[Butll. Soc. Cat. Cièn. Vol. III, 127-146 (1984)]

UN PROGRAMA PER A CALCULAR L'HOMOLOGIA SIMPLICIAL

per

JORDI SALUDES I CLOSA

De la Secció de Matemàtiques de la SCCFQM.

1. INTRODUCCIÓ

Aquest programa ens permet de calcular els grups d'homologia d'un complex simplicial finit de dimensió m; al qual hom adjunta –si volem– una cel·la de dimensió k, segons una aplicació simplicial donada, entre una triangulació de S^{k-1} i el complex.

Sigui K un complex simplicial i anomenem els seus vèrtexs amb nombres naturals diferents. Considerem a K l'orientació α que s'obté a cada símplex en recórrer els seus vèrtexs de manera creixent. Direm K^{α} a K amb aquesta orientació. Així podem referir-nos al complex, donant una llista de tots els seus símplexs com i-uples ($0 \leqslant i \leqslant n$).

A partir d'aquesta llista el programa calcularà els grups d'homologia (des de n fins a 1) del complex que li hem donat.

L'adjunció de cel·les la tractarem més endavant.

2. CÀLCUL DE LES MATRIUS D'INCIDÈNCIA

Si s^r és un r-símplex de K^{α}, la seva vora es fa:

$$s^r = \sum_k [s^r, s^{r-1}]^{\alpha} s^{r-1}, \quad \text{on } s_k^{r-1}$$

recorre els símplexs de dimensió $r - 1$ i $[s^r, s_{k}^{r-1}]$ és el nombre d'incidència hom es defineix com zero si s_k^{r-1} no és cara de s^r; $+ 1$ si s_k^{r-1} és cara de s^r i a més l'orientació α coincideix a s_k^{r-1} amb la induïda per s^r que és:

$$s_k^{r-1} = (-1)^i < a_o, \ldots, a_{i-1}, a_{i+1}, \ldots, a_r >$$

on $s^r = < a_o, \ldots, a_r >$ (s_k^{r-1} s'obté de s^r suprimint-hi un vèrtex); i si α no coincideix amb la induïda $[s^r, s_k^{r-1}]^\alpha = -1$. Com que α és donat pels vèrtexs creixents, és trivial que:

$$[s^r, s_k^{r-1}] = \begin{cases} (-1)^i & \text{Si } s_k^{r-1} = < a_o, \ldots, a_{i-1}, a_{i+1}, \ldots, a_r > \\ 0 & \text{Si } s_k^{s-1} \text{ no és cara de } s^r \end{cases}$$

El càlcul dels nombres d'incidència és, doncs, prou senzill.

$C_r(K^\alpha)$ és un grup abelià lliure amb base $\{s_i^{r-1}\}$. La matriu en aquestes bases de l'aplicació vora:

$$\partial : C_r(K^\alpha) \longrightarrow C_{r-1}(K^\alpha)$$

és la formada pels nombres d'incidència $M_r = ([s_i^r, s_k^{r-1}])$. El programa elabora aquestes matrius generant les cares de cada r-símplex, buscant-les al $(r-1)$-esquelet i posant $+1$ o -1 segons es tracti.

Per tal de calcular l'homologia de dimensió r, necessitem les matrius d'incidència M_{r+1} i M_r corresponents a:

$$C_{r+1}(K^\alpha) \xrightarrow{\partial_{r+1}} C_r(K^\alpha) \xrightarrow{\partial_r} C_{r-1}(K^\alpha)$$

3. CICLES, VORES I EL PAS AL QUOCIENT

Els cicles de $C_r(K)$ són exactament ker ∂_r. Per tant el primer problema és trobar una base del nucli de M_r. Per a fer tot això aprofitem unes subrutines de diagonalització entera de matrius.

Sabem que tota matriu rectangular entera A es pot diagonalitzar (possiblement amb zeros quan s'acaba la diagonal): D = U.A.V. on U i V són les matrius dels canvis de base (inversibles enteres). El mètode és com segueix:

