Pràctiques Integrades
1er de Matemàtiques

Derivades. Congruències. Espais vectorials

Curs 2003–04

Índex

Part IV

Derivades. Congruències. Espais vectorials

13 Càlcul de derivades i determinació de les característiques del gràfic d'una funció

Amb Maple es pot obtenir fàcilment la derivada d'una expressió `expr` qualsevol. Com ja deveu saber la comanda que fa aquesta feina és `diff(expr,var)` on `expr` és l'expressió que es vol derivar i `var` és la variable respecte la que es farà la derivada.

Exemple 13.1

Podem fer la derivada de l'expressió $x^3 - 3x^2 + 5x + 4$ respecte x amb

```
> diff(x^3-3*x^2+5*x+4,x);
```

i si tenim definida una funció $f(x)$ també podrem fer el mateix

```
> f:= x-> x^3-3*x^2+5*x+4;
> diff(f(x),x);
```

Teniu en compte, però, que segons com us poseu a fer els càlculs podeu tenir sorpreses. Podeu preveure quin serà el resultat de l'operació següent després d'haver definit la funció `f`?

```
> diff(f,x);
```

Noteu que Maple necessita conèixer respecte quina variable s'ha de fer la derivada ja que en una mateixa expressió hi poden haver diferents paràmetres. Per exemple, les comandes

Exemple 13.2

```
> diff(x^2*cos(b)-exp(y^2)*tan(x),x);
> diff(x^2*cos(b)-exp(y^2)*tan(x),y);
> diff(x^2*cos(b)-exp(y^2)*tan(x),b);
```

produeixen tres resultats diferents.

Quan treballem amb la comanda `diff` sovint el que ens interessa és definir una nova funció que és la derivada d'una $f(x)$ que ja tenim definida. Com que `diff` treballa amb expressions i dóna com a resultat expressions, aquest procés no és tan immediat com es podria pensar. Si proveu de fer

```
> f:= x-> (x^3-4)/(x^2+3);
> g:= x-> diff(f(x),x);
> g(x);
> g(1);
> g(x+1);
```

veureu que els resultats semblen no tenir massa lògica. Sabeu dir per què Maple dóna aquests resultats?

A part del mecanisme de la comanda `unapply`, que ja hem vist com funciona en una pràctica anterior, quan tenim una funció `f` que depèn d'una variable podem obtenir una nova funció `g` que és la derivada de la primera amb la comanda `D( )`.

Exemple 13.3

Si definim la funció $f(x) = x^3 - 4x$ i volem tenir la funció $g(x) = f'(x)$ (que òbviament ha de ser $g(x) = 3x^2 - 4$) es pot fer

```
> f:= x-> x^3-4*x;
> g:= D(f);
> g(x);
> g(1);
```

Quan es té definida una funció `f` que depèn de més d'una variable es pot especificar respecte quina de les variables es vol fer la derivada utilitzant `D[i](f)` on `i` és un enter que indica que es vol fer la derivada de `f` respecte la variable que ocupa el lloc `i` en la llista dels seus arguments (de fet, per a Maple els arguments d'una funció no són una *llista* si no que són una *seqüència*).

Exemple 13.4

Si tenim definida la funció $f(x, y, b) = x^2 \cos(b) - \exp(y^2) \tan(x)$ la primera variable és la x, la segona és la y i la tercera és la b. Per tant, fent

```
> f:= (x,y,b)-> x^2*cos(b)-exp(y^2)*tan(x);
> g1:= D[1](f);
> g2:= D[2](f);
> g3:= D[3](f);
```

obtindrem tres funcions `g1, g2, g3` que són les derivades de `f` respecte `x, y, b` respectivament.

Exercici 13.1

Definiu una funció `h` que sigui $h(x, y) = x\dfrac{\exp(xy)\sin(y^2)}{\ln(x^2 + y^2 + 2)}$ i calculeu

- La funció `h1` derivada de `h` respecte `x`.
- La funció `h2` derivada de `h` respecte `y`.
- Els valors de `h1` i `h2` per a `x=1` i `y=Pi/4`.
- La funció `h11` derivada segona de la funció `h` respecte `x`.

