用，爲何 Maxima 不一開始就把這些模組都加進來，那是因爲如此一來太佔用記憶體，於是我們要算一向量的 innerproduct 和 gramschmidt，要使用 eigen 這個模組 //讀取 eigen 這個 package，這個模組提供了 inprod 指令去算 innerproduct 及 gramschmidt 指令去算 gramschmidt 正交化

(%i2)
x:matrix([-4,3-2*%i,%i,1-4*%i],[-1-5*%i,5-4*%i,-3+5*%i,7-2*%i],[-27-%i,-7-6*%i,-15+25*%i,-7-6*%i]); //定義一矩陣，矩陣名稱叫做 x，第一列元素有-4, 3-2i, i, 1-4i，第二列元素有-1-5i, 5-4i, -3+5i, 7-2i，第三列元素有-27-i, -7-6i, -15+25i, -7-6i

(%o2) $\begin{bmatrix} -4 & 3-2\%i & \%i & 1-4\%i \\ -5\%i-1 & 5-4\%i & 5\%i-3 & 7-2\%i \\ -\%i-27 & -6\%i-7 & 25\%i-15 & -6\%i-7 \end{bmatrix}$

(%i3) y:gramschmidt(x); Gram-Schmidt 單範正交化指令：gramschmidt(矩陣)，取出 x 矩陣中的列向量去做 Gram-Schmidt 步驟來計算正交向量在將這些向量正規化以得一單範正交基底 //對矩陣 x 作 Gram-Schmidt 正交化得出單範正交集名

稱叫做 y

(%o3) [[- 4 , 3 - 2 %i , %i , 1 - 4 %i] , [-(%i - 3), - 5 %i , 2 (2 %i - 1), %i + 2] , [-(%i + 17), 8 %i - 9 , 2(8 %i - 9), 8 %i - 9]]

(%i4) map(innerproduct,[y[1],y[2],y[3]],[y[2],y[3],y[1]]); //驗證上列結果是否正確，於是驗證矩陣 y 中的列向量內積為 0，表各自垂直，y[1]代表矩陣 y 中的第一列，以此類推

(%o4) [0 , 0 , 0]

(%i5) inprod(y[1],y[1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第一列[-4, 3-2i, i, 1-4i]和矩陣 y 中第一列[-4, 3-2i, i, 1-4i]自己的內積

(%o5) 47

(%i6) inprod(y[2],y[2]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第二列[-(i-3), -5i, 2(2i-1), i+2]和矩陣 y 中第二列[-(i-3), -5i, 2(2i-1), i+2]自己的內積

(%o6) 60

(%i7) inprod(y[3],y[3]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //矩陣 y 中第三列[-(i+17), 8i-9, 2(8i-9), 8i-9]和矩陣 y 中第三列[-(i+17), 8i-9, 2(8i-9), 8i-9]自己的內積

(%o7) 1160

(%i8)
a[1]:inprod([-13-7*%i,-12+3*%i,-39-11*%i,-26+5*%i],1/sqrt(47)*[-4,3-2*%i,%i,1-4*%i]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字

//[-13-7i, -12+3i, -39-11i, -26+5i]和$\frac{1}{\sqrt{47}}$[-4, 3-2i, I, -4i]的內積，名稱給定為 a[1]

(%o8) $\sqrt{47}\,\%i - \sqrt{47}$

(%i9)
a[2]:inprod([-13-7*%i,-12+3*%i,-39-11*%i,-26+5*%i],1/sqrt(60)*[-(%i-3),-5*%i,2*(2*%i-1),%i+2]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的

$\boxed{\text{同義字}}$ //[-13-7i, -12+3i, -39-11i, -26+5i]和$\frac{1}{\sqrt{60}}$[-(i-3), -5i, 2(2i-1), i+2]的內積，名稱給定為 a[2]

(%o9) $-4\sqrt{15}\,\%i-2\sqrt{15}$

(%i10)
a[3]:inprod([-13-7*%i,-12+3*%i,-39-11*%i,-26+5*%i],1/sqrt(1160)*[-(%i+17),8*%i-9,2*(8*%i-9),8*%i-9]); $\boxed{\text{內積的指令：inprod([向量,向量]), inprod 是 innerproduct 的同義字}}$ //[-13-7i, -12+3i, -39-11i, -26+5i]和$\frac{1}{\sqrt{1160}}$[-(i+17), 8i-9, 2(8i-9), 8i-9]的內積，名稱給定為 a[3]

(%o10) $2\sqrt{290}-2\sqrt{290}\,\%i$

(b) V=M $_{2\times2}$ (C)，S={$\begin{pmatrix} 1-i & -2-3i \\ 2+2i & 4+i \end{pmatrix}$, $\begin{pmatrix} 8i & 4 \\ -3-3i & -4+4i \end{pmatrix}$, $\begin{pmatrix} -25-38i & -2-13i \\ 12-78i & -7+24i \end{pmatrix}$},且 A=$\begin{pmatrix} -2+8i & -13+i \\ 10-10i & 9-9i \end{pmatrix}$

(%i1) load(eigen)$ $\boxed{\text{Maxima 本身是不會是不會算 gramschmidt 和 innerproduct 的，我們可以使用適當的模組來做這件事，所謂模組就是一小段程式，通常是增加一些指令供你使用，爲何 Maxima 不一開始就把這些模組都加進來，那是因爲如此一來太佔用記憶體，於是我們要算一向量的 innerproduct 和 gramschmidt，要使用 eigen 這個模組}}$ //讀取 eigen 這個 package，這個模組提供了 inprod 指令去算 innerproduct 及 gramschmidt 指令去算 gramschmidt 正交化

(%i2)
x:matrix([1-%i,2+2*%i,-2-3*%i,4+%i],[8*%i,-3-3*%i,4,-4+4*%i],[-25-38*%i,12-78*%i,-2-13*%i,-7+24*%i]); //定義一矩陣，矩陣名稱叫做 x，第一列元素有 1-i, 2+2i, -2-3i, 4+i，第二列元素有 8i-3, -3-3i, 4, -4+4i，第三列元素有-25-38i, 12-78i, -2-13i, -7+24i

(%o2) $\begin{bmatrix} 1-\%i & 2\%i+2 & -3\%i-2 & \%i+4 \\ 8\%i & -3\%i-3 & 4 & 4\%i-4 \\ -38\%i-25 & 12-78\%i & -13\%i-2 & 24\%i-7 \end{bmatrix}$

(%i3) y:gramschmidt(x); $\boxed{\text{Gram-Schmidt 單範正交化指令：gramschmidt(矩陣)，取出 x 矩陣中的列向量去做 Gram-Schmidt 步驟來計算正交向量在將這些向量正規化以得一單範正交基底}}$ //對矩陣 x 作 Gram-Schmidt 正交化得出單範正交集名稱叫做 y

(%o3) [[1 − %i , 2 %i + 2 , − 3 %i − 2 , %i + 4] , [6 %i , −(3 %i − 1), −(%i + 1), %i + 1] , [−(43 %i + 2), − 68 %i , −(21 %i − 1), 34 %i]]

(%i4) map(innerproduct,[y[1],y[2],y[3]],[y[2],y[3],y[1]]); //驗證上列結果是否正確，於是驗證矩陣 y 中的列向量內積爲 0，表各自垂直，y[1]代表矩陣 y 中的第一列，以此類推

(%o4) [0 , 0 , 0]

(%i5) inprod(y[1],y[1]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //矩陣 y 中第一列[1-i, 2+2i, -2-3i, 4+i]和矩陣 y 中第一列[1-i, 2+2i, -2-3i, 4+i]自己的內積

(%o5) 40

(%i6) inprod(y[2],y[2]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //矩陣 y 中第二列[6i, -(3i-1), -(i+1), i+1]和矩陣 y 中第二列[6i, -(3i-1), -(i+1), i+1]自己的內積

(%o6) 50

(%i7) inprod(y[3],y[3]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //矩陣 y 中第三列[-(43i+2), -68i, -(21i-1), 34i]和矩陣 y 中第三列[-(43i+2), -68i, -(21i-1), 34i]自己的內積