Suposem $a_{11} \neq 0$. (Si $a_{11} = 0$ fent un canvi de files i columnes podem suposar-ho diferent de zero). Si $a_{1i} \neq 0$, calculem d = m.c.d. (a_{11}, a_{1i}) mitjançant l'algorisme d'Euclides. Per la identitat de Bezout: $d = r \cdot a_{11} + s \cdot a_{1i}$; d'altra banda $a_{11} = r_p \cdot d$ $a_{1i} = s_p \cdot d$. Ara ja és clar que:

$$\begin{pmatrix} r & s_p \\ s & -r_p \end{pmatrix},$$

aplicat a les columnes: primera i i-èssima, posa d al lloc (1,1) i zero a (1, i). Aquest procediment aplicat per columnes i després per files als $a_{i1} \neq 0$ permet de passar d'A = $A^{(0)}$ a $A^{(1)}$ tal que

$$a_{1i}^{(1)} = 0 \text{ si } i \neq 1 \text{ i } a_{j1} = 0 \quad j \neq 1.$$

Continuant el procés ara per a la submatriu $A^{(1)}_{i \geqslant 2,\, j \geqslant 2}$ es pot anar diagonalitzant fins que tots els elements que resten siguin zero.

Els canvis que afecten columnes s'enregistren aplicant-los a V que a A (V i U comencen essent la identitat). Els que afecten files s'enregistren a U.

Tornant al nostre problema: x és de ker M_r sii $M_r\,x = 0$ $D = U \cdot M_r \cdot V$, fent el canvi x = Vy i sabent que U és un canvi de base, per tant isomorfisme: $M_r\,x = 0$ sii $Dy = 0$; però per la construcció de la matriu diagonal, això només passa quan y té les últimes components diferents de zero (els zeros de la diagonal es troben al final). Per tant $\{e_i\}$ $i \geqslant p$ formen base dels y, ($e_i = (0, 0, \ldots, 1, 0, \ldots, 0)$), $x_i = V\,e_i$ és la i-èssima columna de V. O sigui: la base dels cicles és donada per les darreres columnes de V.

Les vores de dimensió r estan generades per les columnes de M_{r+1}. Caldrà tenir-les expressades en combinació lineal de la base dels cicles que hem trobat. Sigui ara V la matriu que per columnes té els cicles. Si y és una vora, s'ha de trobar x tal que y = Vx. Si diagonalitzem V: L = U.V.W. tenim: $U'.\, y = L\, z$, on $x = W\, z$. La inversa de la diagonal L és fàcil de calcular; per tant: $L^{-1}.\, U'y = z$, i ara $x = W \cdot L^{-1} \cdot U'y$. Notem que el producte $U'y$ per y vora ens el podem estalviar, si acumulem els canvis de files sobre la matriu M_{r+1}. Així només hem d'anar calculant WL^{-1} sobre cada columna de la M'_{r+1} modificada d'aquesta manera.

Amb aquests x anirem formant una matriu F_r amb les expressions d'un conjunt generador de les vores, a la base dels cicles.

Observem que aquest mètode permet d'obtenir explícitament, si cal, les expressions dels cicles i de les vores en aquesta base. Per a tal cosa només cal suprimir en compilar les "D" de les línies de depuració del programa (que són les que comencen amb aquesta lletra).

Tenim F_r que dóna l'expressió del grup abelià lliure de les vores, dins el grup abelià lliure dels cicles. Pel que sabem de classificació de grups abelians finitament generats, el quocient cicles/ vores (és a dir: *l'homologia*) s'obté diagonalitzant F_r. Ara només cal tenir en compte que a cada zero de la diagonal (o de la darrera columna quan la diagonal s'acaba —recordem que F_r és rectangular—) li correspon un sumand directe Z. És a dir: el rang és el nombre de zeros. Cada coeficient diferent de zero i u, representa un sumand de torsió Z/nZ; i els uns representen sumands trivials que no cal considerar.

Aquest procés és el que se segueix per tal de determinar $H_r(K)$. El programa encadena els esquelets $r+1$, r, $r-1$ de manera que llegint l'esquelet $r-2$ (de la llista inicial que l'hi hem donat) –que passarà a ocupar el lloc del $r-1$, aquest es desplaçarà al r, i aquest darrer al lloc del $r+1$– hom pot calcular $H_{r-1}(K)$ i així anar fent fins a $H_1(M)$.