És clar que la resposta a l'últim apartat de l'exercici anterior pot ser

```
> h11:= D[1](D[1](h));
```

Com que això no és massa pràctic hi ha dreceres que ens permeten fer el mateix amb més facilitat. Una expressió de la forma `x$n` és equivalent a una successió de `n` còpies de l'expressió `x`. Per exemple

```
> x$4;
```

produeix `x,x,x,x`. D'aquesta forma en les comandes `diff` o `D` podem fer

```
> diff(sin(x)-x*(cos(x))^3,x$4);
> D[1$4](cos)(x);
```

per a obtenir les derivades quartes.

13.1 Estudi del comportament del gràfic d'una funció

Un dels problemes més clàssics del càlcul infinitessimal elemental és el de determinar totes les característiques remarcables del comportament d'una funció (asímptotes, creixement, extrems, convexitat, punts d'inflexió). La comanda `plot`, que dibuixa el gràfic d'una funció, ajuda molt a determinar algunes d'aquestes característiques però és clar que la determinació exacta (o amb una bona aproximació) dels punts remarcables no es podrà fer sense solucionar algunes equacions, determinar algun límit i calcular les derivades que calgui. Amb Maple tots els càlculs necessaris es podran fer sense massa problemes i en l'exemple següent podeu veure com fer tot el procés d'una manera sistemàtica.

Exemple 13.5

Considerem la funció $f(x) = \dfrac{5x}{x^2 - 4}$ i determinem totes les característiques del seu gràfic.

En primer lloc definim la funció i les derivades primera i segona

```
> f:=x-> (5*x)/(x^2-4);
> fp:= D(f);
> fpp:= D(fp);
```

A continuació podem calcular els límits de `f` a l'infinit (encara que en aquest cas tothom hauria de poder veure el resultat sense Maple)

```
> limit(f(x),x=infinity);
> limit(f(x), x=-infinity);
```

Seguidament, i tenint en compte que l'expressió amb la que estem tractant és una fracció, veiem que hi ha asímptotes verticals ja que els límits

```
> limit(f(x),x=2,right);limit(f(x),x=2,left);
> limit(f(x), x=-2,right);limit(f(x), x=-2,left);
```

són infinits.

El següent pas consistirà en determinar els extrems locals i els intervals de creixement i decreixement de la funció

```
> solve(fp(x));
> solve(fp(x)>0);
> solve(fp(x)<0);
```

Finalment es pot determinar la convexitat i els punts d'inflexió

```
> solve(fpp(x));
> solve(fpp(x)>0);
> solve(fpp(x)<0);
```

Com que qualsevol estudi d'aquest tipus no està complet si no es fa un dibuixet podeu fer l'exercici següent.

Exercici 13.2

Feu un gràfic en el que apareixen:

- Les asímptotes de f dibuixades en color verd.
- Els punts d'extrem local de la funció f.
- Els punts d'inflexió del gràfic de f .
- El gràfic de la funció f amb colors diferents d'acord amb la convexitat que tingui (vermell per un tipus de convexitat i marró per l'altre).

Nota: Mireu en l'ajuda de Maple quina opció de `plot` es pot utilitzar per a fer que es tinguin en compte les possibles discontinuïtats a l'hora de dibuixar un gràfic.

Exercici 13.3

Per a practicar una mica més podeu mirar de fer una cosa semblant amb les funcions següents.

$$f(x) = (x^2 - 3x + 2)\exp(x) \qquad g(x) = \frac{x^2 - x}{16 - x^2} \qquad h(x) = 2(\cos^2 x) - x^2$$

Fent per a cada una d'elles dos gràfics de colors, un on es posi de manifest els intervals de creixement i decreixement i l'altre on es mostri la convexitat.

14 Treballar amb nombres enters i congruències

Maple proporciona funcions per a realitzar les principals manipulacions de l'aritmètica amb nombres enters. Veurem a continuació algunes de les més significatives.