(%o7) 8075

(%i8)
a[1]:inprod([-2+8*%i,10-10*%i,-13+%i,9-9*%i],1/sqrt(40)*[1-%i,2*%i+2,-3*%i-2,%i+4]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$

//[-2+8i, 10-10i, -13+i, 9-9i]和 $\frac{1}{\sqrt{40}}$ [1-i, i+2, -3i-2, i+4]的內積，名稱給定爲 a[1]

(%o8) $6\sqrt{10}\,\%i + 2\sqrt{10}$

(%i9)
a[2]:inprod([-2+8*%i,10-10*%i,-13+%i,9-9*%i],1/sqrt(50)*[6*%i,-(3*%i-1),-(%i+1),%i+1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字
//[-2+8i, 10-10i, -13+i, 9-9i]和 $\frac{1}{\sqrt{50}}$ [6i, -(3i-1), -(i+1), i+1]的內積，名稱給定爲 a[2]

(%o9) $10\sqrt{2}$

(%i10)
a[3]:inprod([-2+8*%i,10-10*%i,-13+%i,9-9*%i],1/sqrt(8075)*[-(43*%i+2),-68*%i,-(21*%i-1),34*%i]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-2+8i, 10-10i, -13+i, 9-9i]和 $\frac{1}{\sqrt{8075}}$ [-(43i+2), -68i, -(21i-1), 34i]的內積，名稱給定爲 a[3]

(%o10) 0

(c) V=M $_{2\times 2}$ (C)，S={ $\begin{pmatrix} -1+i & -i \\ 2-i & 1-+3i \end{pmatrix}$, $\begin{pmatrix} -1-7i & -9-8i \\ 1+10i & -6-2i \end{pmatrix}$, $\begin{pmatrix} -11-132i & -34-31i \\ 7-126i & -71-5i \end{pmatrix}$ }，且 A={ $\begin{pmatrix} -7+5i & 3+18i \\ 9-6i & -3+7i \end{pmatrix}$ }

(%i135) load(eigen)$ Maxima 本身是不會是不會算 gramschmidt 和 innerproduct 的，我們可以使用適當的模組來做這件事，所謂模組就是一小段程式，通常是增加一些指令供你使用，爲何 Maxima 不一開始就把這些模組都加進來，那是因爲如此一來太佔用記憶體，於是我們要算一向量的 innerproduct 和 gramschmidt，要使用 eigen 這個模組 //讀取 eigen 這個 package，這個模組提供了 inprod 指令去算 innerproduct 及 gramschmidt 指令去算 gramschmidt 正交化
x:matrix([-1+%i,2-%i,-%i,1+3*%i],[-1-7*%i,1+10*%i,-9-8*%i,-6-2*%i],[-11-132*%i,7-126*%i,-34-31*%i,-71-5*%i]); //定義一矩陣，矩陣名稱叫做 x，第一列元素有-1+i, 2-i, -i, 1+3i，第二列元素有-1-7i, 1+10i, -9-8i, 6-2i，第三列元素有-11-132i, 7-126i, -34-31i, -71-5i

(%o136) $\begin{bmatrix} \%i-1 & 2-\%i & -\%i & 3\%i+1 \\ -7\%i-1 & 10\%i+1 & -8\%i-9 & -2\%i-6 \\ -132\%i-11 & 7-126\%i & -31\%i-34 & -5\%i-71 \end{bmatrix}$

(%i137) y:gramschmidt(x); Gram-Schmidt 單範正交化指令：gramschmidt(矩陣)，取出 x 矩陣中的列向量去做 Gram-Schmidt 步驟來計算正交向量在將這些向量正

$\boxed{\text{規化以得一單範正交基底}}$ //對矩陣 x 作 Gram-Schmidt 正交化得出單範正交集名稱叫做 y

(%o137) [[%i - 1 , 2 - %i , - %i , 3 %i + 1] , [- 4 %i , 5 %i + 1 , - (9 %i + 11), -(%i - 1)] , [-(118 %i + 5), - 145 %i , -(26 %i + 7), - 229]]

(%i138) map(innerproduct,[y[1],y[2],y[3]],[y[2],y[3],y[1]]); //驗證上列結果是否正確，於是驗證矩陣 y 中的列向量內積爲 0，表各自垂直，y[1]代表矩陣 y 中的第一列，以此類推

(%o138) [0 , 0 , 0]

(%i139) inprod(y[1],y[1]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //矩陣 y 中第一列[-1+i, 2-i, -i, 1+3i]和矩陣 y 中第一列[-1+i, 2-i, -i, 1+3i]自己的內積

(%i140) inprod(y[2],y[2]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //矩陣 y 中第二列[-4i, 5i+1, -(9i+11), -(i-1)]和矩陣 y 中第二列[-4i, 5i+1, -(9i+11), -(i-1)]自己的內積

(%o140) 246

(%i141) inprod(y[3],y[3]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //矩陣 y 中第三列[-(118i+5), -145i, -(26i+7), -229]和矩陣 y 中第三列[-(118i+5), -145i, -(26i+7), -229]自己的內積

(%o141) 39063

(%i142)
a[1]:inprod([-7+5*%i,9-6*%i,3+18*%i,-3+7*%i],1/sqrt(18)*[%i-1,2-%i,-%i,3*%i+1]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //[-7+5i, 9-6i, 3+18i, -3+7i]和 $\frac{1}{\sqrt{18}}$[i-1, 2-i, -i, 3i+1]的內積，名稱給定爲 a[1]

(%o142) $6\sqrt{2} - 3\sqrt{2}\,\%i$

(%i143)
a[2]:inprod([-7+5*%i,9-6*%i,3+18*%i,-3+7*%i],1/sqrt(246)*[-4*%i,5*%i+1,-(9*%i+11),-(%i-1)]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同}}$

義字 //[-7+5i, 9-6i, 3+18i, -3+7i]和$\frac{1}{\sqrt{246}}$[-4i, 5i+1, -(9i+11), -(i-1)]的內積，名稱給定為 a[2]

(%o143) $\sqrt{246}\%i-\sqrt{246}$

(%i144)
a[3]:inprod([-7+5*%i,9-6*%i,3+18*%i,-3+7*%i],1/sqrt(39063)*[-(118*%i+5),-145*%i,-(26*%i+7),-2*29]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[-7+5i, 9-6i, 3+18i, -3+7i]和$\frac{1}{\sqrt{39063}}$[-(118i+5), -145i, -(26i+7), -58]的內積，名稱給定為 a[3]

(%o144) 0

3.在 R^2，令 $\beta=\{(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}),(\frac{1}{\sqrt{2}},\frac{-1}{\sqrt{2}})\}$，求(3,4)相對於 β 的傅笠爾係數

(%i1) load(eigen)$ Maxima 本身是不會是不會算 gramschmidt 和 innerproduct 的，我們可以使用適當的模組來做這件事，所謂模組就是一小段程式，通常是增加一些指令供你使用，為何 Maxima 不一開始就把這些模組都加進來，那是因為如此一來太佔用記憶體，於是我們要算一向量的 innerproduct 和 gramschmidt，要使用 eigen 這個模組 //讀取 eigen 這個 package，這個模組提供了 inprod 指令去算 innerproduct 及 gramschmidt 指令去算 gramschmidt 正交化
inprod([3,4],[1/sqrt(2),1/sqrt(2)]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //向量(3, 4)跟向量($\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}$)作內積

(%o2) $\frac{7}{\sqrt{2}}$

(%i3) inprod([3,4],[1/sqrt(2),-1/sqrt(2)]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //向量(3, 4)跟向量($\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}$)作內積

(%o3) $-\frac{1}{\sqrt{2}}$

4.令 S={(1, 0, i), (1, 2, 1)}$\subset C^3$，試求 $S^{\perp}$

19.在下列各小題，求已知內積空間 V 的子空間 W 上給定向量的正交投影

(a) V=R^2，u=(2, 6), W={(x, y)：y=4x}

(b) V=R^3，u=(2, 1, 3), W={(x, y, z)：x+3y-2z=0}

(c) V=P(R)，其中內積<f, g>=$\int_0^1 f(t)g(t)dt$，h(x)=4+3x-2x^2，W=P_1(R)