4. L'ADJUNCIO DE CEL·LES

Suposem ara que tenim una aplicació simplicial f d'una triangulació de l'esfera S^{r-1} ($r \geqslant 2$) al complex K. El complex K_f que s'obté adjuntant una r-cel·la E^r mitjançant f, el podem suposar: $K_f = K \cup \overline{E}^r$ (una triangulació de E^r amb vora la triangulació de S^{r-1}). $/ K \cap \overline{E}^r / \cong S^{r-1}$. Aplicant la successió exacta de Mayer-Vietoris al complex K_f i als subcomplexos K i E^r:

$$0 \to H_r(K) \to H_r(K_f) \xrightarrow{j} H_{r-1}(S^{r-1}) \xrightarrow{f_*} H_{r-1}(K) \to H_{r-1}(K_f) \to 0$$

Com que $H_s(S^k) = (0)$ si $k \neq s$ i $H_r(E^k) = (0)$ si $r \neq 0$, $H_{r-1}(S^{r-1}) = Z$. Suposem que aquest Z és generat per γ. De l'exactitud de la successió es desprèn que $H_{r-1}(K_f) \cong H_{r-1}(K)/\mathrm{Im}\, f_*$ i, de

$$H_r(K) \to H_r(K_f) \to Z,$$

que la imatge només pot ésser zero o isomorfa als enters. En el cas $\mathrm{Im}\, j = (0) \Leftrightarrow \mathrm{Ker}\, f_* = (0)$ tenim:

$$H_r(K) \cong H_r(K_f);$$

en cas que $\mathrm{Im}\, j \cong Z \Leftrightarrow \mathrm{Ker}\, f_* \neq (0)$ (es a dir: $f_*(n\gamma) = 0$ per alguna n), tenim:

$$0 \to H_r(K) \to H_r(K_f) \xrightarrow{j} Z \to 0,$$

i com que Z és lliure, escindeix en suma directa: $H_r(K_f) \cong H_r(K) \oplus Z$.

O sigui: en adjuntar una cel·la poden sortir més $(r-1)$-vores i més r-cicles.

Per tal que el programa tingui en compte l'adjunció, haurem de donar una llista dels símplexs que formen la triangulació de S^{r-1}. Ara bé: els nombres associats als vèrtexs d'aquesta triangulació seran els corresponents als vèrtexs imatge dins K; d'aquesta manera es descriu l'estructura de f. Els símplexs es donaran, no en l'orientació de vèrtexs creixents, sinó en la que és induïda per una orientació global β de S^{r-1}.

L'objectiu d'això és: Si $\{s_i^\beta\}$ són els símplexs de S^{r-1} amb l'orientació induïda per β tindrem que $\sum_i s_i^\beta$ serà un cicle; és a dir, podrem prendre

$$\gamma = \sum_i s_i^\beta .$$

Ara és clar que quan donem una tal llista de símplexs, com hem dit, estem donant $f_\#(\gamma)$;

$$f_\# : C_{r-1}(S^{r-1}) \longrightarrow C_{r-1}(K^\alpha)$$

per tal de trobar l'expressió de $f_\#(\gamma)$ en la base de les cadenes de $C_{r-1}(K^\alpha)$, hem de calcular els nombres d'incidència

$$[f_\#(s_i^\beta), s_j'^{\alpha}]$$

—on $s_j'^{\alpha}$ són els $(r-1)$-símplexs de K^α donats per la llista inicial que descriu K—, car tenim:

$$f_\#(\gamma) = \Sigma\, [f_\#(s_i^\beta), s_j'^{\alpha}]\, s_j'^{\alpha} .$$

(Val a dir que la cel·la ha d'ésser no *"degenerada"*, en el sentit que l'aplicació simplicial ha de fer anar $(r-1)$-símplexs a $(r-1)$-símplexs). Clarament $[f(s_i^\beta), s_j'^{\alpha}]$ és zero si s_i^β no s'aplica sobre $s_j'^{\alpha}$; i si s_i^β s'aplica sobre $s_j'^{\alpha}$ serà el signe de la permutació

$$\sigma\, s_i^\beta = (a_1, a_2, \ldots, a_{r-1})\, s_j'^{\alpha} = (a_{\sigma(1)}, a_{\sigma(2)}, \ldots, a_{\sigma(r-1)})$$

que es pot calcular com el signe de:

$$\epsilon = \prod_{j > i}^{r-1} (a_j - a_i)$$

per tal com, per l'orientació $\alpha\ (a_1\ a_2\ a_3 \ldots a_{r-1})$, tenim $\epsilon = 1$.