14.1 Nombres primers

Si us en recordeu, en la primera pràctica ja es presentava la funció `ifactor`, que determina la descomposició en factors primers d'un nombre enter. Com que el resultat que dóna aquesta funció no és fàcilment manipulable (en particular, no és fàcil extreure d'aquest resultat quins són els primers que divideixen al nombre que tenim, ni les potències d'aquests factors) existeix també la funció `ifactors` que dóna com a resultat una llista amb els factors primers, i les potències d'aquests factors, amb els que descompon un nombre enter donat.

Exercici 14.1

Considereu $n = 2^3 \cdot 5^4 \cdot 13^5$. Intenteu tornar a obtenir els factors 2, 5 i 13 i les potències respectives 3, 4 i 5 del resultat de la comanda `ifactor(n);`.

Utilitzeu ara `ifactors(n);` (i alguna cosa més) per a recuperar, separadament, els valors dels factors primers i de les potències corresponents de la descomposició del nombre n posant aquests resultats en dues llistes (una que sigui `f` per als factors i l'altra que sigui `p` per a les potències).

També és possible verificar si un nombre donat és primer o no sense necessitat de fer la seva descomposició. La funció que s'encarrega d'això és `isprime`. Podeu veure en l'exemple següent com funciona.

Exemple 14.1

```
> isprime(4);
> isprime(7);

> provap:= n-> if isprime(n) then print("és primer")
> else print("no és primer") end if;

> provap(456);
```

Noteu que la funció dóna com a resultat un valor cert o fals que es pot utilitzar com a condicional d'una comanda `if`.

Altres comandes relacionades amb els nombres primers són `ithprime(i)` (que dóna el primer que ocupa el lloc `i` en la llista dels primers ordenats), `nextprime(i)` (que dóna com a resultat el nombre primer més petit dels que són més grans que el nombre `i`) i `prevprime(i)` (que dóna el primer que és més gran entre els que són menors que `i`). Practiqueu amb alguns valors de `i` i mireu què passa quan `i` és molt gran.

14.2 Divisió entera

Cada cop que realitzem una divisió amb enters, són igualment significatius el quocient i la resta d'aquesta divisió. Les comandes de Maple `iquo(m,n)` i `irem(m,n)` donen com a resultat nombres q i r tals que $m = q \cdot n + r$ (amb $|r| < |n|$ i m i r amb el mateix signe).

Exemple 14.2

```
> iquo(158,18);
```

Les comandes `iquo` i `irem` tenen un tercer argument opcional que assigna a una variable qualsevol el valor del resultat de l'altra comanda. Així podem fer, per exemple,

Exemple 14.3

```
> iquo(487,18,'r');
> r;
> irem(487,18,'q');
> q;
```

(Les cometes de `'q'` i `'r'` serveixen per a poder fer l'assignació de valors sense preocuparse si les variables `q` i `r` ja tenien valors assignats).

Utilitzant aquestes instruccions podeu fer l'exercici següent:

Exercici 14.2

Feu un procediment que tingui com arguments un parell de nombres enters i que vagi donant els resultats parcials dels càlculs que es realitzen per a calcular el màxim comú divisor d'aquests dos nombres utilitzant l'algoritme d'Euclides (s'hauria de veure els diferents dividends i divisors i els quocients i restes de cada divisió, fins arribar a la divisió de resta 0).

Apliqueu la funció a uns quants parells de nombres triats a l'atzar.

Tot i que el procediment de l'exercici anterior calcula el màxim comú divisor de dos nombres, com ja podeu suposar ja hi ha comandes de Maple programades que calculen el màxim comú divisor i el mínim comú múltiple de dos nombres enters. La comanda `igcd(m1, m2, m3,...)` dóna com a resultat el màxim comú divisor dels nombres enters `m1, m2, m3,...`, mentre que la comanda `ilcm(m1, m2, m3, ...)` calcula el mínim comú múltiple.

Exemple 14.4

Proveu les comandes anteriors amb

```
> igcd(868,784,900);
> ilcm(25,350,875);
```

i amb un parell d'exemples triats per vosaltres mateixos.