20.在 19，求已知向量至子空間 W 的距離

6.3 伴隨線性算子

2.對下列各內積空間 V(佈於 F)及線性變換 g：V→F，求向量 y 使 g(x)=<x, y>, ∀x∈V

(a) V=R^3，g(a_1, a_2, a_3)=a_1-2a_2+4a_3

(%i5) g(a1,a2,a3):=a1-2*a2+4*a3; 定義函數的指令：函數名稱(變數 1，…，變數 n):=函數式 //函數名稱爲 g，變數 1 爲 a1、變數 2 爲 a2、變數 3 爲 a3，函數 g=a_1-2a_2+4a_3

(%o5) $g(a1, a2, a3) := a1 - 2\, a2 + 4\, a3$

(%i6) g(1,0,0); //將 a1=1，a2=0，a3=0 帶入函數 g 求出值

(%o6) 1

(%i7) g(0,1,0); //將 a1=0，a2=1，a3=0 帶入函數 g 求出值

(%o7) -2

(%i8) g(0,0,1); //將 a1=0，a2=0，a3=1 帶入函數 g 求出值

(%o8) 4

(b) $V=C^2$，$g(z_1, z_2)=z_1-2z_2$

(%i1) g(z1,z2):=z1-2*z2; 定義函數的指令：函數名稱(變數 1，…，變數 n):=函數式 //函數名稱爲 g，變數 1 爲 z1、變數 2 爲 z2，函數 $g=z_1-2z_2$

(%o1) $g(z1, z2):=z1-2\,z2$

(%i2) g(1,0); //將 z1=1，z2=0，帶入函數 g 求出值

(%o2) 1

(%i3) g(0,1); //將 z1=0，z2=1，帶入函數 g 求出值

(%o3) -2

(c) $V=P_2(R)$，其中$<f, g>=\int_0^1 f(t)h(t)dt$，$g(f)=f(0)+f'(1)$

3.對下列各內積空間 V 及 V 上的線性算子 T，求 V 上已知向量的 T^* 值

(a) $V=R^2$, $T(a, b)=(2a+b, a-3b)$, $x=(3,5)$

(b) $V=C^2$，$T(z_1, z_2)=(2z_1+iz_2, (1-i)z_1)$, $x=(3-i, 1+2i)$

(c) $V=P_1(R)$，其中$<f, g>=\int_{-1}^1 f(t)g(t)dt$，$T(f)=f'+3f$，$f(t)=4-2t$

6.4 正規算子與自伴算子

2.對每一個在內積空間 V 上的線性算子，試決定 T 是否爲正規算子，自伴算子，或兩者均不是，若可能，找出一組由 T 之固有向量所組成的 V 之單範正交基底並列出所對應的特徵值

(a) $V=R^2$ 且 T 被定義爲 $T(a, b)=(2a-2b, -2a+5b)$

(b) $V=R^3$ 且 T 被定義爲 $T(a, b, c)=(-a+b, 5b, 4a-2b+5c)$

(c) $V=C^2$ 且 T 被定義爲 $T(a, b)=(2a+ib, a+2b)$

(d) V=P_2(R)且 T 被定義為 T(f)= f' ，其中<f, g>=$\int_0^1 f(t)g(t)dt$

(e) V=$M_{2\times 2}$(R)且 T 被定義為 T(A)=A^t

(f) V=$M_{2\times 2}$(R)且 T 被定義為 T$\begin{pmatrix} a & b \\ c & d \end{pmatrix}=\begin{pmatrix} c & d \\ a & b \end{pmatrix}$

6.5 么正算子與正交算子及其矩陣

2.對下列各矩陣 A，求出一正交或么正矩陣 P 及對角矩陣 D 使 P^*AP=D

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

(%i1) A:matrix([1,2],[2,1]); //定義一矩陣，矩陣名稱叫做 A，第一列元素有 1, 2，第二列元素有 2, 1

(%o1) $\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$

(%i2) eigenvectors(A); eigenvector 也會把特徵值列出來，所以是包含前面 eigenvalues 功能的指令。特徵向量指令：eigenvector(矩陣)，第一部分和 eigenvalues 輸出一樣，這些向量會生成相對應特徵值的向量空間 //第一部分和 eigenvalues 輸出一樣，對於特徵值 3 有一個對應的特徵向量[1,1]，而特徵值-1 對應到的特徵向量為[1, -1]

(%o2) [[[3,-1],[1,1]],[1,1],[1,-1]]

(%i3) P:1/sqrt(2)*matrix([1,1],[1,-1]); //定義一矩陣，矩陣名稱叫做 P，第一列的元素有$\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}$，第二列的元素有$\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}$

(%o3) $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$

(%i4) D=invert(P).A.P;　 //求 P 的反矩陣乘上矩陣 A 乘上矩陣 P，所求出矩陣爲一對角矩陣，名稱叫做 D

(%o4) $D = \begin{bmatrix} 3 & 0 \\ 0 & -1 \end{bmatrix}$

(b) $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

(%i1) A:matrix([0,-1],[1,0]);　 //定義一矩陣，矩陣名稱叫做 A，第一列的元素有 0, -1，第二列的元素有 1, 0

(%o1) $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

(%i2) eigenvectors(A);　 eigenvector 也會把特徵值列出來，所以是包含前面 eigenvalues 功能的指令。特徵向量指令：eigenvector(矩陣)，第一部分和 eigenvalues 輸出一樣，這些向量會生成相對應特徵值的向量空間　 //第一部分和 eigenvalues 輸出一樣，對於特徵值-i 有一個對應的特徵向量[1, i]，而特徵值 i 對應到的特徵向量爲[1, -i]

(%o2) [[[-%i, %i], [1, 1]], [1, %i], [1, -%i]]

(%i3) P:1/sqrt(2)*matrix([1,1],[%i,-%i]);　 /定義一矩陣，矩陣名稱叫做 P，第一列的元素有 $\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}$ ，第二列的元素有 $\frac{i}{\sqrt{2}}, -\frac{i}{\sqrt{2}}$

(%o3) $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{\%i}{\sqrt{2}} & -\frac{\%i}{\sqrt{2}} \end{bmatrix}$

(%i4) D=invert(P).A.P;　 //求 P 的反矩陣乘上矩陣 A 乘上矩陣 P，所求出矩陣爲一對角矩陣，名稱叫做 D

(%o4) $D = \begin{bmatrix} -\%i & 0 \\ 0 & \%i \end{bmatrix}$

(c) $\begin{pmatrix} 2 & 3-3i \\ 3+3i & 5 \end{pmatrix}$

(d) $\begin{pmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix}$

(%i1) A:matrix([0,2,2],[2,0,2],[2,2,0]); //定義一矩陣，矩陣名稱叫做 A，第一列元素有 0, 2, 2，第二列元素有 2, 0, 2，第三列元素有 2, 2, 0

(%o1) $\begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix}$

(%i2) eigenvectors(A); $\boxed{\text{eigenvector 也會把特徵值列出來，所以是包含前面 eigenvalues 功能的指令。特徵向量指令：eigenvector(矩陣)，第一部分和 eigenvalues 輸出一樣，這些向量會生成相對應特徵值的向量空間}}$ //第一部分和 eigenvalues 輸出一樣，對於特徵值 4 有一個對應的特徵向量[1, 1, 1]，而特徵值-2 對應到的特徵向量爲[1, 0, -1]、[0, 1, -1]

(%o2) [[[4,-2],[1,2]],[1,1,1],[1,0,-1],[0,1,-1]]

(%i3) inprod([1,1,1],[1,1,1]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //[1, 1, 1]和[1, 1, 1]的內積

(%o3) 3

(%i4) inprod([1,0,-1],[1,0,-1]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //[1, 0, -1]和[1, 0, -1]的內積

(%o4) 2

(%i5) inprod([0,1,-1],[0,1,-1]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //[0, 1, -1]和[0, 1, -1]的內積

(%o5) 2

(%i6)

P:matrix([1/sqrt(3),1/sqrt(2),0],[1/sqrt(3),0,1/sqrt(2)],[1/sqrt(3),-1/sqrt(2),-1/sqrt(2)]);

/定義一矩陣，矩陣名稱叫做 P，第一列的元素有 $\frac{1}{\sqrt{3}},\frac{1}{\sqrt{2}},0$ ，第二列的元素有 $\frac{1}{\sqrt{3}},0,\frac{1}{\sqrt{2}}$ ，第三列的元素有 $\frac{1}{\sqrt{3}},-\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}$