En aquest moment tenim $f_\#(\gamma)$ expressat en la base de les cadenes de K^α. Haig de calcular:

$$H_{r-1}(K_f) = H_{r-1}(K)/\mathrm{Im}\, f_*$$

on

$$H_{r-1}(K) = Z_{r-1}(K)/V_{r-1}(K) \text{ (Z cicles, V vores).}$$

Fixem-nos que f_*:

$$H_{r-1}(S^{r-1}) \longrightarrow H_{r-1}(K)$$

és la induïda a l'homologia per $f_\#$:

$$Z_{r-1}(S^{r-1}) \longrightarrow Z_{r-1}(K).$$

Així que

$$\operatorname{Im} f_* = \operatorname{Im} f_\# \,/\, (V_{r-1}(K) \cap \operatorname{Im} f_\#).$$

Pels teoremes d'isomorfia tenim:

$$\frac{\operatorname{Im} f_\#}{V_{r-1}(K) \cap \operatorname{Im} f_\#} \cong \frac{\operatorname{Im} f_\# + V_{r-1}(K)}{V_{r-1}(K)}$$

$$\frac{H_{r-1}(K)}{\operatorname{Im} f_*} \cong \frac{Z_{r-1}(K)/V_{r-1}(K)}{\operatorname{Im} f_\# + V_{r-1}(K) \,/\, V_{r-1}(K)} \cong \frac{Z_{r-1}(K)}{\operatorname{Im} f_\# + V_{r-1}(K)}$$

Això ens diu que podem tractar $f_\# \gamma$ com si fos una vora més i per tant podem afegir a M_r la columna de l'expressió de $f_\# \gamma$ i seguir el mateix algorisme d'abans: l'homologia que obtindrem serà la $H_{r-1}(K_f)$. (Aquesta matriu modificada l'anomenarem $\widetilde{M}_r$).

Ara resta per determinar $H_r(K_f)$. Hem vist que es redueix a saber si f_* és injectiva. Si no ho és, existeix n tal que $f_*(n\gamma) = 0$, és a dir

$$f_\#(n\gamma) = n\, f_\#(\gamma) \in V_{r-1}(K)$$

aleshores rang M_r = rang $\widetilde{M}_r$; perquè $nf_\# \gamma$ és combinació lineal de les columnes de M_r (generen $V_{r-1}(K)$). Tot es redueix, doncs, a calcular aquests rangs, però tant li fa fer-ho per M_r i $\widetilde{M}_r$ com per F_r i $\widetilde{F}_r$ (perquè només es diferencien en un canvi de base: de les cadenes als cicles). I rang F_r (o respectivament rang $\widetilde{F}_r$) és únicament comptar els elements diferents de zero de la diagonal. En adjuntar una cel·la no tenim F_r sinó directament $\widetilde{F}_r$, perquè hem comptat $f_\#(\gamma)$ com si fos una altra vora. De tota manera com que adjuntem una sola cel·la, podem obtenir aquest rang, si en el pas anterior —quan calculem l'homologia r— prenem nota del rang de M_r (per calcular aquesta homologia partim de les matrius M_{r+1} i M_r).

El procediment, a grans trets, serà així: Calculem l'homologia $H_r(K_f)$ com is fos $H_r(K)$ –amb les matrius M_{r+1} i M_r–; i quan dono $H_r(K)$ puc calcular ensems rang F_r.

Calculem ara $H_{r-1}(K_f)$ com si fos $H_{r-1}(K)$ però amb $f_{\#}(\gamma)$ com una vora més: amb les matrius $\widetilde{M}_r$ i M_{r-1}, dono $H_{r-1}(K_f)$ i calculo rang $\widetilde{F}_r$.

Si els rangs de F_r i $\widetilde{F}_r$ coincideixen:

$$\operatorname{Ker} f_* \neq (0) \text{ i } K_r(K_f) = H_r(K) \oplus Z$$

aleshores com que hem donat $H_r(K)$ com si fos el de K_f, el programa corregeix l'error a posteriori dient que l'homologia de grau r puja en Z. Si els rangs no coincideixen, $\operatorname{Ker} f_* = (0)$ i ja està ben donat.

Si la cel·la que s'adjunta és de dimensió màxima (n), posarem rang $M_n = 0$, per tla com no hi ha vores no trivials.

5. INPLEMENTACIÓ I ÚS

Seguidament s'inclou un llistat del programa i de les subrutines.

Les línies que comencen amb "D" a la primera columna no s'han de compilar, si hom desitja la versió normal. Ara bé: si alhora que hom fa l'homologia hom vol que es faci una llista dels cicles del complex, de les vores expressades en aquesta base dels cicles, etc.; aleshores caldrà eliminar aquestes "D" del programa font a fi i efecte que es compilin.

