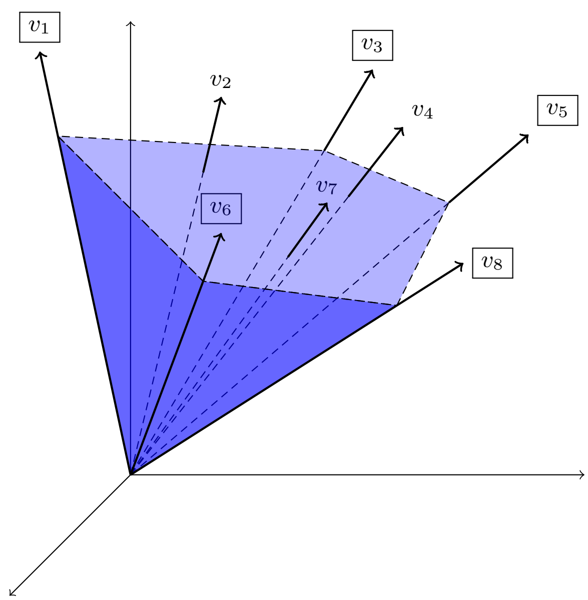
Eines Informàtiques per a les Matemàtiques

Treball individual de llenguatge `C`

Programa a lliurar fins el 31 de maig de 2020

Resum

La tasca que haurà de realitzar el programa serà, donats vectors $v_1, \ldots, v_n$ a $\mathbb{R}^3$, distingir entre aquells que generen arestes i els que són interns al con convex que generen.



Normes generals

Caldrà escriure, individualment, un programa en llenguatge `C` per dur a terme la tasca proposada. Un cop el treball lliurat i corregit, hi haurà una entrevista individual amb el professor, on caldrà mostrar i explicar el funcionament del programa.

Consells

- Recordeu que aquest és un treball *individual*. És possible, i fins i tot recomanable, intercanviar informació i idees amb els companys, però copiar està estrictament prohibit.
- En particular, *no compartiu arxius*. Entregar un programa copiat total o parcialment pot implicar suspendre *l'assignatura* (vegeu la guia docent).
- Us podeu també ajudar amb materials obtinguts d'internet, on podeu trobar codi, vídeos explicatius i altre material útil per algunes de les tasques que ha de fer el programa. Compte amb copiar *codi*! Haureu d'explicar com funciona el programa en una entrevista.
- Tots els fitxers que entregueu han de començar per `CognomNom` (on `Nom` vol dir el vostre nom i `Cognom` el vostre cognom). Per exemple, el fitxer font es pot dir `GarciaJoan.c`.
- Podeu usar qualsevol compilador i sistema operatiu en preparar el programa, però us haureu d'assegurar que useu `C` estàndard. El codi lliurat s'haurà de poder compilar amb la instrucció

```
gcc -Wall -std=c99 -pedantic -o CognomNom CognomNom.c -lm
```

1 Funcions bàsiques amb vectors de $\mathbb{R}^3$

El programa que heu de fer és força complex. Per aquest motiu hem desglossat la feina en tres seccions. En el primer bloc descrivim algunes funcions bàsiques necessàries per al programa, que ja podeu començar a programar (algunes ja les heu fet a les classes pràctiques). Mentre avanceu en aquesta part, us recomanem que aneu familiaritzant-vos amb l'algorisme explicat a les dues seccions següents, i pensant com es pot programar.

Per comprovar cada funció us recomanem escriure un programa amb una `main` molt simple, que llegeixi del teclat les dades (`scanf`) i cridi a cadascuna de les funcions amb els paràmetres adients.

1. Càlcul del producte escalar $u \cdot v$ de dos vectors a $\mathbb{R}^3$.
2. Càlcul de la norma $|u|$ d'un vector a $\mathbb{R}^3$. Recordeu que $|u| = \sqrt{u \cdot u}$.
3. Decidir si tres vectors són base de $\mathbb{R}^3$.
4. Càlcul de l'angle entre dos vectors de $\mathbb{R}^3$. S'entén en aquest apartat l'angle sense signe, entre 0 i π. Recordeu que, si α és l'angle entre els vectors u i v, es compleix
$$\cos\alpha = \frac{u \cdot v}{|u|\,|v|}.$$
5. Càlcul del determinant de tres vectors a $\mathbb{R}^3$ (és a dir, el determinant de la matriu formada pels tres vectors columna).
6. Decidir si dos vectors són perpendiculars.
7. *Projecció ortogonal.* Donats dos vectors u, v, amb $u \neq 0$, el vector
$$\varphi_u(v) = v - \frac{u \cdot v}{u \cdot u} u$$
és perpendicular a u. Fer una funció que calculi $\varphi_u(v)$. *Observació: φ_u és una aplicació lineal de $\mathbb{R}^3$ al pla $u^\perp \subset \mathbb{R}^3$ format pels vectors perpendiculars a u, i s'anomena projecció ortogonal a u).*

Recordeu que els nombres reals es guarden a les variables en forma aproximada, i que haureu d'establir una tolerància.

Proposta opcional, no necessària per al programa final: A partir del que teniu fet, ara seria molt senzill programar l'ortonormalització de Gram-Schmidt. Donats tres vectors, decidir si són base i en cas afirmatiu trobar una base ortonormal a partir d'ells pel procés de Gram-Schmidt.
`https://ca.wikipedia.org/wiki/Proc%C3%A9s_d%27ortogonalitzaci%C3%B3_de_Gram-Schmidt`

2 Base teòrica i algorisme per al cas pla

Cons i combinacions còniques

Definició 1. Sigui E un espai vectorial sobre $\mathbb{R}$ (per al treball, $E = \mathbb{R}^3$ o un subespai de $\mathbb{R}^3$). Una *combinació cònica* de vectors de E és una combinació lineal

$$a_1 v_1 + \cdots + a_n v_n$$

amb coeficients no negatius: $a_1, \ldots, a_n \geq 0$. Un *con convex* és un subconjunt $C \subset E$ tancat per combinacions còniques, és a dir,

$$C \text{ és un con convex } \iff \forall u, v \in C, \forall a, b \geq 0,\ au + bv \in C.$$

Observem que tot subespai vectorial és en particular un con, però el recíproc clarament no és cert:

Exemple 2. A la figura 1 hi ha representats dos cons convexos, un a $\mathbb{R}^2$ i un a $\mathbb{R}^3$.