(%o6) $\begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$

(%i7) D=invert(P).A.P; //求 P 的反矩陣乘上矩陣 A 乘上矩陣 P，所求出矩陣爲一對角矩陣，名稱叫做 D

(%o7) $D = \begin{bmatrix} 4 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{bmatrix}$

(e) $\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix}$

(%i1) A:matrix([2,1,1],[1,2,1],[1,1,2]); //定義一矩陣，矩陣名稱叫做 A，第一列元素有 2, 1, 1，第二列元素有 1, 2, 1，第三列元素有 1, 1, 2

(%o1) $\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$

(%i2) eigenvectors(A); eigenvector 也會把特徵值列出來，所以是包含前面 eigenvalues 功能的指令。特徵向量指令：eigenvector(矩陣)，第一部分和 eigenvalues 輸出一樣，這些向量會生成相對應特徵值的向量空間 //第一部分和 eigenvalues 輸出一樣，對於特徵值 4 有一個對應的特徵向量[1, 1, 1]，而特徵值 1 對應到的

特徵向量爲[1, 0, -1]、[0, 1, -1]

(%o2) [[[4,1],[1,2]],[1,1,1],[1,0,-1],[0,1,-1]]

(%i3) inprod([1,1,1],[1,1,1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[1, 1, 1]和[1, 1, 1]的內積

(%o3) 3

(%i4) inprod([1,0,-1],[1,0,-1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[1, 0, -1]和[1, 0, -1]的內積

(%o4) 2

(%i5) inprod([0,1,-1],[0,1,-1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //[0, 1, -1]和[0, 1, -1]的內積

(%o5) 2

(%i6) P:matrix([1/sqrt(3),1/sqrt(2),0],[1/sqrt(3),0,1/sqrt(2)],[1/sqrt(3),-1/sqrt(2),-1/sqrt(2)]); /定義一矩陣，矩陣名稱叫做 P，第一列的元素有 $\frac{1}{\sqrt{3}},\frac{1}{\sqrt{2}},0$，第二列的元素有 $\frac{1}{\sqrt{3}},0,\frac{1}{\sqrt{2}}$，第三列的元素有 $\frac{1}{\sqrt{3}},-\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}$

(%o6) $\begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$

(%i7) D=invert(P).A.P; //求 P 的反矩陣乘上矩陣 A 乘上矩陣 P，所求出矩陣爲一對角矩陣，名稱叫做 D

(%o7) $D=\begin{bmatrix} 4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

5.下列何對矩陣是么正等價

(a) $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ 及 $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

(%i1) A:matrix([1,0],[0,1]); //定義一矩陣，矩陣名稱叫作 A，第一列元素有 1, 0，第二列元素有 0, 1

(%o1) $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

(%i2) f:charpoly(A,t); $\boxed{\text{特徵多項式指令：charpoly(矩陣，變數)}}$ //求 A 的特徵多項式，特徵多項式名稱定爲 f，f 內的變數爲 t

(%o2) $(1-t)^2$

(%i3) factor(f); $\boxed{\text{多項式分解指令：factor(多項式名稱)}}$ //將特徵多項式分解，可以清楚看到 A 有幾個特徵值和各特徵值的代數重數

(%o3) $(t-1)^2$

(%i4) B:matrix([0,1],[1,0]); //定義一矩陣，矩陣名稱叫作 B，第一列元素有 0, 1，第二列元素有 1, 0

(%o4) $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

(%i5) g:charpoly(B,t); $\boxed{\text{特徵多項式指令：charpoly(矩陣，變數)}}$ //求 B 的特徵多項式，特徵多項式名稱定爲 g，g 內的變數爲 t

(%o5) t^2-1

(%i6) factor(g); $\boxed{\text{多項式分解指令：factor(多項式名稱)}}$ //將特徵多項式分解，可以清楚看到 B 有幾個特徵值和各特徵值的代數重數

(%o6) $(t-1)(t+1)$

(b) $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ 及 $\begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}$

(%i1) A:matrix([0,1],[1,0]); //定義一矩陣，矩陣名稱叫作 A，第一列元素有 0, 1，第二列元素有 1, 0

(%o1) $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

(%i2) f:charpoly(A,t); $\boxed{\text{特徵多項式指令：charpoly(矩陣，變數)}}$ //求 A 的特徵多項式，特徵多項式名稱定爲 f，f 內的變數爲 t

(%o2) t^2-1

(%i3) factor(f); $\boxed{\text{多項式分解指令：factor(多項式名稱)}}$ //將特徵多項式分解，可以清楚看到 A 有幾個特徵值和各特徵值的代數重數

(%o3) $(t-1)(t+1)$

(%i4) B:matrix([0,1/2],[1/2,0]); //定義一矩陣，矩陣名稱叫作 B，第一列元素有 $0,\frac{1}{2}$，第二列元素有 $\frac{1}{2}, 0$

(%o4) $\begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix}$

(%i5) g:charpoly(B,t); $\boxed{\text{特徵多項式指令：charpoly(矩陣，變數)}}$ //求 B 的特徵多項式，特徵多項式名稱定爲 g，g 內的變數爲 t

(%o5) $t^2-\frac{1}{4}$

(%i6) factor(g); $\boxed{\text{多項式分解指令：factor(多項式名稱)}}$ //將特徵多項式分解，可以清楚看到 B 有幾個特徵值和各特徵值的代數重數

(%o6) $\frac{(2t-1)(2t+1)}{4}$

(c) $\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ 及 $\begin{pmatrix} 2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

(%i1) A:matrix([0,1,0],[-1,0,0],[0,0,1]); //定義一矩陣，矩陣名稱叫作 A，第一列元素有 0, 1, 0，第二列元素有-1, 0, 0，第三列元素有 0, 0, 1

(%o1) $\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

(%i2) f:charpoly(A,t); $\boxed{\text{特徵多項式指令：charpoly(矩陣，變數)}}$ //求 A 的特徵多項式，特徵多項式名稱定為 f，f 內的變數為 t

(%o2) $(1-t)t^2-t+1$

(%i3) factor(f); $\boxed{\text{多項式分解指令：factor(多項式名稱)}}$ //將特徵多項式分解，可以清楚看到 A 有幾個特徵值和各特徵值的代數重數

(%o3) $-(t-1)\left(t^2+1\right)$

(%i4) B:matrix([2,0,0],[0,-1,0],[0,0,0]); //定義一矩陣，矩陣名稱叫作 B，第一列元素有 2, 0, 0，第二列元素有 0, -1, 0，第三列元素有 0, 0, 0

(%o4) $\begin{bmatrix} 2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

(%i5) g:charpoly(B,t); $\boxed{\text{特徵多項式指令：charpoly(矩陣，變數)}}$ //求 B 的特徵多項式，特徵多項式名稱定為 g，g 內的變數為 t

(%o5) $-(-t-1)(2-t)t$

(%i6) factor(g); $\boxed{\text{多項式分解指令：factor(多項式名稱)}}$ //將特徵多項式分解，可以清楚看到 B 有幾個特徵值和各特徵值的代數重數

(%o6) $-(t-2)t(t+1)$

(d) $\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ 及 $\begin{pmatrix} 1 & 0 & 0 \\ 0 & i & 0 \\ 0 & 0 & -i \end{pmatrix}$

(%i1) A:matrix([0,1,0],[-1,0,0],[0,0,1]); //定義一矩陣，矩陣名稱叫作 A，第一列元素有 0, 1, 0，第二列元素有-1, 0, 0，第三列元素有 0, 0, 1

(%o1) $\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

(%i2) f:charpoly(A,t); $\boxed{\text{特徵多項式指令：charpoly(矩陣，變數)}}$ //求 A 的特徵多項式，特徵多項式名稱定為 f，f 內的變數為 t

(%o2) $(1-t)t^2-t+1$

(%i3) factor(f); $\boxed{\text{多項式分解指令：factor(多項式名稱)}}$ //將特徵多項式分解，可以清楚看到 A 有幾個特徵值和各特徵值的代數重數