El complex s'ha d'entrar com un arxiu assignat a la unitat 2 de lectura. A cada línia, hi posem un símplex; cada un és expressat pels nombres corresponents als vèrtexs donats de manera ascendent. El format és: un espai en blanc a la primera posició, 5 camps enters de dues posicions per a allotjar cada un dels vèrtexs (la dimensió del complex està limitada a 4 en aquesta versió), separats per un espai en blanc. A la posició 30, una xifra, que indica el nombre de vèrtexs del símplex (és a dir la dimensió més u).

Els símplexs els donarem segons dimensió decreixent; començant pels de dimensió igual a la del complex i acabant pels 0-símplexs: els vèrtexs del complex. Al final de tot s'inclourà un registre amb zeros, per tal d'assenyalar el final de l'arxiu.

En aquest arxiu, s'hauran de posar tots els símplexs del complex. És a dir: el fet de posar-n'hi un no vol dir que es tinguin en compte les seves cares automàticament, sinó que s'hauran d'explicitar. Tanmateix no és difícil de fer un altre programa que desenvolupi el complex i hi afegeixi les cares.

L'estructura de l'aplicació f d'adjunció es donarà, com ja s'ha dit, a través d'una descomposició simplicial de S^{r-1} en què els nombres dels vèrtexs corresponen amb els de llur imatge al complex; i llur ordre dins el símplex és com l'induït per una orientació global de S^{r-1}.

Amb aquesta triangulació hom farà un arxiu, completament similar al que descriu el complex, llevat que no cal posar el nombre de vèrtexs de cada complex a la posició 30. Ja hem dit que l'adjunció no pot ésser degenerada: és a dir, no pot aparèixer dues vegades el mateix vèrtex en una línia. Aquest arxiu s'assignarà a la unitat 3 de lectura.

En fer funcionar el programa, s'han de contestar tres qüestions: la dimensió del complex, la dimensió de la cel·la (que serà zero si no s'adjunta res) i un nombre més gran que el cardinal de qualsevol esquelet del complex (aquesta xifra està limitada per 200). Fet això, el programa llista els grups d'homologia del complex amb una cel·la adjuntada segons f de n fins a u.

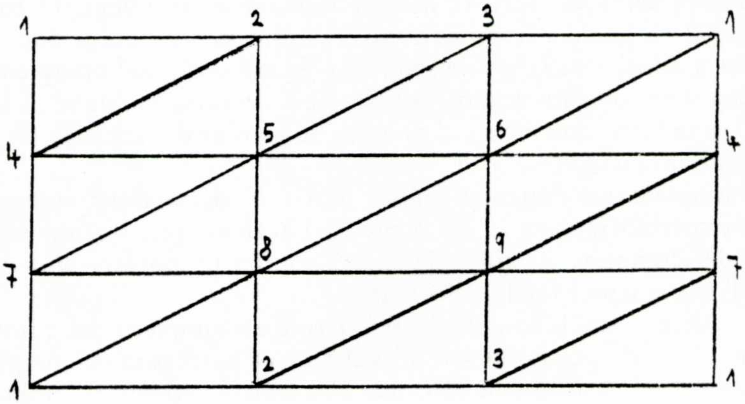
Aquest programa ha estat provat al VAX/VMS del centre de càlcul de l'U.A.B. i en aquesta versió admet símplexs de dimensió fins a 4, i esquelets de fins a 200 símplexs.

Ha estat provat amb èxit sobre una descomposició de 96 triangles referits al dodecàedre esfèric.

6. EXEMPLES

Per a fer-nos-en una idea, donarem, finalment, uns exemples:

El primer que trobem rera el llistat del programa correspon a una triangulació del tor $S^1 \times S^1$ segons la figura:



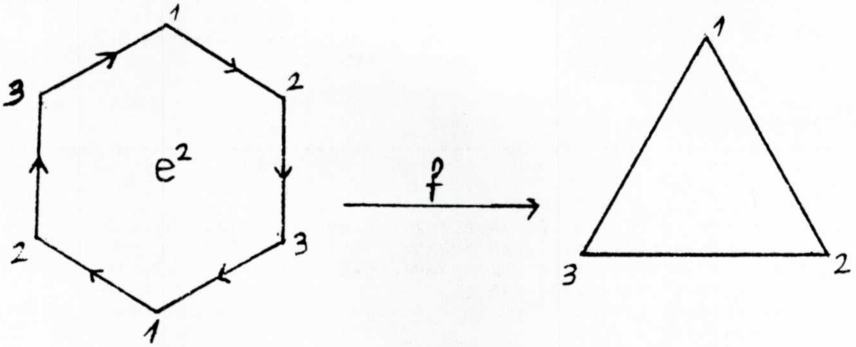
S'hi troben els símplexs des dels de dimensió 2, fins als zero (vèrtexs). El programa només té en compte les primeres 3, 2, o 1 xifres de cada línia –segons allò que hom indiqui a la columna 30. No fem cabal del valor de les restants (són residus d'un programa anterior per a desenvolupar les cares del complex), car el programa les ignora.