Existeix també una altra comanda que calcula el màxim comú divisor de dos nombres enters, encara més útil que `igcd`. La comanda `igcdex(m,n)` dóna igualment el màxim comú divisor de `m` i `n` però té un parell d'arguments opcionals que permeten conèixer un parell de nombres `a, b` tals que

$$\texttt{igcd(m,n)} = \texttt{a} \cdot \texttt{m} + \texttt{b} \cdot \texttt{n}$$

En l'exemple següent podeu veure un exemple de com fer servir aquesta comanda

Exemple 14.5

```
> igcdex(258,438);
> igcdex(258,438,'a','b');
> a;
> b;
> a*258+b*438;
```

Noteu que s'ha fet servir la mateixa estratègia de les comentes `' '` que en les comandes `iquo` i `irem` per assignar a les variables `a` i `b` el valors que ens interessava guardar.

14.3 Reducció mòdul p

Maple també té funcions per a treballar *mòdul p*. El resultat de l'expressió `m mod p` o de la funció `modp(m,p)` és el residu entre 0 i $p-1$ de m mòdul p.

Exemple 14.6

Podeu provar:

```
> 378946 mod 7;
> 345*267 mod 4;
> 985*x mod 7;
> modp(378946, 7);
> modp(345*267, 4);
> modp(985*x, 7);
```

Amb `mod` i `modp` també podeu mirar si dos nombres coincideixen, o no, mòdul un p donat.

Exemple 14.7

```
> evalb(378946 mod 7= 1 mod 7);
> evalb(345*267 mod 4=2 mod 4);
```

Si en lloc d'utilitzar `modp(m,p)` utilitzeu `mods(m,p)` (mòdul amb signe) obtindreu un resultat entre $-\frac{p}{2}$ i $\frac{p}{2}$.

14.4 Solucions enteres d'equacions

Existeix també una comanda específica per a determinar *totes* les solucions enteres d'una equació. Aquesta comanda és `isolve` i s'utilitza gairebé de la mateixa manera que `solve`. És a dir, la sintaxi normal serà `isolve({eq1, eq2, ...})` on `eq1, eq2, ...` són les equacions de les que es volen trobar les solucions enteres comuns.

Exemple 14.8

Les solucions enteres del sistema d'equacions $\left.\begin{array}{r} x^3 + a \cdot x = 14 \\ a^2 - x = 7 \end{array}\right\}$ es poden determinar amb

```
> isolve({x^3+a*x=14, a^2-x=7});
```

Exercici 14.3

Quines són les solucions enteres de l'equació dels triangles rectangles $x^2 + y^2 = z^2$?

Si heu vist la manera en que es mostren les solucions de l'equació dels triangles pitagòrics de l'exercici anterior, veureu que és molt útil la possibilitat d'afegir un argument opcional a la comanda `isolve`, que permet assignar un nom concret als diferents paràmetres que poden sortir en el resultat. En l'exemple següent podeu veure com fer una construcció d'aquest estil

Exemple 14.9

```
> isolve(x^2+y^2=5*z^2,{a,b,c});
```

Com podreu veure, els paràmetres dels que depèn la solució seran `a, b, c`.

Quina és la solució de l'equació anterior si preneu $a = 1$, $b = -1$ i $c = 3$?

Tingueu en compte que el problema de determinar les solucions enteres d'una equació arbitrària és tremendament complicat i que Maple *només fa el que sap i pot.* Per exemple, Maple no queda gaire bé intentant resoldre l'equació $x^3 - 3y^2 = 0$ (que, com a mínim es pot solucionar amb $x = 0$, $y = 0$ i amb $x = y = 3$).

Exemple 14.10

```
> isolve(x^3-3*y^2=0);
```

14.5 Solucions de sistemes de congruències

Un problema bastant típic quan es treballa amb enters, principalment quan es plantegen qüestions relatives a reparticions, consisteix en resoldre un sistema de congruències de la forma

$$x \equiv a \pmod{\alpha}$$
$$x \equiv b \pmod{\beta}$$

(o potser amb més equacions).

Quan α i β no tenen factors comuns (el seu màxim comú divisor és 1) la comanda de Maple `chrem` pot obtenir directament les solucions. Si es vol resoldre el sistema anterior, la sintaxi que s'ha d'utilitzar és `chrem([a, b], [alpha, beta])` i el resultat serà l'únic nombre enter entre 0 i $\alpha \cdot \beta - 1$ que compleix les dues equacions (totes les altres solucions es poden obtenir sumant múltiples de $\alpha \cdot \beta$ a aquesta solució).