(%o3) $-(t-1)\left(t^2+1\right)$

(%i4) B:matrix([1,0,0],[0,%i,0],[0,0,-%i]); //定義一矩陣，矩陣名稱叫作 B，第一列元素有 1, 0, 0，第二列元素有 0, i, 0，第三列元素有 0, 0, -i

(%o4) $\begin{bmatrix} 1 & 0 & 0 \\ 0 & \%i & 0 \\ 0 & 0 & -\%i \end{bmatrix}$

(%i5) g:charpoly(B,t); $\boxed{\text{特徵多項式指令：charpoly(矩陣，變數)}}$ //求 B 的特徵多項式，特徵多項式名稱定爲 g，g 內的變數爲 t

(%o5) $(1-t)(-t-\%i)(\%i-t)$

(%i6) factor(g); $\boxed{\text{多項式分解指令：factor(多項式名稱)}}$ //將特徵多項式分解，可以清楚看到 B 有幾個特徵值和各特徵值的代數重數

(%o6) $-(t-1)(t-\%i)(t+\%i)$

(e) $\begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{pmatrix}$ 及 $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$

(%i1) A:matrix([1,1,0],[0,2,2],[0,0,3]); //定義一矩陣，矩陣名稱叫作 A，第一列元素有 1, 1, 0，第二列元素有 0, 2, 2，第三列元素有 0, 0, 3

(%o1) $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{bmatrix}$

(%i2) f:charpoly(A,t); $\boxed{\text{特徵多項式指令：charpoly(矩陣，變數)}}$ //求 A 的特徵多項式，特徵多項式名稱定爲 f，f 內的變數爲 t

(%o2) $(1-t)(2-t)(3-t)$

(%i3) factor(f); $\boxed{\text{多項式分解指令：factor(多項式名稱)}}$ //將特徵多項式分解，可以清楚看到 A 有幾個特徵值和各特徵值的代數重數

(%o3) $-(t-3)(t-2)(t-1)$

(%i4) B:matrix([1,0,0],[0,2,0],[0,0,3]); //定義一矩陣，矩陣名稱叫作 B，第一列元素有 1, 0, 0，第二列元素有 0, 2, 0，第三列元素有 0, 0, 3

(%o4) $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$

(%i5) g:charpoly(B,t); 特徵多項式指令：charpoly(矩陣，變數) //求 B 的特徵多項式，特徵多項式名稱定爲 g，g 內的變數爲 t

(%o5) $(1-t)(2-t)(3-t)$

(%i6) factor(g); 多項式分解指令：factor(多項式名稱) //將特徵多項式分解，可以清楚看到 B 有幾個特徵值和各特徵值的代數重數

(%o6) $-(t-3)(t-2)(t-1)$

27.求新座標 x' ，y' 使下列二形式可表成 $\lambda_1(x')^2+\lambda_2(y')^2$

(a) $x^2+4xy+y^2$

(%i1) A:matrix([1,2],[2,1]); //定義一矩陣，矩陣名稱叫作 A，第一列元素有 1, 2，第二列元素有 2, 1

(%o1) $\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$

(%i2) f:matrix([x,y]).A.matrix([x],[y]); //定義一多項式，爲矩陣 $\begin{pmatrix} x & y \end{pmatrix}$ 乘上矩陣 A 乘上矩陣 $\begin{pmatrix} x \\ y \end{pmatrix}$，多項式名稱爲 f

(%o2) $x(2y+x)+y(y+2x)$

(%i3) expand(f); 展開多項式指令：expand(多項式名稱) //將多項式 f 展開得到 $x^2+4xy+y^2$

(%o3) $y^2+4xy+x^2$

(%i4) eigenvectors(A); eigenvector 也會把特徵值列出來，所以是包含前面 eigenvalues 功能的指令。特徵向量指令：eigenvector(矩陣)，第一部分和 eigenvalues 輸出一樣，這些向量會生成相對應特徵值的向量空間 //第一部分和 eigenvalues 輸出一樣，對於特徵值 3 有一個對應的特徵向量[1, 1]，而特徵值-1 對應到的特徵向量爲[1, -1]

(%o4) [[[3,-1],[1,1]],[1,1],[1,-1]]

(%i5) inprod([1,1],[1,1]); 內積的指令：inprod([向量，向量])，inprod 是

$\boxed{\text{innerproduct 的同義字}}$ //向量(1, 1)跟向量(1, 1)作內積

(%o5) 2

(%i6) inprod([1,-1],[1,-1]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是}}$ $\boxed{\text{innerproduct 的同義字}}$ //向量(1, -1)跟向量(1, -1)作內積

(%o6) 2

(%i7) P:matrix([1/sqrt(2),1/sqrt(2)],[1/sqrt(2),-1/sqrt(2)]); //定義一矩陣，矩陣名稱叫做 P，第一列的元素有 $\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}$ ，第二列的元素有 $\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}$

(%o7) $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$

(%i8) D=invert(P).A.P; //求 P 的反矩陣乘上矩陣 A 乘上矩陣 P，所求出矩陣爲一對角矩陣，名稱叫做 D

(%o8) $D = \begin{bmatrix} 3 & 0 \\ 0 & -1 \end{bmatrix}$

(%i9)
matrix([x],[y])=matrix([1/sqrt(2),1/sqrt(2)],[1/sqrt(2),-1/sqrt(2)]).matrix([x],[y]);

//可以得到新座標與舊座標的關係式

(%o9) $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{y}{\sqrt{2}} + \frac{x}{\sqrt{2}} \\ \frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}} \end{bmatrix}$

(b) $2x^2 + 2xy + 2y^2$

(%i1) A:matrix([2,1],[1,2]); //定義一矩陣，矩陣名稱叫作 A，第一列元素有 2, 1，第二列元素有 1, 2

(%o1) $\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$

(%i2) f:matrix([x,y]).A.matrix([x],[y]);　//定義一多項式，爲矩陣$\begin{pmatrix} x & y \end{pmatrix}$乘上矩陣 A 乘上矩陣$\begin{pmatrix} x \\ y \end{pmatrix}$，多項式名稱爲 f

(%o2) $y(2y+x)+x(y+2x)$

(%i3) expand(f);　展開多項式指令：expand(多項式名稱)　//將多項式 f 展開得到 $2x^2+2xy+2y^2$

(%o3) $2y^2+2xy+2x^2$

(%i4) eigenvectors(A);　eigenvector 也會把特徵值列出來，所以是包含前面 eigenvalues 功能的指令。特徵向量指令：eigenvector(矩陣)，第一部分和 eigenvalues 輸出一樣，這些向量會生成相對應特徵值的向量空間　//第一部分和 eigenvalues 輸出一樣，對於特徵值 3 有一個對應的特徵向量[1, 1]，而特徵值 1 對應到的特徵向量爲[1, -1]

(%o4) [[[3,1],[1,1]],[1,1],[1,-1]]

(%i5) inprod([1,1],[1,1]);　內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字　//向量(1, 1)跟向量(1, 1)作內積

(%o5) 2

(%i6) inprod([1,-1],[1,-1]);　內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字　//向量(1, -1)跟向量(1, -1)作內積

(%o6) 2

(%i7) P:matrix([1/sqrt(2),1/sqrt(2)],[1/sqrt(2),-1/sqrt(2)]);　//定義一矩陣，矩陣名稱叫做 P，第一列的元素有$\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}$，第二列的元素有$\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}$

(%o7) $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$

(%i8) D=invert(P).A.P; //求 P 的反矩陣乘上矩陣 A 乘上矩陣 P，所求出矩陣爲一對角矩陣，名稱叫做 D

(%o8) $D = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$

(%i9)
matrix([x],[y])=matrix([1/sqrt(2),1/sqrt(2)],[1/sqrt(2),-1/sqrt(2)]).matrix([x],[y]);

//可以得到新座標與舊座標的關係式

(%o9) $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{y}{\sqrt{2}} + \frac{x}{\sqrt{2}} \\ \frac{x}{\sqrt{2}} - \frac{y}{\sqrt{2}} \end{bmatrix}$