Proposició 3. *Donat un con $C \subset E$, el conjunt dels vectors de C que tenen el vector oposat també a C,*

$$v(C) = \{v \in C \mid -v \in C\},$$

formen un subespai vectorial de E.

Si hi estàs interessat, pots demostrar la proposició com un exercici d'àlgebra no massa difícil.

Definició 4. El subespai $v(C)$ de la proposició 3 s'anomena *vèrtex* del con. Diem que C és un con *amb punta* si el seu vèrtex és l'origen. Un altre subespai naturalment associat a C és el subespai generat per ell, $\langle C \rangle$. Definim la dimensió de C com la dimensió de $\langle C \rangle$ en tant que espai vectorial.

Observem que un con és igual al seu vèrtex si i només si és un subespai vectorial. En tots els altres casos es té $v(C) \subsetneq C \subsetneq \langle C \rangle$.

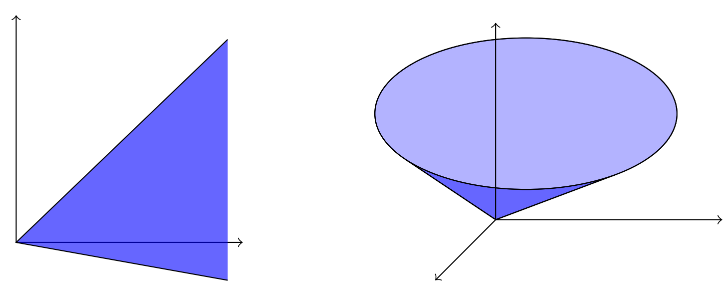


Figura 1: A l'esquerra un con pla; a la dreta un con de l'espai.

Cons de dimensió ≤ 2

L'únic con de dimensió zero és el que es redueix a un punt: $\{0\}$.

Suposem que C és un con de dimensió 1 dins $E = \mathbb{R}^3$. Per tant el subespai generat $\langle C \rangle$ és una recta $L \subset E$, i $C \subset \langle C \rangle = L \cong \mathbb{R}^1$. Sense entrar en detalls del que és un isomorfisme de cons, podem entendre que un con C contingut a L serà com un con de $E' = \mathbb{R}$.

L'espai vectorial $E' = \mathbb{R}$ conté (a més del con de dimensió zero) exactament 3 cons:

1. $C_+ = \{x \in \mathbb{R} \,|\, x \geq 0\}$ és un con; el seu vèrtex és $\{0\}$.
2. $C_- = \{x \in \mathbb{R} \,|\, x \leq 0\}$ és un con; el seu vèrtex és $\{0\}$.
3. $\mathbb{R}$, que és un con per ser subespai vectorial.

Deduïm que hi ha dos tipus de con de dimensió 1: la **semirecta** (com C_+ i C_-) i la **recta**.

Definició 5. Donat un conjunt de vectors $S \subset E$, el mínim con que conté S és el conjunt de totes les combinacions còniques de vectors de S, i s'anomena *con generat* per S,

$$\mathrm{co}(S) = \{a_1 v_1 + \cdots + a_n v_n \,|\, a_i \geq 0, v_i \in S\}.$$

Si C és un con per al qual existeix un conjunt finit S amb $C = \mathrm{co}(S)$, diem que és un con finitament generat.

Els cons de $\mathbb{R}$ són finitament generats: $C_+ = \mathrm{co}(1)$, $C_- = \mathrm{co}(-1)$, $\mathbb{R} = \mathrm{co}(1, -1)$. En canvi, en dimensió ≥ 2 hi ha cons que no són finitament generats; per exemple, el con de la dreta a la figura 1 no és finitament generat. En aquest treball només ens interessen els cons finitament generats.

Un con $C \subset E = \mathbb{R}^3$ de dimensió 2 estarà contingut en un pla, i per tant serà com un con de dimensió 2 a $\mathbb{R}^2$. Classifiquem els cons finitament generats de dimensió 2 segons la dimensió del seu vèrtex:

Angle del pla Si $\dim v(C) = 0$, C és un con amb punta, generat per dos vectors als seus costats: $C = co(v_1, v_2)$.

Semiplà Si $\dim v(C) = 1$, C és un semiplà, generat per tres vectors: $C = (v, -v, u)$, on v és un vector del vèrtex i u és un vector interior del con.

Pla Si $\dim v(C) = 2$, $C = \mathbb{R}^2$, i és fàcil veure que és finitament generat, ja que òbviament $C = \mathrm{co}((1,0), (0,1), (-1,0), (0,-1))$. De fet és possible donar un conjunt de 3 vectors que ja generin $C = \mathbb{R}^2$, per exemple $C = \mathrm{co}((1,0), (0,1), (-1,-1))$.

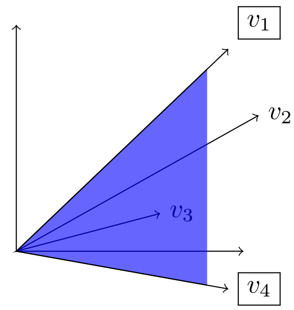


Figura 2: Els vectors v_1 i v_4 són un sistema minimal de generadors de $\mathrm{co}(v_1, v_2, v_3, v_4)$, mentre que v_2, i v_3 són vectors interiors.