A continuació els grups d'homologia tal com els dóna.

L'altre exemple il·lustra com funciona amb l'adjunció de cel·les: el pla projectiu obtingut com adjunció d'una 2-cel·la a l'esfera mitjançant

$$\begin{array}{ccc} S^1 & \xrightarrow{f} & S^1 \\ z & \longrightarrow & z^2 \end{array}$$

Aquesta f té una aproximació simplicial com s'indica:



El primer arxiu (S^1) s'assigna a la unitat 2 de lectura, el segon (la cel·la) a la unitat 3. Dimensió del complex = 1, dimensió de la cel·la = 2, i el 30 és un nombre arbitrari més gran que qualsevol dels esquelets que hi intervenen.

Notem que a l'homologia que dóna, no fa cap correcció sobre l'homologia 2, per què s'ha d'entendre que és zero.

7. LLISTAT DEL PROGRAMA I DE LES SUBRUTINES

```
C......................................................................
C.
C.       AQUEST PROGRAMA CALCULA ELS GRUPS D'HOMOLOGIA
C.       D'UN POLIEDRE AMB ELS VERTEXS IDENTIFICATS
C.       ADJUNTANT-LI UNA CEL.LA.
C.
C.     ARXIUS:
C.       FOR002: LLISTA DELS SIMPLEXS DEL COMPLEX,
C.               SEGONS DIMENSIO DECREIXENT (ELS VERTEXS
C.               DE CADA SIMPLEX CREIXENTS)
C.               AL FINAL: UN REGISTRE AMB ZEROS.
C.      FOR003: LLISTA DE LES IMATGES SEGONS F#,DE LA
C.              VORA DE LA CEL.LA; D'ACORD AMB L'ORIENTACIO
C.              SUBORDINADA AL CICLE FONAMENTAL DE LA VORA.
C.              AL FINAL: UN REGISTRE AMB ZEROS.
C.
C.     ENTRADA:
C.       D1: DIMENSIO DEL COMPLEX
C.        X: DIMENSIO DE LA CEL.LA; SI NO S'ADJUNTA RES: ZERO
C.      LDU: UN ENTER MES GRAN O IGUAL AL MAXIM DELS CARDINALS
C.           DELS ESQUELETS DEL COMPLEX, MES 1.
C.
C.
C.   RUTINES:    IDAGO, IRESOL, IDSM
C......................................................................
C
C
      IMPLICIT INTEGER*2 (A-Z)
      DIMENSION DELTA1(200,200),DELTA2(200,200),COM(5),CA1(200,5)
        DIMENSION CA2(200,5),CA0(200,5)
      DIMENSION H(5),G(5),V(200,200),W(200,200),FINAL(200,200)
2     FORMAT(1X,5(I2,1X),T30,I1)
        WRITE(6,*)'DIMENSIO DEL COMPLEX,DIMENSIO DE LA CEL.LA,MAXIMA OCUP
        READ(5,*)D1,X,LDU
        D1=D1+1
        IF(X.EQ.D1)RANG2=0
        DU=D1+1
        D2=D1-1
90      M1=M1+1
        READ(2,2)(CA1(M1,J),J=1,5),D
        IF(D.EQ.D1)GOTO 90
        M1=M1-1
        BACKSPACE 2
        GO TO 80
C......................................................................
C.......GRUP DE DIMENSIO D1-1: TRANSFEREIX CA1, CA2,LLEGEIX CA2.........
C......................................................................
48      IF(D1.LE.2)STOP
        DO I=1,LDU
        DO J=1,LDU
        DELTA2(I,J)=0
        DELTA1(I,J)=0
        ENDDO
        ENDDO
        DO I=1,M1
        DO J=1,D1
        CA0(I,J)=CA1(I,J)
        ENDDO
        ENDDO
        DO I=1,M2
```

```
           DO J=1,D2
           CA1(I,J)=CA2(I,J)
           ENDDO
           ENDDO
           M0=M1
           M1=M2
           M2=0
           D0=D1
           D1=D2
           D2=D1-1
80         M2=M2+1
           READ(2,2)(CA2(M2,J),J=1,5),D
           IF(D.EQ.D2)GOTO 80
           M2=M2-1
           BACKSPACE 2
           WRITE(6,*)' '
           WRITE(6,*)'HOMOLOGIA DE DIMENSIO:',D1-1
           IF(X.NE.0)WRITE(6,*)'ADJUNTANT UNA CEL.LA',X
           WRITE(6,*)'H(XF)='
           IF(M0.EQ.0)GOTO 70