Exemple 14.11

```
> s:=chrem([1,3],[5,9]);
> s mod 5; s mod 9;
> s+45 mod 5; s+45 mod 9;
```

Quan els nombres α i β no són coprimers la comanda `chrem` no funciona i, per tant, no ens permet resoldre directament el problema. Això és només un petit inconvenient, ja que Maple posa a la nostra disposició totes les eines necessàries per a poder determinar la solució de qualsevol sistema de congruències del tipus que hem plantejat al principi d'aquest apartat.

Probablement ja sabeu

a) que el sistema $x \equiv a \pmod{\alpha}$, $x \equiv b \pmod{\beta}$ té solució si, i només si, $a - b$ és un múltiple del màxim comú divisor de α i β,

b) que la diferència entre dues solucions sempre és un múltiple del mínim comú múltiple de α i β,

c) que una solució particular es pot determinar prenent k i l tals que $k \cdot \alpha - l \cdot \beta = b - a$ i considerant $x = a + k \cdot \alpha = b + l \cdot \beta$.

Exercici 14.4

Feu un procediment que utilitzi el mateix conveni que `chrem` per a interpretar els arguments (dues llistes, una dels valors a, b i l'altre dels mòduls de les equacions α, β), que determini si el sistema és compatible i que, quan ho sigui, doni com a resultat una seqüència amb una solució particular x_0 i el mínim comú múltiple de α i β per a un sistema de la forma

$$x \equiv a \pmod{\alpha}$$
$$x \equiv b \pmod{\beta}$$

Comproveu que el vostre procediment funciona per a l'equació

$$x \equiv 2 \pmod 6$$
$$x \equiv 4 \pmod 8$$

No hauria de ser ara massa complicat fer el següent

Exercici 14.5

Modifiqueu el procediment anterior per a poder resoldre un sistema de congruències amb un nombre arbitrari d'equacions. Fixeu-vos que l'únic que cal fer és anar reduint el nombre d'equacions substituint les dues primeres per la seva solució (que torna a venir donada per una equació del mateix tipus) fins a quedar-se amb un sistema de dues equacions.

15 Espais vectorials i bases

En una pràctica anterior ja heu utilitzat algunes de les funcions que conté el paquet **LinearAlgebra**. Avui estudiarem aquelles que ens permeten analitzar la dimensió d'un subespai vectorial i obtenir-ne una base de vectors.

Ja sabeu com definir matrius i vectors i també les operacions bàsiques de sumes i productes que es poden realitzar amb ells. Començarem veient algunes funcions més referents a la manipulació de matrius i vectors. Recordeu com definíem matrius i vectors amb el següent exemple.

Exemple 15.1

```
> M:=<<1,-2>|<0,1>>;
> v:=<2,3>;
> w:=<<2>|<3>>;
> M.v;
> w.M;
```

Amb les comandes `Matrix` i `Vector` també podem definir matrius i vectors. Aquestes comandes ens permeten definir matrius i vectors utilitzant una funció pels seus coeficients.

Exemple 15.2

```
> M:=Matrix([[1,-2],[0,1]]);
> v:=Vector([2,3]);
> w:=Vector[row]([2,3]);
> f:=(i,j)->x^i*y^j;
> Nf:=Matrix(2,f);
> N:=subs(x=3,y=2,Nf);
```

Hi ha d'altres operacions elementals que es poden realitzar amb matrius, per exemple transposar. Transposar una matriu consisteix en intercanviar les files per les columnes. Maple té una comanda per realitzar aquesta operació, `Transpose`. També podem calcular-ne el determinant (`Determinant`) i si aquest és diferent de zero, aleshores obtenir la matriu inversa, `MatrixInverse`. En aquest cas també podem escriure `1/A`.

Exemple 15.3

```
> Transpose(M);
> d:=Determinant(M);
> I:=MatrixInverse(M);
> I.M; M.I;
> 1/M; M^(-1);
```

També podem recuperar el nombre de files i columnes d'una matriu amb les comandes `RowDimension` i `ColumnDimension`.