Sistema minimal de generadors

A diferència del que passa amb els subespais vectorials, no tots els cons tenen una "base", ni tan sols els finitament generats. No obstant, sí que podem considerar els sistemes de generadors *minimals*:

Definició 6. Un conjunt finit de vectors $S = \{v_1, \ldots, v_n\}$ és un *sistema minimal de generadors* del con C que generen si cap d'ells és combinació cònica dels altres. Equivalentment, cal que compleixi les dues propietats següents:

1. S genera C, és a dir, $\mathrm{co}(S) = C$,
2. tots els vectors de S són necessaris per generar C, és a dir, $\forall i$, $\mathrm{co}\left(S \setminus \{v_i\}\right) \subsetneq C$.

Exemple 7. Considerem els vectors $v_1, \ldots, v_4$ de la figura 2. Els vectors interiors v_2 i v_3 són combinació cònica de v_1 i v_4, i per tant no són necessaris per generar el con blau. El sistema minimal de generadors d'un angle del pla en tant que con està format per dos vectors: un sobre cada costat.

Observem que no sempre tots els sistemes minimals de generadors de C tenen el mateix nombre de vectors, és a dir, no hi ha un lema de Steinitz per a les combinacions còniques com el de les combinacions lineals. Per exemple, el conjunt de vectors $A = \{(1,0), (0,1), (-1,0), (0,-1)\}$ genera $\mathbb{R}^2$ com a con convex, i és un sistema minimal de generadors, perquè cap d'ells és combinació lineal amb coeficients positius dels altres. No obstant, el mateix con $\mathbb{R}^2$ admet com a sistema de generadors el conjunt $B = \{(1,0), (0,1), (-1,-1)\}$, amb només tres vectors. Fixeu-vos que B no és un subconjunt de A perquè $(-1,-1) \notin A$. El que sí que es compleix és:

Lema 8. *Donat un conjunt finit de vectors $S = \{v_1, \ldots, v_n\}$, existeix un subconjunt $M \subset S$ de vectors que formen un sistema minimal de generadors del con* $\mathrm{co}(S)$.

Demostració. Per inducció sobre n. El resultat és obvi si $n = 1$. Suposem $n > 1$. Si cap dels vectors v_i és combinació cònica dels altres $S \setminus \{v_i\}$, llavors S és un sistema minimal de generadors. En l'altre cas existeix un v_i combinació cònica dels $S \setminus \{v_i\}$; llavors $\mathrm{co}(S) = \mathrm{co}\,(S \setminus \{v_i\})$ i per hipòtesi d'inducció hi ha un subconjunt

$$M \subset S \setminus \{v_i\} \subset S$$

que és sistema minimal de generadors de $\mathrm{co}(S) = \mathrm{co}\,(S \setminus \{v_i\})$. □

Algorisme per a cons plans

Suposem que se'ns dona un conjunt $S = \{v_1, \ldots, v_n\}$ de vectors no nuls de $\mathbb{R}^3$. Volem *classificar* el con $C = \mathrm{co}(S)$ generat per aquests vectors (és a dir, determinar si el con es redueix a un punt, o és una semirecta, una recta, o si es tracta d'un dels tres tipus de con bidimensional de l'apartat anterior, o té dimensió 3) i trobar un subconjunt de S que siguin *generadors minimals* de C.

A continuació expliquem un algorisme per fer aquesta tasca per als cons de dimensió ≤ 2.

Suposarem que coneixem un vector u que és perpendicular a tots els generadors $v_1, \ldots, v_n$. Això vol dir que el con està contingut al pla ortogonal $u^\perp$ i per tant té dimensió ≤ 2. En aquest cas tota la informació necessària per resoldre el problema està en els angles

$$\alpha_{ij} = \widehat{v_i, v_j}.$$

Efectivament, veiem a la figura 2 que els dos vectors generadors són els que formen entre ells l'angle màxim, i per tant es poden trobar buscant aquest màxim. En el cas dels altres possibles cons de dimensió ≤ 2, la resposta també es troba en els angles, per exemple en un semiplà dos dels generadors formen un angle de π radians.

Abans d'entrar en detalls de l'algorisme, observem que fixar el vector perpendicular u permet definir el *signe* de l'angle $\alpha_{ij} = \widehat{v_i, v_j}$. Recordeu que el valor absolut d'aquest angle es calcula per la regla del cosinus a partir del producte escalar; considerarem que té el mateix signe que el determinant de la matriu formada pels vectors u, v_i, v_j en aquest ordre (i llavors l'angle α_{ji} tindrà el signe oposat).

Atenció: El signe de l'angle π és indeterminat! Efectivament, si $|\alpha_{ij}| = \pi$, llavors v_i i v_j són linealment dependents i per tant $\det(u, v_i, v_j) = 0$, que no és negatiu ni positiu. Podem establir en aquest cas per conveni que el signe és positiu. D'aquesta manera tindrem sempre $-\pi < \alpha_{ij} \leq \pi$.

A continuació introduirem el pseudocodi corresponent a una funció, anomenada `tipus2d`, que permet classificar el con pla donat. Suposarem ja programades les funcions de la primera secció (producte escalar, norma, etc.)

1. Modifiqueu la funció de càlcul de l'angle entre dos vectors de manera que, donats 3 vectors u, v_1, v_2, on v_1 i v_2 pertanyen al pla perpendicular a u, retorni l'angle amb signe entre v_1 i v_2 d'acord amb la regla anterior.

2. Programeu una funció per calcular el *producte vectorial* de dos vectors a $\mathbb{R}^3$. Recordeu que

$$(x, y, z) \wedge (x', y', z') = (yz' - zy', zx' - xz', xy' - yx').$$

(Aquesta última funció només serà necessària al programa final que s'ocupa de cons tridimensionals).