(c) $x^2 - 12xy - 4y^2$

(%i1) A:matrix([1,-6],[-6,-4]); //定義一矩陣，矩陣名稱叫作 A，第一列元素有 1, -6，第二列元素有-6, -4

(%o1) $\begin{bmatrix} 1 & -6 \\ -6 & -4 \end{bmatrix}$

(%i2) f:matrix([x,y]).A.matrix([x],[y]); //定義一多項式，爲矩陣$\begin{pmatrix} x & y \end{pmatrix}$乘上矩陣 A 乘上矩陣$\begin{pmatrix} x \\ y \end{pmatrix}$，多項式名稱爲 f

(%o2) $(-4\,y-6\,x)\,y+x\,(x-6\,y)$

(%i3) expand(f); 展開多項式指令：expand(多項式名稱) //將多項式 f 展開得到 $x^2 - 12xy - 4y^2$

(%o3) $-4\,y^2-12\,x\,y+x^2$

(%i4) eigenvectors(A); eigenvector 也會把特徵值列出來，所以是包含前面 eigenvalues 功能的指令。特徵向量指令：eigenvector(矩陣)，第一部分和 eigenvalues 輸出一樣，這些向量會生成相對應特徵值的向量空間 //第一部分和 eigenvalues 輸出一樣，對於特徵值-8 有一個對應的特徵向量[1, $\frac{3}{2}$]，而特徵值 5 對應到的特

徵向量爲[1, $-\frac{2}{3}$]

(%o4) $[[[-8,5],[1,1]],[1,\frac{3}{2}],[1,-\frac{2}{3}]]$

(%i5) inprod([1,3/2],[1,3/2]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //向量(1, $\frac{3}{2}$)跟向量(1, $\frac{3}{2}$)作內積

(%o5) $\frac{13}{4}$

(%i6) inprod([1,-2/3],[1,-2/3]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //向量(1, $-\frac{2}{3}$)跟向量(1, $-\frac{2}{3}$)作內積

(%o6) $\frac{13}{9}$

(%i7) P:matrix([2/sqrt(13),3/sqrt(13)],[3/sqrt(13),-2/sqrt(13)]); //定義一矩陣，矩陣名稱叫做 P，第一列的元素有 $\frac{2}{\sqrt{13}},\frac{3}{\sqrt{13}}$，第二列的元素有 $\frac{3}{\sqrt{13}},-\frac{2}{\sqrt{13}}$

(%o7) $\begin{bmatrix} \frac{2}{\sqrt{13}} & \frac{3}{\sqrt{13}} \\ \frac{3}{\sqrt{13}} & -\frac{2}{\sqrt{13}} \end{bmatrix}$

(%i8) D=invert(P).A.P; //求 P 的反矩陣乘上矩陣 A 乘上矩陣 P，所求出矩陣爲一對角矩陣，名稱叫做 D

(%o8) $D=\begin{bmatrix} -8 & 0 \\ 0 & 5 \end{bmatrix}$

(%i9)
matrix([x],[y])=matrix([2/sqrt(13),3/sqrt(13)],[3/sqrt(13),-2/sqrt(13)]).matrix([x],[y]);

//可以得到新座標與舊座標的關係式

$$(\%o9)\quad \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{3\,y}{\sqrt{13}} + \frac{2\,x}{\sqrt{13}} \\ \frac{3\,x}{\sqrt{13}} - \frac{2\,y}{\sqrt{13}} \end{bmatrix}$$

(d) $3x^2 + 2xy + 3y^2$

(%i1) A:matrix([3,1],[1,3]); //定義一矩陣，矩陣名稱叫作 A，第一列元素有 3, 1，第二列元素有 1, 3

$$(\%o1)\quad \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$$

(%i2) f:matrix([x,y]).A.matrix([x],[y]); //定義一多項式，爲矩陣 $\begin{pmatrix} x & y \end{pmatrix}$ 乘上矩陣 A 乘上矩陣 $\begin{pmatrix} x \\ y \end{pmatrix}$，多項式名稱爲 f

$$(\%o2)\quad y\,(3\,y + x) + x\,(y + 3\,x)$$

(%i3) expand(f); $\boxed{\text{展開多項式指令：expand(多項式名稱)}}$ //將多項式 f 展開得到 $3x^2 + 2xy + 3y^2$

$$(\%o3)\quad 3\,y^2 + 2\,x\,y + 3\,x^2$$

(%i4) eigenvectors(A); $\boxed{\text{eigenvector 也會把特徵值列出來，所以是包含前面 eigenvalues 功能的指令。特徵向量指令：eigenvector(矩陣)，第一部分和 eigenvalues 輸出一樣，這些向量會生成相對應特徵值的向量空間}}$ //第一部分和 eigenvalues 輸出一樣，對於特徵值 4 有一個對應的特徵向量[1, 1]，而特徵值 2 對應到的特徵向量爲[1, -1]

(%o4) [[[4 , 2] , [1 , 1]] , [1 , 1] , [1 , - 1]]

(%i5) inprod([1,1],[1,1]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //向量(1, 1)跟向量(1, 1)作內積

(%o5) 2

(%i6) inprod([1,-1],[1,-1]); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字 //向量(1, -1)跟向量(1, -1)作內積

(%o6) 2

(%i7) P:matrix([1/sqrt(2),1/sqrt(2)],[1/sqrt(2),-1/sqrt(2)]); //定義一矩陣，矩陣名稱叫做 P，第一列的元素有 $\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}$ ，第二列的元素有 $\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}$

(%o7) $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$

(%i8) D=invert(P).A.P; //求 P 的反矩陣乘上矩陣 A 乘上矩陣 P，所求出矩陣爲一對角矩陣，名稱叫做 D

(%o8) $D=\begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}$

(%i9)
matrix([x],[y])=matrix([1/sqrt(2),1/sqrt(2)],[1/sqrt(2),-1/sqrt(2)]).matrix([x],[y]);

//可以得到新座標與舊座標的關係式

(%o9) $\begin{bmatrix} x \\ y \end{bmatrix}=\begin{bmatrix} \frac{y}{\sqrt{2}}+\frac{x}{\sqrt{2}} \\ \frac{x}{\sqrt{2}}-\frac{y}{\sqrt{2}} \end{bmatrix}$

(e) $x^2-2xy+y^2$

(%i1) A:matrix([1,-1],[-1,1]); //定義一矩陣，矩陣名稱叫作 A，第一列元素有 1, -1，第二列元素有-1, 1

(%o1) $\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$

(%i2) f:matrix([x,y]).A.matrix([x],[y]); //定義一多項式，爲矩陣 $\begin{pmatrix} x & y \end{pmatrix}$ 乘上矩陣 A 乘上矩陣 $\begin{pmatrix} x \\ y \end{pmatrix}$，多項式名稱爲 f

(%o2) $y(y-x)+x(x-y)$

(%i3) expand(f); $\boxed{\text{展開多項式指令：expand(多項式名稱)}}$ //將多項式 f 展開得到 $x^2-2xy+y^2$

(%o3) $y^2-2xy+x^2$

(%i4) eigenvectors(A); $\boxed{\text{eigenvector 也會把特徵值列出來，所以是包含前面 eigenvalues 功能的指令。特徵向量指令：eigenvector(矩陣)，第一部分和 eigenvalues 輸出一樣，這些向量會生成相對應特徵值的向量空間}}$ //第一部分和 eigenvalues 輸出一樣，對於特徵值 0 有一個對應的特徵向量[1, 1]，而特徵值 2 對應到的特徵向量爲[1, -1]

(%o4) [[[0,2],[1,1]],[1,1],[1,-1]]

(%i5) inprod([1,1],[1,1]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //向量(1, 1)跟向量(1, 1)作內積

(%o5) 2

(%i6) inprod([1,-1],[1,-1]); $\boxed{\text{內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字}}$ //向量(1, -1)跟向量(1, -1)作內積

(%o6) 2

(%i7) P:matrix([1/sqrt(2),1/sqrt(2)],[1/sqrt(2),-1/sqrt(2)]); //定義一矩陣，矩陣名稱叫做 P，第一列的元素有 $\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}$，第二列的元素有 $\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}}$

(%o7) $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$

(%i8) D=invert(P).A.P; //求 P 的反矩陣乘上矩陣 A 乘上矩陣 P，所求出矩陣為一對角矩陣，名稱叫做 D

(%o8) $D=\begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}$

(%i9)
matrix([x],[y])=matrix([1/sqrt(2),1/sqrt(2)],[1/sqrt(2),-1/sqrt(2)]).matrix([x],[y]);