C.......................................................................
C.........CALCULA DELTA1 I DELTA2.......................................
C.......................................................................
         DO L=1,M0
         RU1=-1
         DO 5 I=1,D0
         S=0
         RU1=-RU1
         DO J=1,D1
         IF(I.EQ.J)S=1
         H(J)=CA0(L,J+S)
         ENDDO
C...........BUSCA I COMPARA.................
         DO 4 K=1,M1
         DO J=1,D1
         IF(H(J).NE.CA1(K,J))GOTO 4
         ENDDO
         DELTA2(K,L)=RU1
         GO TO 5
4        CONTINUE
           PAUSE '*********ESQUELET <I> INCOMPLET**********************'
5          CONTINUE
           ENDDO
C
C   LLEGEIX ELS SIMPLEXS <I-1>
C
70         DO L=1,M1
        RU2=-1
        DO 8 IB=1,D1
        RU2=-RU2
        S2=0
        DO J=1,D2
        IF(IB.EQ.J)S2=1
        G(J)=CA1(L,S2+J)
        ENDDO
C..............BUSCA I COMPARA..........................
        DO 7 K=1,M2
        DO J=1,D2
        IF(G(J).NE.CA2(K,J))GOTO 7
        ENDDO
        DELTA1(K,L)=RU2
```

```
      GO TO 8
7     CONTINUE
        PAUSE '**************ESQUELET <I-1> INCOMPLET******************'
8       CONTINUE
      ENDDO
C.................................................................
C.....................CALCULA LA BASE DELS CICLES..................
C.................................................................
        CALL IDAGO(M2,M1,DELTA1,V)
        DO IZ=1,M1
        IF(DELTA1(MINO(IZ,M2),IZ).EQ.0)GOTO 11
        ENDDO
        WRITE(6,*)'(0); NO HI HAN CICLES'
        IF(X.EQ.D2)RANG2=M1
        IF(X.EQ.D1)WRITE(6,*)'L''HOMOLOGIA SUPERIOR AUGMENTA EN Z'
        GO TO 48
11      DO J=IZ,M1
        DO I=1,M1
        V(I,J-IZ+1)=V(I,J)
        ENDDO
        ENDDO
        P=M1-IZ+1
        IF(X.EQ.D2)RANG2=M1-P
D       DO J=1,P
D       WRITE(6,20)J,(V(I,J),(CA1(I,K),K=1,5),I=1,M1)
D20     FORMAT(7H1CICLE:,I2/(3X,I2,T20,5(I2,1X)))
D       ENDDO
C
C          LECTURA DE LA CEL.LA
C
        IF(X.NE.D1)GOTO 60
        M0=M0+1
400     READ(3,2)(COM(I),I=1,D1)
        IF(COM(1).EQ.0)GOTO 60
        C=1
        DO I=1,D2
        DO J=I+1,D1
        C=C*(COM(J)-COM(I))
        ENDDO
        ENDDO
        DO 300 I=1,M1
        DO 200 J=1,D1
        DO K=1,D1
        IF(COM(J).EQ.CA1(I,K))GO TO 200
        ENDDO
        GO TO 300
200     CONTINUE
        DELTA2(I,M0)=DELTA2(I,M0)+ISIGN(1,C)
        GOTO 400
300     CONTINUE
        PAUSE '*******ADJUNCIO INCORRECTA**********'
        GOTO 400
C.............................................................
C..........EXPRESA LES VORES I F#(ALFA) EN LA BASE DELS CICLES.....
C.............................................................
60      IF(M0.EQ.0)THEN
        DO L=1,P
        WRITE(6,*)'Z'
        ENDDO
        GO TO 48
        ENDIF
```

```
          CALL IRESOL(M1,P,V,W,DELTA2,MU)
          DO I=1,MU
          DO J=1,P
          DELTA2(J,I)=DELTA2(J,I)/V(J,J)
          ENDDO
          DO J=1,P
          AUX=0
          DO K=1,P
          AUX=AUX+W(J,K)*DELTA2(K,I)
          ENDDO
          FINAL(J,I)=AUX
          ENDDO
          ENDDO
D         DO J=1,MU
D         WRITE(6,40)J,(FINAL(I,J),I=1,P)
D         ENDDO
40        FORMAT('0'/25(2X,I2))
C..........................................................
C...........DIAGONALITZA PER TROBAR EL GRUP QUOCIENT...