Anirem comentant l'algorisme de la funció `tipus2d` a mesura que presentem el pseudocodi. En primer lloc fixem els paràmetres d'entrada i la sortida que tindrà aquesta funció, i les variables locals que necessita:

```
Entrada:
v[][] és una llista de generadors,
      codificada com un vector de vectors no nuls (x,y,z).
n és el nombre de generadors.
u és un vector perpendicular a tots els generadors.
vmin[] és una llista d'enters, inicialment buida.
Sortida:
0 si el con generat és el {0}
1 si el con generat és una semirecta
2 si el con generat és una recta
3 si el con generat és un angle del pla
4 si el con generat és un semiplà
5 si el con generat és un pla
vmin[] conté els índexs dels v_i que són generadors minimals,
       en sortir de la funció
Algorisme:  tipus2d(v,n,u,vmin) és
var
i,j:enter
dr,esq:enter
angle[][]:real
anglemax:real
fivar
```

Tal com hem comentat, el primer pas és calcular els angles entre cada parella de vectors. Això es podrà fer sempre i quan hi hagi almenys una parella, és clar! Però si no és així, directament podem dir de quin tipus és el con. Recordem que els vectors han de ser no nuls.

```
si n=0 llavors retorna 0; fisi // Con zero
si n=1 llavors                 // Semirecta
  afegeix 0 a vmin;
  retorna 1;
fisi                           // A partir d'aquí n>1
dr=0;
esq=1;
anglemax=angle entre v[0] i v[1];
```

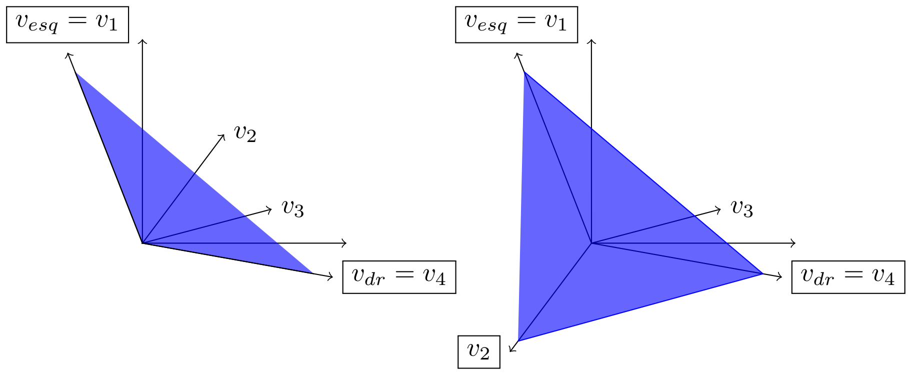


Figura 3: A les dues figures l'angle màxim és el mateix, però el con de la dreta és tot el pla $\mathbb{R}^2$. La diferència la marca que hi ha un angle $\widehat{v_{dr}, v_i}$ amb signe negatiu, en aquest cas $\widehat{v_4, v_2}$. Veiem a més que els vectors v_1, v_2, v_4 són un sistema minimal de generadors.

```
per i=0 fins n-1 fer
  per j=0 fins n-1 fer
    angle[i][j]=angle entre v[i] i v[j];
    si angle[i][j] > anglemax llavors
      dr=i;
      esq=j;
      anglemax=angle[i][j];
    fisi
  fiper
fiper
```

En el pseudocodi hem usat el conveni que els índexs de les components d'un vector comencen a 0, com en `C`. Per això en els bucles la variable `i` o `j` va de 0 a $n-1$.

Arribats en aquest punt, l'angle entre `v[dr]` i `v[esq]` és màxim (i no negatiu). A partir d'aquí podem raonar quin tipus de con és. Serà una semirecta si i només si tots els vectors apunten en la mateixa direcció (l'angle màxim és zero). El cas "més habitual" serà el d'un angle pla, limitat pels vectors `v[dr]` i `v[esq]`, però si hi ha un vector que apunta "enfora" de l'angle limitat per aquests dos, llavors el con és el pla sencer. Per detectar aquest fenomen hem de mirar els signes dels angles (vegeu figura 3)

```
si anglemax=0 llavors // És una semirecta
  afegeix 0 a vmin;
  retorna 1;
fisi
si anglemax < π llavors // És un angle pla o un pla
  per i=0 fins n-1 fer
```

```
    si angle[dr][i] < 0 llavors
      afegeix dr a vmin;
      afegeix esq a vmin;
      afegeix i a vmin; // És un pla!
      retorna 5;
    fisi
  fiper // Si hem arribat aquí, és un angle pla
  afegeix dr a vmin;
  afegeix esq a vmin;
  retorna 3;
fisi
```

Si hem arribat aquí, l'angle màxim és π, i per tant haurem de determinar si es tracta d'una recta, un semiplà, o un pla. Si tots els angles entre parelles de vectors són 0 o π, el con generat és una recta (que necessita dos generadors minimals, `v[dr]` i `v[esq]`). Si no és aquest el cas llavors caldrà distingir, segons si tots els generadors estan o no al mateix costat de `v[dr]` i `v[esq]`, entre el pla i el semipla. Per descobrir-ho caldrà usar de nou els signes dels angles. Acaba d'escriure el pseudocodi tu mateix!

fialgorisme

Programa per a cons plans

Fes un programa que llegeixi del teclat un vector u i uns generadors $v_1, \ldots, v_n$, comprovi si tots ells pertanyen al pla perpendicular a u, i en cas afirmatiu, classifiqui el con generat i en trobi un sistema minimal de generadors seguint el pseudocodi anterior. (Inicialment, haurà de demanar a l'usuari quants generadors té el con, i usar assignació dinàmica de memòria, com a la pràctica 6).

La funció `tipus2d` hauria d'utilitzar les funcions programades anteriorment per calcular angles, normes, determinants, etc.

Opcionalment pots fer que el programa demani a l'usuari només els generadors $v_1, \ldots, v_n$ i calculi mitjançant el producte vectorial un vector u perpendicular a tots ells (fixa't que això no funcionaria si la dimensió del con és 1, per tant el programa hauria de fer les comprovacions pertinents, possiblement calculant primer els angles sense signe).