//可以得到新座標與舊座標的關係式

(%o9) $\begin{bmatrix} x \\ y \end{bmatrix}=\begin{bmatrix} \frac{y}{\sqrt{2}}+\frac{x}{\sqrt{2}} \\ \frac{x}{\sqrt{2}}-\frac{y}{\sqrt{2}} \end{bmatrix}$

6.6 正交投影與譜定理

6.7 奇異值分解及擬逆變換

2.令 T：V→W 是秩為 r 的線性變換，其中 V 及 W 為有限維度內積空間，對下列各小題，求單範正交基底$\{v_1,v_2,...,v_n\}$給 V 及$\{u_1,u_2,...,u_m\}$給 W，且求 T 的非零奇異值$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r$ 滿足 T(v_r)=$\sigma_i v_i$ ，對 $1 \leq i \leq r$

(a) T：R^2→R^3 定義為 T(x_1,x_2)=(x_1, x_1+x_2, x_1-x_2)

(b) T：P_2(R)→P_1(R)，其中 T($f(x_1)$)= $f''(x)$ ，且內積如例 1 所定義的

(c)令 V=W=span({1, sinx, cosx})具內積<f, g>=$\int_0^{2\pi} f(t)g(t)dt$ ，且 T 定義為
T(f)= $f'+2f$

(d) T：C^2→C^2 定義為 T(z_1,z_2)=($(1-i)z_2,(1-i)z_1+z_2$)

3.求下面各矩陣的奇異值分解

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

(b) $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & -1 \end{pmatrix}$

(c) $\begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$

(d) $\begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{pmatrix}$

(e) $\begin{pmatrix} 1+i & 1 \\ 1-i & -i \end{pmatrix}$

(f) $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & -2 & 1 \\ 1 & -1 & 1 & 1 \end{pmatrix}$

4.求下面各矩陣的極分解

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

(b) $\begin{pmatrix} 20 & 4 & 0 \\ 0 & 0 & 1 \\ 4 & 20 & 0 \end{pmatrix}$

6.使用 3 的結果，求下面各矩陣的擬反矩陣

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

(b) $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & -1 \end{pmatrix}$

(c) $\begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$

(d) $\begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{pmatrix}$

(e) $\begin{pmatrix} 1+i & 1 \\ 1-i & -i \end{pmatrix}$

(f) $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & -2 & 1 \\ 1 & -1 & 1 & 1 \end{pmatrix}$

8.對下面各線性方程組

(Ⅰ)若方程組爲相容，求唯一的具最小範數解

(Ⅱ)若方程組爲矛盾，求最小範數解的最佳近似，如定理 6.30(b)所描述的(使用您的答案至 6(a)及 6(b))

(a) $\begin{aligned} x_1 + x_2 &= 1 \\ x_1 + x_2 &= 2 \\ -x_1 - x_2 &= 0 \end{aligned}$

(b) $\begin{aligned} x_1 + x_2 + x_3 + x_4 &= 2 \\ x_1 - 2x_3 + x_4 &= -1 \\ x_1 - x_2 + x_3 + x_{4=2} \end{aligned}$

6.8 雙線性與二次形式

5.驗證已知各映射是雙線性形式，再求出相對於已知有序基底的矩陣表示

(a) H：$R^3 \times R^3 \to R$，其中 $H\left(\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}, \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}\right) = a_1 b_1 - 2a_1 b_2 + a_2 b_1 - a_3 b_3$，令

$\beta = \left\{\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}\right\}$

(b)設 V=$M_{2\times 2}(R)$，有序基底 $\beta = \{\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}\}$，定義 H：V×V→R 爲 H(A, B)=tr(A)$\bullet$tr(B)

(c)令 β={(cost, sint, cos2t, sin2t)}，則 β 是 V=span(β)的一組有序基底，且 V 是一個所有在 R 上連續函數所成空間之 4 維子空間，令函數 H：V×V→R 被定義爲 H(f, g)= $f'(0) \bullet g''(0)$

17.給定下列二形式 K 佈於實內積空間 V，求一種對稱雙線性形式 H 使 K(x)=H(x, x)，∀x∈V，再找出 V 的一組單範正交基底 β 使 Ψ_β(H)是對角矩陣

(a) K：$R^2 \to R$ 定義爲 $K\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = -2t_1^2 + 4t_1 t_2 + t_2^2$

(b) K：$R^2 \to R$ 定義爲 $K\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = 7t_1^2 - 8t_1 t_2 + t_2^2$

(c) K：$R^3 \to R$ 定義爲 $K\begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} = 3t_1^2 + 3t_2^2 + 3t_3^2 - 2t_1 t_3$

22.對下列佈於實數的各矩陣 A，求一對角矩陣 D 及可逆矩陣 Q 使得 $Q^{-1}AQ=D$

(a) $\begin{pmatrix} 1 & 3 \\ 3 & 2 \end{pmatrix}$

(b) $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

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

6.9 愛因斯坦特殊相對論

6.10 附件問題與雷利商

2.計算下列矩陣的範數

(a) $\begin{pmatrix} 4 & 0 \\ 1 & 3 \end{pmatrix}$

(%i1) load(eigen)$ 我們要算一矩陣的 eigenvalus 要使用 eigen 這個模組，此模組提供了 eigenvalus 指令去計算 eigenvalues。 //讀取 eigen 模組
A:matrix([4,0],[1,3]); //定義一矩陣，矩陣名稱叫作 A，第一列元素有 4, 0，第二列元素有 1, 3

(%o2) $\begin{bmatrix} 4 & 0 \\ 1 & 3 \end{bmatrix}$

(%i3) B:transpose(A); 轉置指令：transpose(矩陣) //對矩陣 A 做轉置，轉置後的矩陣名稱定爲 B

(%o3) $\begin{bmatrix} 4 & 1 \\ 0 & 3 \end{bmatrix}$

(%i4) C:B.A; 矩陣相乘用「.」 //矩陣 A 乘上矩陣 B，計算出來的矩陣名稱叫做 C

(%o4) $\begin{bmatrix} 17 & 3 \\ 3 & 9 \end{bmatrix}$

(%i5) eigenvalues(C); 特徵值的指令：eigenvalue(矩陣)，結果中會出現 2 個 list，第一個 list 中是真正的特徵值，而第二個 list 則是代表每個特徵值的幾何重數，也就是每個特徵值對應的特徵向量空間之維度 //計算 C 之特徵值，真正的特徵值是放在結果的第一個 list 中，也就是 18 跟 8，而第二個 list 代表每個特徵值的幾何重數，也就是每個特徵值對應的特徵向量空間之維度

(%o5) [[18 , 8] , [1 , 1]]

(%i6) norm=sqrt(18); //可得出矩陣 C 的範數為 $\sqrt{18}$

(%o6) $norm = 3\sqrt{2}$

(b) $\begin{pmatrix} 5 & 3 \\ -3 & 3 \end{pmatrix}$

(%i1) load(eigen)$ 我們要算一矩陣的 eigenvalus 要使用 eigen 這個模組，此模組提供了 eigenvalus 指令去計算 eigenvalues。 //讀取 eigen 模組

A:matrix([5,3],[-3,3]); //定義一矩陣，矩陣名稱叫作 A，第一列元素有 5, 3，第二列元素有-3, 3

(%o2) $\begin{bmatrix} 5 & 3 \\ -3 & 3 \end{bmatrix}$

(%i3) B:transpose(A); 轉置指令：transpose(矩陣) //對矩陣 A 做轉置，轉置後的矩陣名稱定為 B

(%o3) $\begin{bmatrix} 5 & -3 \\ 3 & 3 \end{bmatrix}$

(%i4) C:B.A; 矩陣相乘用「.」 //矩陣 A 乘上矩陣 B，計算出來的矩陣名稱叫做 C

(%o4) $\begin{bmatrix} 34 & 6 \\ 6 & 18 \end{bmatrix}$

(%i5) eigenvalues(C); 特徵值的指令：eigenvalue(矩陣)，結果中會出現 2 個 list，第一個 list 中是真正的特徵值，而第二個 list 則是代表每個特徵值的幾何重數，也就是每個特徵值對應的特徵向量空間之維度 //計算 C 之特徵值，真正的特徵值是放在結果的第一個 list 中，也就是 36 跟 16，而第二個 list 代表每個特徵值