Exercici 15.1

Realitzeu un procediment `dd` que tingui com a únic argument un nombre natural n i calculi el determinant d'una matriu $n \times n$ que conté x's a la diagonal i y's fora de la diagonal (noteu que l'únic que cal fer és construir la matriu i aplicar la funció `Determinant`).

15.1 El rang d'una matriu

En una pràctica anterior vam veure com resoldre sistemes d'equacions lineals.

Exercici 15.2

Digueu si els vectors següents són linealment independents o no, $(1,-3,2)$, $(2,1,-5)$ i $(2,-13,13)$.

Recordeu que el rang d'una matriu és el nombre màxim de files linealment independents. És el mateix que el màxim número de columnes linealment independents. I també es pot definir com l'ordre màxim d'una submatriu quadrada invertible. `Rank(A)` és la comanda que ens retorna el rang de la matriu A.

Exemple 15.4

```
> M:=<<1,-2>|<0,1>>; N:=<<1,-2>|<2,-4>>;
> Rank(M);
> Rank(N);
```

Noteu que la funció `Rank` pot donar resultats incorrectes si entre els coeficients de la matriu hi ha paràmetres indeterminats. Per exemple

Exemple 15.5

Si s'aplica `Rank` a la matriu $\begin{pmatrix} x & 1 \\ 1 & x \end{pmatrix}$ el resultat serà 2 mentre que és clar que per a $x = 1$ o $x = -1$ el rang és 1.

Exercici 15.3

Determineu el rang de la matriu següent

$$\begin{pmatrix} 4-x & 2 & -2+\frac{2}{3}x & 8-x & -4 \\ \frac{13}{2} & -1 & x-3 & 10 & -9/2 \\ 4x-4 & 6 & 2-\frac{4}{3}x & 3x-8 & 4 \\ 6x-4 & 3 & 2-2x & 3x-8 & 4+x \end{pmatrix}$$

en funció del paràmetre x.

Si E és un subespai vectorial generat per vectors $\{v_1, v_2, \ldots, v_n\}$, la dimensió del subespai vectorial E és el rang de la matriu que conté els vectors generadors a les files (o a les columnes). D'aquesta manera utilitzant la comanda `Rank` podem calcular la dimensió d'un subespai vectorial donada una família de vectors generadors.

Exercici 15.4

Calculeu la dimensió del subespai vectorial de $\mathbb{R}^4$ generat pels vectors $v_1 = (1, 3, -4, 5)$, $v_2 = (0, 6, 5, -1)$ i $v_3 = (1, -9, -14, 7)$.

15.2 Bases d'espais vectorials

Donada una matriu, Maple no només en sap calcular el rang sinó que a més també ens calcula una base del subespai generat tant pels vectors files com columnes. Les comandes que proporcionen aquesta informació són `RowSpace` i `ColumnSpace`. Observeu el seu funcionament en el següent exemple.

Exemple 15.6

```
> M:=<<1,-2,5,6>|<0,1,7,3>|<1,-4,-9,0>>;
> ColumnSpace(M);
> RowSpace(N);
```

Exercici 15.5

Calculeu una base del subespai vectorial E de $\mathbb{R}^4$ generat pels vectors $v_1 = (1, 3, -4, 5)$, $v_2 = (0, 6, 5, -1)$ i $v_3 = (1, -9, -14, 7)$.

Si E i F són subespais vectorial de $\mathbb{R}^n$ generats per $\{v_1, \ldots, v_e\}$ i $\{w_1, \ldots, w_f\}$ respectivament, aleshores recordeu que el subespai vectorial suma està generat per $\{v_1, \ldots, v_e, w_1, \ldots, w_f\}$.

D'altra banda, la intersecció dels subespais E i F està format per vectors de E que també es poden expressar en funció dels generadors de F. És a dir, existeixen $\lambda_i, \mu_i \in \mathbb{R}$ tals que

$$\lambda_1 v_1 + \cdots + \lambda_e v_e = \mu_1 w_1 + \cdots + \mu_f w_f$$
$$\lambda_1 v_1 + \cdots + \lambda_e v_e - \mu_1 w_1 - \cdots - \mu_f w_f = 0$$

Per tant, si M és la matriu formada pels vectors generadors de E i F, aleshores una base del subespai intersecció es pot obtenir analitzant les solucions del sistema d'equacions homogeni $M \cdot x = 0$. Donada una matriu M, Maple també ens permet obtenir una base pel subespai vectorial de solucions del corresponent sistema homogeni mitjançant la comanda `NullSpace`.