Nota: programar llistes i "afegir"

Una llista és un vector la longitud del qual varia. Afegir un element a la llista és incrementar la longitud i copiar l'element a l'última posició del vector.

En el cas del pseudocodi per classificar cons, s'utilitza una llista d'enters. En el llenguatge `C` no hi ha un tipus de variable que sigui una llista. Quan implementis l'algorisme en el teu programa, et suggerim que declaris la llista com *una parella* de variables: el vector d'enters que contindrà la llista pròpiament (que serà "en realitat" de longitud fixada) i una variable entera que contindrà la longitud actual de la llista (com que la llista inicialment està buida, el valor inicial d'aquesta variable serà zero). Perquè al vector hi càpiga la llista sencera final, haurà d'estar declarat de la mida màxima que pugui possiblement ser necessària. Per a un con pla, la màxima mida d'un sistema de generadors *minimal* és de quatre vectors.

Així doncs, la llista `vmin` podria estar declarada en el programa com

```
unsigned int vmin[4];
unsigned int long_vmin=0;
```

on `long_vmin` és la variable que contindrà la longitud de la llista. Llavors el pseudocodi **`afegeix`** `i` **`a`** `vmin;` es podria programar com un parell d'instruccions

```
vmin[long_vmin]=i;
long_vmin++;
```

Fixeu-vos que a la funció `tipus2d`, la llista `vmin` conté a la sortida els índexs dels generadors minimals, és a dir, que s'ha de passar *per adreça* (apuntador) perquè la funció principal pugui "veure" el resultat. Això és automàtic en `C` per als vectors, però en aquest cas també s'ha d'aplicar a la longitud, que està en una variable entera `long_vmin`. Tingueu-ho en compte!

A la última part del treball caldrà fer un ús més intens de llistes, i en particular necessitarem *buidar* una llista. Si l'heu programat seguint aquests suggeriments, buidar la llista és simplement fer `long_vmin=0;`

A la pràctica 8 veurem com combinar dues o més variables en una `struct`. Si ho vols, les dues variables que formen la *llista* (és a dir, el vector `vmin` pròpiament dit i la seva longitud `long_vmin`) es poden combinar en una `struct llista`. *Aquesta modificació és opcional.*

3 Algorisme en dimensió 3

Classificació teòrica en dimensió 3

Tal com ho vam fer per als cons plans, els de dimensió 3 es poden classificar segons la dimensió del vèrtex:

Con polièdric (també anomenat **angle sòlid**) Si $\dim v(C) = 0$, C és un con amb punta, generat per vectors situats cadascun sobre una aresta: $C = \mathrm{co}(v_1, v_2, \ldots, v_n)$. El nombre mínim de generadors pot ser qualsevol $n \geq 3$.

Angle dièdric sòlid Si $\dim v(C) = 1$, C té la forma d'un tascó (falca, cuny) infinit. Es pot generar amb quatre vectors: dos sobre la recta vèrtex i dos sobre les cares planes, $C = \mathrm{co}(v_1, -v_1, v_2, v_3)$.

Semiespai Si $\dim v(C) = 2$, C és un semiespai, que es pot generar amb quatre vectors: $C = (v_1, v_2, -v_1 - v_2, v_3)$, on v_1, v_2 són vectors linealment independents del pla vèrtex i v_3 és un vector interior del con.

Espai Si $\dim v(C) = 3$, $C = \mathbb{R}^3$ es pot generar amb quatre vectors, per exemple $C = \mathrm{co}((1,0,0), (0,1,0), (0,0,1), (-1,-1,-1))$.

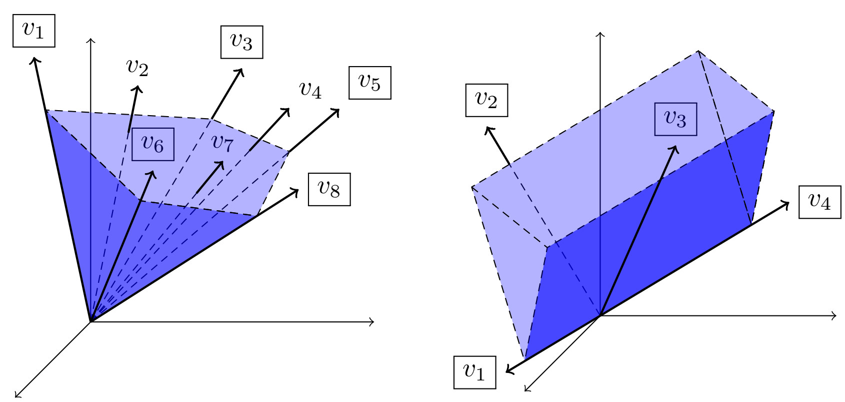


Figura 4: A la imatge de l'esquerra hi ha representat un con polièdric generat per 8 vectors, $\mathrm{co}(v_1, v_2 \ldots, v_8)$. Els vectors "interiors" v_2, v_4 i v_7 són combinació cònica de v_1, v_3, v_5, v_6 i v_8, i per tant no són necessaris per generar el con. Els vectors v_1, v_3, v_5, v_6 i v_8 (un sobre cada aresta) són un sistema minimal de generadors. A la dreta hi ha representat un angle dièdric sòlid generat per quatre vectors $v_1, \ldots, v_4$ amb $v_4 = -2\,v_1$ (són un sistema minimal de generadors).