的幾何重數，也就是每個特徵值對應的特徵向量空間之維度

(%o5) [[36, 16], [1, 1]]

(%i6) norm=sqrt(36);　//可得出矩陣 C 的範數爲 $\sqrt{36}$

(%o6) $norm = 6$

(c) $\begin{pmatrix} 1 & \frac{-2}{\sqrt{3}} & 0 \\ 0 & \frac{-2}{\sqrt{3}} & 1 \\ 0 & \frac{2}{\sqrt{3}} & 1 \end{pmatrix}$

(%i1) load(eigen)$　$\boxed{\text{我們要算一矩陣的 eigenvalus 要使用 eigen 這個模組，此模組提供了 eigenvalus 指令去計算 eigenvalues。}}$　//讀取 eigen 模組

A:matrix([1,-2/sqrt(3),0],[0,-2/sqrt(3),1],[0,2/sqrt(3),1]);　//定義一矩陣，矩陣名稱叫作 A，第一列元素有 1, $-\frac{2}{\sqrt{3}}$, 0，第二列元素有 0, $-\frac{2}{\sqrt{3}}$, 1，第三列元素有 0, $\frac{2}{\sqrt{3}}$, 1

(%o2) $\begin{bmatrix} 1 & -\frac{2}{\sqrt{3}} & 0 \\ 0 & -\frac{2}{\sqrt{3}} & 1 \\ 0 & \frac{2}{\sqrt{3}} & 1 \end{bmatrix}$

(%i3) B:transpose(A);　$\boxed{\text{轉置指令：transpose(矩陣)}}$　//對矩陣 A 做轉置，轉置後的矩陣名稱定爲 B

(%o3) $\begin{bmatrix} 1 & 0 & 0 \\ -\frac{2}{\sqrt{3}} & -\frac{2}{\sqrt{3}} & \frac{2}{\sqrt{3}} \\ 0 & 1 & 1 \end{bmatrix}$

(%i4) C:B.A; $\boxed{\text{矩陣相乘用「.」}}$ //矩陣 A 乘上矩陣 B，計算出來的矩陣名稱叫做 C

$$(\%o4)\quad \begin{bmatrix} 1 & -\frac{2}{\sqrt{3}} & 0 \\ -\frac{2}{\sqrt{3}} & 4 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

(%i5) eigenvalues(C); $\boxed{\text{特徵值的指令：eigenvalue(矩陣)，結果中會出現 2 個 list，第一個 list 中是真正的特徵值，而第二個 list 則是代表每個特徵值的幾何重數，也就是每個特徵值對應的特徵向量空間之維度}}$ //計算 C 之特徵值，真正的特徵值是放在結果的第一個 list 中，也就是 18 跟 8，而第二個 list 代表每個特徵值的幾何重數，也就是每個特徵值對應的特徵向量空間之維度

$$(\%o5)\quad [[-\frac{\sqrt{129}-15}{6}, \frac{\sqrt{129}+15}{6}, 2], [1, 1, 1]]$$

(%i6) norm=sqrt((sqrt(129)+15)/6); //可得出矩陣 C 的範數爲 $\frac{\sqrt{\sqrt{129}+15}}{\sqrt{6}}$

$$(\%o6)\quad norm=\frac{\sqrt{\sqrt{129}+15}}{\sqrt{6}}$$

4.令 A 和 A^{-1} 如下：

A= $\begin{pmatrix} 6 & 13 & -17 \\ 13 & 29 & -38 \\ -17 & -38 & 50 \end{pmatrix}$ 及 $A^{-1}=\begin{pmatrix} 6 & -4 & 1 \\ -4 & 11 & 7 \\ -1 & 7 & 5 \end{pmatrix}$，A 的特徵值近似 84.74，0.2007, 及 0.0588

(a) 近似 $\|A\|, \|A^{-1}\|$, 及 cond(A)(參考 3)

(%i1) norm(A)=84.74;　//由 3 知，由於 A 是對稱矩陣，則 A 的範數是 A 的最大特徵值

(%o1) norm(A)= 84.74

(%i2) norm(inv(A))=1/0.0588;　//A 的反矩陣的範數則爲 1/最小特徵值

(%o2) norm(inv(A))= 17.00680272108844

(%i3) cond(A)=84.74*17.00680272108844;　//矩陣 A 的條件數爲 $\|A\| \bullet \|A^{-1}\|$

(%o3) cond(A)= 1441.156462585034

(b) 假設有二個向量 x 及 $\tilde{x}$ 滿足 Ax=b 及 $\|b - A\tilde{x}\| \le 0.001$，利用(a)來決定

$\|\tilde{x} - A^{-1}b\|$ (絕對誤差)的上界及 $\|\tilde{x} - A^{-1}b\| / \|A^{-1}b\|$ (相對誤差)

5.設 x 是 Ax=b 的真正解且用電腦產生近似解 $\tilde{x}$，若 con(A)=100，$\|b\|=1$，且

$\|b - A\tilde{x}\| = 0.1$，則求出 $\|x - \tilde{x}\| / \|x\|$ 的上界及下界

6.令 B= $\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$，試求 R $\begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix}$，$\|B\|$，及 cond(B)

(%i1) load(eigen)$　我們要算一矩陣的 eigenvalus 要使用 eigen 這個模組，此模組提供了 eigenvalus 指令去計算 eigenvalues。　//讀取 eigen 模組
B:matrix([2,1,1],[1,2,1],[1,1,2]);　//定義一矩陣，矩陣名稱叫作 B，第一列元素有 2, 1, 1，第二列元素有 1, 2, 1，第三列元素有 1, 1, 2

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

(%i3) x:matrix([1],[-2],[3]);　//定義一矩陣，矩陣名稱叫作 x，第一列元素有 1，第二列元素有-2，第三列元素有 3

$$(\%o3)\quad \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix}$$

(%i4) inprod(B.x,x); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字　//矩陣 B 乘上矩陣 x 跟矩陣 x 作內積

(%o4) 18

(%i5) inprod(x,x); 內積的指令：inprod([向量，向量])，inprod 是 innerproduct 的同義字　//向量 x 跟向量 x 作內積

(%o5) 14

(%i6) R(x)=18/14;　//minR(x)是自伴矩陣 B 的最小特徵值

$$(\%o6)\quad R\left(\begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix}\right)=\frac{9}{7}$$

(%i7) eigenvalues(B); 特徵值的指令：eigenvalue(矩陣)，結果中會出現 2 個 list，第一個 list 中是真正的特徵值，而第二個 list 則是代表每個特徵值的幾何重數，也就是每個特徵值對應的特徵向量空間之維度　//計算 B 之特徵值，真正的特徵值是放在結果的第一個 list 中，也就是 4 跟 1，而第二個 list 代表每個特徵值的幾何重數，也就是每個特徵值對應的特徵向量空間之維度

(%o7) [[4,1],[1,2]]

(%i8) norm(B)=4;　//矩陣 B 的範數爲 B 的最大特徵值

$$(\%o8)\quad norm\left(\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}\right)=4$$

(%i9) cond(B)=4*1;　//B 的條件數等於 B 的範數乘上 B 的反矩陣的範數，也就是矩陣 B 的最大特徵值乘上 B 的最小特徵值

$$\text{(\%o9)}\quad \mathrm{cond}\left(\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}\right) = 4$$

6.11 正交算子的幾何性

3.令 A=$\begin{pmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{pmatrix}$及 B=$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

(a)試證 L_A 是一種鏡射

(b)找出 L_A 鏡射 R^2 所對應的軸，亦即 L_A 作用爲單位變換在 R^2 的子空間

(c)試證 L_{AB} 與 L_{BA} 皆是旋轉

4.任給實數ϕ，令 A=$\begin{pmatrix} \cos\phi & \sin\phi \\ \sin\phi & -\cos\phi \end{pmatrix}$，

(a)試證 L_A 是一種鏡射

(b)找出 L_A 鏡射 R^2 所對應的軸