Exemple 15.7

```
> M:=<<1,-2,5,6>|<0,1,7,3>|<1,-4,-9,0>>;
> NullSpace(M);
```

Exercici 15.6

Sigui E el subespai vectorial de l'exercici anterior i F el subespai generat per $w_1 = (1, -6, -18, 12)$ i $w_2 = (3, 0, 0, 1)$. Calculeu la dimensió i bases per E, F, $E + F$ i $E \cap F$. Verifiqueu la fòrmula de Grassmann.

Exercici 15.7

Sigui $E =< (4 - \lambda), 2, 1), (2 - \lambda, 4, 2) >$ i $F =< (2 - \lambda, 4, 8) >$. Determineu les dimensions de E, F, $E + F$ i $E \cap F$ segons el paràmetre λ.

Teniu en compte que el paquet **LinearAlgebra** conté les funcions `IntersectionBasis` i `SumBasis` que fan, exactament, el que indica el seu nom i es poden utilitzar en aquest tipus de càlculs.

També és ben conegut que qualsevol subespai de $\mathbb{R}^n$ coincideix amb l'espai de solucions d'un determinat sistema homogeni. En l'exemple següent podeu veure com obtenir aquest sistema a partir d'un conjunt de generadors del subespai.

Exemple 15.8

Considerem el subespai de $\mathbb{R}^5$ generat pels vectors $(1, 2, 2, -1, 0)$, $(0, 1, -1, 2, 3)$ i $(2, -1, -4, 5, 1)$. Definim la matriu `M` com la que té per columnes les components dels tres vectors donats

```
> M:=<<1,2,2,-1,0>|<0,1,-1,2,3>|<2,-1,-4,5,1>>;
```

afegim una columna al final que representa les components d'un vector qualsevol de $\mathbb{R}^5$

```
> Ma:= <M|<x1,x2,x3,x4,x5>>;
```

realitzem les operacions de reducció per files fins a arribar a una forma reduïda. Les equacions en `x1, x2, x3, x4, x5` corresponents a files nules donen un les equacions d'un sistema homogeni que té com espai de solucions l'espai generat pels vectors que hem considerat al principi. Donarà un resultat equivalent al sistema $13x_4 - 3x_1 - 9x_2 + 17x_3 = 0$, $13x_5 - 23x_2 + 14x_1 + 16x_3 = 0$.

15.3 Operacions elementals i matrius elementals

És un fet ben conegut que les matrius invertibles es poden escriure sempre com a producte de matrius *elementals*. Recordeu que aquestes matrius elementals s'obtenen realitzant sobre la matriu identitat una qualsevol de les operacions elementals de reducció (multiplicar una fila per un escalar no nul, intercanviar dues files entre si o sumar a una fila un múltiple de qualsevol de les altres), i que el producte (per l'esquerra) d'una matriu qualsevol amb una d'aquestes matrius elementals té el mateix efecte que realitzar l'operació elemental de reducció corresponent sobre la matriu donada.

Exemple 15.9

El procediment següent defineix una funció `MatElem` que té com arguments un nombre natural `n` i una seqüència, que segueix el conveni de les operacions elementals de la comanda `RowOperation`, per a produir com a resultat la matriu elemental d'ordre `n` corresponent a l'operació elemental codificada en els altres arguments.

```
> MatElem:= proc()
> RowOperation(IdentityMatrix(args[1]),args[2..nargs]);
> end proc;
```

Noteu que dins la definició de `MatElem` no ha fet falta especificar ni el nom, ni el nombre dels arguments. En comptes d'això hem utilitzat les variables `args` i `nargs` que donen, cada cop que s'executa el procediment i només mentre s'està executant, una llista amb tots els arguments i el nombre total d'aquests arguments.