L'algorisme que proposem per a classificar els cons de l'espai es basa en fer projeccions ortogonals. Es justifica pels lemes següents:

Lema 9. *Siguin $S = \{v_1, \ldots, v_n\}$ un conjunt de vectors de $\mathbb{R}^3$ i $u \in \mathbb{R}^3$ un vector no nul. Anomenem $\varphi_u : \mathbb{R}^3 \to u^\perp$ la projecció ortogonal sobre el pla*

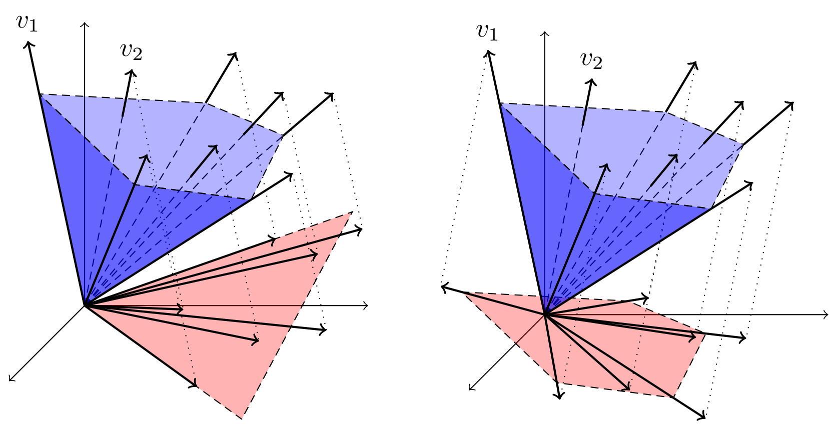


Figura 5: Dues projeccions ortogonals del con poliedral de la figura 4. Els vectors de projecció són v_1 i v_2 respectivament. Com que v_1 està sobre una aresta, la primera projecció és un angle del pla ortogonal. En canvi, v_2 és un vector interior i la projecció corresponent és tot el pla ortogonal a v_2.

perpendicular a u. Llavors la imatge per φ_u del con $\mathrm{co}(S)$ generat per S és un con; més precisament, és el con $\mathrm{co}(S')$ generat per $S' = \{\varphi_u(v_1), \ldots, \varphi_u(v_n)\}$.

Lema 10. *En la mateixa situació del lema anterior, si $u \in \mathrm{co}(S)$ llavors $\dim \mathrm{co}(S') = \dim \mathrm{co}(S) - 1$. A més, en aquest cas:*

1. *$\mathrm{co}(S')$ és un pla si i només si u es troba a l'interior de $\mathrm{co}(S)$.*
2. *$\mathrm{co}(S')$ és un semiplà si i només si u es troba sobre una cara, però no sobre una aresta, de $\mathrm{co}(S)$.*
3. *$\mathrm{co}(S')$ és un angle del pla si i només si u es troba sobre una aresta de $\mathrm{co}(S)$.*

Per simplicitat no incloem les demostracions d'aquests lemes.

- Escriu una funció `projectacon(n,u,v,proj)` que, donada una llista de n vectors no nuls v_i de $\mathbb{R}^3$ i un vector no nul u, calculi la llista dels vectors no nuls $proj_i = \varphi_u(v_i)$ perpendiculars a u. En el cas que algun dels vectors projectats sigui nul (vol dir que v_i és un múltiple de u) cal treure'l de la llista. La funció hauria de retornar un enter: el nombre de vectors amb projecció no nul·la.

Algorisme general

Suposem que se'ns dona un conjunt $S = \{v_1, \ldots, v_n\}$ de vectors no nuls de $\mathbb{R}^3$. Volem *classificar* el con $C = \mathrm{co}(S)$ generat per aquests vectors (és a dir, determinar si el con és un con polièdric, un angle dièdric, un semiespai, l'espai sencer, o un dels cons de dimensió ≤ 2 estudiats anteriorment) i trobar un subconjunt de S que siguin *generadors minimals* de C.

A continuació presentem un algorisme per fer aquesta tasca, en forma de pseudocodi. Es basa en un argument de reducció en la dimensió. Es consideren les projeccions planes del con C ortogonals a cada generador (que són cons plans calculats segons la funció `projectacon` que has escrit abans) i es resol el problema per als cons projectats. A partir d'aquestes solucions es mostra com resoldre el problema original.

```
Entrada:
v[][] és una llista de generadors,
      codificada com un vector de vectors no nuls (x,y,z).
n és el nombre de generadors.
vmin[] és una llista d'enters, inicialment buida.
Sortida:
0 si el con generat és el {0}
1 si el con generat és una semirecta
2 si el con generat és una recta
3 si el con generat és un angle pla
4 si el con generat és un semiplà
5 si el con generat és un pla
6 si el con generat és polièdric
7 si el con generat és un angle dièdric
8 si el con generat és un semiespai
9 si el con generat és tot l'espai
vmin[] conté els índexs dels v_i que són generadors minimals,
       en sortir de la funció
Algorisme:  tipus3d(v,n,vmin) és
var
i,j,m,tip:enter
repetit:enter
w[][]:real      // Llista de m ≤ n vectors
wmin[]:enter    // Llista de ≤ 4 enters
vcares[]:enter  // Llista de ≤ n-1 enters
perp[]:real
fivar
```

El primer pas és separar els casos de dimensió ≤ 2. Com en la funció `tipus2d`, si només hi ha un generador, la resposta és immediata. També pot passar que sigui un con de dimensió 1 (recta o semirecta) i això es detectarà perquè projectant ortogonalment respecte d'un generador qualsevol (per exemple, el primer) tots els generadors projectats s'anul·len.

```
si n=0 llavors retorna 0; fisi // Con zero
si n=1 llavors                 // Semirecta
  afegeix 0 a vmin;
```

```
  retorna 1;
fisi                    // A partir d'aquí n>1
m=projectacon(n,v[0],v,w);
si m=0 llavors          // Projecció igual a punt:  dimensió 1
  per i=0 fins n-1 fer  // Cal distingir si recta o semirecta
    si producte_escalar(v[0],v[i]) < 0 llavors
      afegeix 0 a vmin; // Algun angle π:  recta
      afegeix i a vmin;
      retorna 2;
    fisi
  fiper
  afegeix 0 a vmin;     // Cap angle π:  semirecta
  retorna 1;
fisi
```

La següent possibilitat és que el con sigui de dimensió 2. Això ho detectarem perquè una projecció té dimensió 1, i llavors podrem usar la funció `tipus2d` per resoldre el problema. Però ens cal trobar un vector perpendicular al con per usar `tipus2d`. Aquest el trobem com el producte vectorial del vector de projecció per qualsevol vector projectat que sigui no nul.

```
tip=tipus2d(w,m,v[0],wmin);
si tip<3 llavors
  perp=producte_vectorial(v[0],w[0]); // Vector perpendicular
  retorna tipus2d(v,n,perp,vmin);
fisi // A partir d'aquí:  és con de dimensió 3
```

Fixeu-vos que `w[0]` no és la projecció de `v[0]`, sinó la projecció del primer vector independent de `v[0]`, ja que els vectors amb projecció nul·la s'han tret de la llista `w`.

Per classificar els cons de dimensió 3 el que farem és buscar les arestes i els generadors que es troben sobre una cara, en cas que n'hi hagi algun. Calcularem les projeccions ortogonals respecte de cada generador; pel lema 10, el generador v_i està sobre una aresta exactament quan la projecció $\phi_{v_i}(co(S))$ és un angle del pla. Tots aquests generadors els incorporarem a la llista de generadors minimals, tenint la precaució de comprovar que no n'hi hagi de repetits (és a dir, si dos generadors estan sobre la mateixa aresta, cosa que es pot comprovar perquè l'angle que formen és nul, només en guardarem un). Independentment del tipus de con que sigui, són vectors que han d'aparèixer en tot sistema minimal de generadors. D'altra banda, farem una llista amb els vectors que es troben en una cara, que ens faran falta si el con és dièdric o un semiespai.

En aquest pseudocodi usarem `long(llista)` per denotar la longitud d'una llista. Recordem que en la implementació en `C` aquesta longitud es guardarà en una variable destinada a això.

```
per i=0 fins n-1 fer
  m=projectacon(n,v[i],v,w);
  tip=tipus2d(w,m,v[i],wmin);
```

```
  si tip=3 llavors   // projecció és angle; v[i] genera aresta
    repetit=0;       // control d'aresta ja trobada
    per j=0 fins long(vmin)-1 fer
      si angle(v[i],v[vmin[j]])=0 llavors repetit=1; fisi
    fiper
    si repetit=0 llavors afegeix i a vmin; fisi
  fisi
  si tip=4 llavors   // projecció és semiplà; v[i] està a cara
    afegeix i a vcares; fisi
fiper                // En aquest punt vmin generen les arestes
```

Un cop feta la llista, si hi ha més de dos vectors generant arestes diferents, significarà que el con és polièdric i ja tenim els generadors minimals. Si n'hi ha exactament dos (hauran d'estar sobre la mateixa recta i en sentit oposat) es tractarà d'un con dièdric. En aquest cas cal afegir als vectors aresta dos vectors més, cadascun en una cara diferent. Tenim els índexs dels vectors de cares a la llista `vcares`, i volem distingir quan es troben en cares diferents. Això correspon a què les seves projeccions ortogonals respecte de l'aresta formin un angle no nul.

```
si long(vmin)>2 llavors retorna 6; fisi  // con polièdric
si long(vmin)=2 llavors      // con dièdric; cal trobar dues cares
  perp=v[vmin[0]];           // vector aresta, per projectar
  afegeix vcares[0] a vmin;           // primera aresta-cara
  w[0]=proj(perp,v[vcares[0]]);       // projecció aresta-cara
  per i=1 fins long(vcares)-1 fer
    w[1]=proj(perp,v[vcares[i]]);     // projecció
    si angle(w[0],w[1])!=0 llavors    // és a l'altra cara
      afegeix vcares[i] a vmin;       // segona aresta-cara
      retorna 7;
    fisi
  fiper
fisi
```

Aquí hem denotat `proj` la funció demanada al punt 7 de la pàgina 3, és a dir, `proj(perp,v[vcares[i]])` serà la projecció ortogonal a perp del i-èsim vector cara: ϕ_{perp}(v[vcares[i]]).

Arribats en aquest punt, només ens queda l'opció que no hi hagi cap aresta, el que pot correspondre tant a un semiespai com a l'espai sencer. Distingirem els dos casos segons si hi ha o no vectors cara.

```
si long(vcares)>0 llavors // És un semiespai
  buida_llista w;         // Col·leccionem els vectors cara
  j=n-1;                    // i també un vector interior
  per i=0 fins long(vcares)-1 fer
    afegeix v[vcares[i]] a w;
    si 0 < |angle(w[0],w[i])| < π llavors
      perp=producte_vectorial(w[0],w[i]);
```

```
    fisi                  // Un vector ortogonal
    si j=n-1 i vcares[i]>i llavors
      j=i                 // Generador interior al con
    fisi
  fiper
  tip=tipus2d(w,long(w),perp,wmin); // Gen. minimals pla
  per i=0 fins long(wmin) fer       // wmin inclou vmin, però
    afegeix vcares[wmin[i]] a vmin; // reindexats respecte v
  fiper
  afegeix j a vmin;
retorna 8 fisi
```

Un petit comentari sobre com trobar un vector interior: es tracta simplement de trobar un índex `j` que no està a la llista `vcares`. El que s'ha fet en aquest pseudocodi és escollir com a `j` el mínim `i` tal que `vcares[i]>i`. Per exemple, si `vcares={0,1,3,4,7}`, el primer `i` tal que `vcares[i]>i` és 2.