Exercici 15.8

i) Utilitzeu la comanda `MatElem` de l'exemple anterior per a generar les matrius elementals d'ordre 4 que corresponen a les operacions: intercanviar la primera i la tercera files, sumar a la segona fila 3 vegades la quarta, multiplicar per 5 la primera fila.

ii) Considereu la matriu $M = \begin{pmatrix} 2 & -1 & 0 & 5 & 2 & 3 & 1 \\ 1 & 1 & 0 & 0 & 2 & 9 & -1 \\ 0 & 0 & -2 & -1 & 3 & 5 & 0 \\ 1 & 2 & 3 & 2 & 1 & 0 & -2 \end{pmatrix}$ i realitzeu sobre ella les operacions elementals donades en l'apartat anterior, multiplicant per les matrius elementals corresponents.

Aquest apartat començava dient que les matrius invertibles es poden escriure com a producte de matrius elementals. Recordeu que aquestes matrius elementals són justament les que corresponen a la cadena d'operacions elementals de reducció que es necessiten per a transformar una matriu invertible qualsevol en la matriu identitat. L'objectiu de l'exercici següent és fabricar un procediment que permeti visualitzar aquest fet d'una manera el més còmoda possible.

Exercici 15.9

Fabriqueu un procediment `Reducc` que admeti com arguments una matriu i una seqüència d'expressions, del tipus de les que es posen en `RowOperation`, que realitzi l'operació elemental de reducció, codificada en aquestes expressions, sobre la matriu introduïda i, a més, guardi en una variable global (per exemple `OpR`) una llista amb les matrius elementals de les operacions que es vagin fent utilitzant aquest procediment.

Exercici 15.10

Considereu la matriu invertible $M = \begin{pmatrix} 1 & 0 & -1 \\ -1 & 1 & 3 \\ 1 & -1 & -1 \end{pmatrix}$ (el seu determinant està calculat amb Maple i dóna 2). Reduïu M utilitzant `Reducc` i comproveu que el producte de les matrius elementals que queden en la llista `OpR` és la matriu inversa de M fent el producte de totes elles amb M. Després comproveu que la inversa de cada una de les matrius que surt en la llista `OpR` torna a ser una matriu elemental. Finalment comproveu que el producte de les inverses de les matrius que surten en `OpR` coincideix amb M.

15.4 PAQ reducció

Donada una matriu real A, existeixen matrius P, Q tals que PAQ és de la forma

$$\left(\begin{array}{c|c} I_r & 0 \\ \hline 0 & 0 \end{array}\right).$$

Les matrius P i Q permeten resoldre, d'una manera o altre, gairebé qualsevol dels problemes d'àlgebra lineal que es puguin plantejar en els que intervingui la matriu A.

Per a obtenir les matrius P i Q es procedeix de la següent manera. Si $A \in M_{p\times q}(\mathbb{R})$. Prenem la matriu quadrada

$$\left(\begin{array}{c|c} I_p & A \\ \hline 0 & I_q \end{array}\right)$$

i mitjançant operacions elementals de files i columnes la transformem en una matriu on A té zeros i uns a la diagonal i zeros fora, és a dir

$$\left(\begin{array}{c|c} P & \left(\begin{array}{c|c} I_r & 0 \\ \hline 0 & 0 \end{array}\right) \\ \hline 0 & Q \end{array}\right)$$

(amb r igual al rang de A). Les caixes que contenien I_p i I_q es transformen en matrius P i Q respectivament que satisfan la relació

$$PAQ = \left(\begin{array}{c|c} I_r & 0 \\ \hline 0 & 0 \end{array}\right).$$

Exercici 15.11

Determineu la PAQ-reducció de la matriu $A = \begin{pmatrix} 1 & 1 & 0 \\ 3 & 0 & 1 \\ 3 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}$

Feu un procediment que donada una matriu A en calculi la seva PAQ-reducció donant com a resultat les dues matrius P i Q tals que PAQ és de la forma $\left(\begin{array}{c|c} I_r & 0 \\ \hline 0 & 0 \end{array}\right)$ amb r igual al rang de A. (No és tan fàcil com podria semblar en un principi).