Si el programa arriba en aquest punt sense haver trobat el sistema minimal de generadors, sabem que el con és tot $\mathbb{R}^3$. Ens falta escollir un sistema minimal de generadors, i ho farem recursivament, mirant quins d'ells es poden treure. Utilitzarem la llista `w` per posar-hi temporalment tots els vectors generadors menys un (que anirà variant de v_1 a v_n) i comprovar si segueixen generant tot $\mathbb{R}^3$. En cas que per algun v_i sigui així, la pròpia funció `tipus3d` cridada recursivament ens en donarà un sistema de generadors minimal. Per contra, si per a cada vector de la llista, treure'l ens dona un subcon que no és tot $\mathbb{R}^3$, vol dir que el sistema de generadors donat ja era minimal.

```
buida_llista w;
per i=1 fins n-1 fer  // Tots menys v[0]
  afegeix v[i] a w;
fiper
i=0;
mentre tipus3d(w,n-1,vmin) < 9 fer  // No es pot treure v[i]
  w[i]=v[i]; // Posar-lo, i ara no incloure v[i+1]
  i=i+1;
  si i=n llavors // No se'n pot treure cap!
    afegeix n-1 a vmin;
    retorna 9;
  fisi
  buida_llista vmin; // Preparat per cridar tipus3d
fimentre
per j=i fins long(vmin) fer // Com que v[i] falta a w, corregir
  vmin[j]=vmin[j]+1         // els índexs trobats recursivament
fiper
retorna 9; // vmin s'ha calculat recursivament
fialgorisme
```

Les línies en groc faltaven a la primera versió d'aquest document.

Programa a realitzar

Has d'entregar al Campus Virtual un fitxer anomenat amb el teu nom en el format `CognomNom.c`, que es pugui compilar mitjançant la instrucció

```
gcc -Wall -std=c99 -pedantic -o CognomNom CognomNom.c -lm
```

El programa haurà de llegir els generadors $v_1, \ldots, v_n$ d'un fitxer. El nom del fitxer es pot preguntar per pantalla amb `scanf` o pot ser donat com un argument `argv[1]` (com a l'exercici 7.1). El fitxer tindrà tantes línies com generadors, i cada línia constarà de les tres coordenades d'un generador separades per comes, sense parèntesis.

Cal que el programa determini la mida de la llista de generadors (matriu $n \times 3$) en funció del nombre de línies del fitxer d'entrada. És recomanable comprovar que els vectors siguin no nuls.

Mitjançant la funció `tipus3d` que hem descrit, el programa haurà de decidir de quin tipus és el con que generen els vectors del fitxer d'entrada, i imprimir per pantalla la resposta així com els índexs dels vectors que formen un sistema minimal de generadors. En un altre fitxer haurà d'escriure-hi els vectors generadors minimals.

Exemple 11. Suposem que el fitxer "entrada.dat" conté les quatre línies següents:

```
 1.0, 1.0, 1.0
-1.0, 0.0, 0.0
 0.0,-1.0, 0.0
 0.0, 0.0, 1.0
```

Llavors, escrivint `./CognomNom entrada.dat sortida.dat`, el programa hauria d'imprimir a la pantalla

```
S'ha llegit un con generat per 4 vectors
És un con polièdric amb 3 arestes
Un sistema minimal de generadors és v_1, v_2, v_3
```

i hauria de generar un fitxer anomenat `sortida.dat` format per les 3 primeres línies d'`entrada.dat`.

Notes finals

Sobre l'avaluació del treball

El programa final a entregar abans de finals de maig tindrà una base formada per les funcions bàsiques (secció 1) i les que implementen l'algorisme de càlcul i classificació de cons plans (secció 2), completat amb l'algorisme de càlcul i classificació de cons tridimensionals de la secció 3. És admissible entregar un programa que només tracti els cons de dimensió 2 (seccions 1 i 2). En aquest cas la nota màxima que es podrà assolir és un 6,9. Per aspirar a un Notable o Excel·lent cal implementar tots els algorismes.

De manera semblant, qualsevol programa que no compleixi tots els requisits demanats tindrà penalització (per exemple, un programa que implementi tot l'algorisme tal com s'ha descrit però no compleixi el requisit de llegir les dades d'un fitxer).

Un programa que no implementi correctament l'algorisme també tindrà penalització. Podreu trobar al Campus Virtual fitxers d'entrada de mostra per comprovar el funcionament del vostre programa.

Un programa que generi *warnings* en compilar-se amb la instrucció

```
gcc -Wall -std=c99 -pedantic -o CognomNom CognomNom.c -lm
```

també tindrà penalització. Un programa que tingui errors sintàctics i no generi un executable serà suspès.

Sobre l'algorisme i el pseudocodi

El programa entregat ha de fer la feina de classificiació de cons i determinació de sistemes minimals de generadors seguint l'algorisme indicat. Inventar (o descobrir a internet) un algorisme diferent per fer la mateixa feina pot ser molt interessant, però no és el que es demana, i serà penalitzat.

Ara bé, el pseudocodi que hem inclòs aquí és només indicatiu, per a explicar l'algorisme, i no és necessari que el programa el segueixi al peu de la lletra. Per exemple, després de decidir que un con té dimensió 3 (havent-lo projectat des de `v[0]`) es podria aprofitar el contingut de les variables `m` i `tip`. O es poden endreçar els càlculs de manera diferent: per exemple, es podria començar calculant la llista de tots els vectors-aresta i vectors-cara abans de determinar si el con és de dimensió 1 o 2. Igualment, en el cas dels cons plans, es podria començar calculant els angles sense signe, i buscar els signes només a posteriori si l'angle màxim no és zero ni pi. O si es troba un angle igual a pi, ja no cal calcular-ne més.

L'algorisme proposat és poc eficient en el cas que el con sigui tot l'espai $\mathbb{R}$ (perquè funciona recursivament). Això és força habitual: un algorisme pot estar optimitzat per a la situació que es preveu més freqüent, i no ser tan eficaç en casos especials. També en aquest cas us demanem que programeu l'algorisme proposat (recursiu) encara que es permeten canvis en la manera concreta d'implementar-lo (no cal que sigui traducció directa del pseudocodi).

En aquest programa, estem suposant que “habitualment” el con donat no serà tot l’espai perquè no és un con gaire interessant. Això sí, un bon programa ha de donar una resposta correcta en tots els casos.