C..........................................................
          CALL IDSM(P,MU,FINAL)
D         WRITE(6,40)((FINAL(I,J),J=1,25),I=1,P)
          Q=0
          S=0
          DO 14 I=1,P
          TORSIO=IABS(FINAL(I,MINU(I,MU)))
          IF(TORSIO.EQ.1)THEN
          S=S+1
          GOTO 14
          ENDIF
          IF(TORSIO.EQ.0)THEN
          WRITE(6,*)'Z'
          Q=Q+1
          ELSE
          WRITE(6,50) TORSIO
50        FORMAT(5H Z / ,I2)
          ENDIF
14        CONTINUE
          IF(P.EQ.S)WRITE(6,*)'(0)'
          IF((RANG2.EQ.P-Q).AND.(X.EQ.D1))THEN
          WRITE(6,*)'L ''HOMOLOGIA SUPERIOR AUGMENTA EN Z'
          ENDIF
      GO TO 48
      END
```

```
C.................................................................
C.       APLICA A LES COLUMNES I,J DE MT LA TRANSFORMACIO:        .
C.                 / A1I      A1J /                               .
C.                 /              /                               .
C.                 / A2I      A2J /                               .
C.     (LES COLUMNES SON DE LONGITUD N)                           .
C.................................................................
      SUBROUTINE OPCOL (A1I,A1J,A2I,A2J,MT,N,I,J)
      IMPLICIT INTEGER*2 (A-Z)
      DIMENSION MT(200,200),H1(200),H2(200)
      DO K=1,N
      H1(K)=A1I*MT(K,I)+A1J*MT(K,J)
      H2(K)=A2I*MT(K,I)+A2J*MT(K,J)
      ENDDO
      DO K=1,N
      MT(K,I)=H1(K)
      MT(K,J)=H2(K)
      ENDDO
      RETURN
      END


C.................................................................
C.       APLICA A LES FILES LA TRANSFORMACIO:                     .
C.                 / A1I      A1J /                               .
C.                 /              /                               .
C.                 / A2I      A2J /                               .
C.     (LES FILES SON DE LONGITUD M)                              .
C.................................................................
      SUBROUTINE OPFIL (A1I,A1J,A2I,A2J,MT,M,I,J)
      IMPLICIT INTEGER*2 (A-Z)
      DIMENSION MT(200,200),H1(200),H2(200)
      DO K=1,M
      H1(K)=A1I*MT(I,K)+A1J*MT(J,K)
      H2(K)=A2I*MT(I,K)+A2J*MT(J,K)
      ENDDO
      DO K=1,M
      MT(I,K)=H1(K)
      MT(J,K)=H2(K)
      ENDDO
      RETURN
      END

C.................................................................
C.       CALCULA ELS COEFICIENTS DE LA IDENTITAT DE BEZOUT DE R I S .
C.       I EL M.C.D. D PER MITJA DE L'ALGORISME D'EUCLIDES          .
C.................................................................
      SUBROUTINE BEZOUT (R,S,A,B,D)
      IMPLICIT INTEGER*2 (A-Z)
      M=IABS(R)
      N=IABS(S)
      IF(M.EQ.N)THEN
      A=ISIGN(1,R)
      D=M
      B=0
      RETURN
      ENDIF
      P0=1
      Q0=0
      P1=0
      Q1=1
```

```
      L=1
1     P2=P1
      Q2=Q1
      P1=P0
      Q1=Q0

      L=-L
      QU0=M/N
      H=N
      N=M-N*QU0
      M=H
      P0=QU0*P1+P2
      Q0=QU0*Q1+Q2
      IF(N.NE.0)GOTO 1
      D=M
      A=ISIGN(Q1,R)*L
      B=ISIGN(P1,S)*(-L)
      RETURN
      END
C.......................................................................
C.    RUTINA IRESOL:
C.
C.                        DIAGONALITZA UNA MATRIU ENTERA MT;MODIFICA U  EN:
C.       U.MT.V=L          I CONSERVA V (INVERSIBLE ENTERA)
C.
C.    SUBRUTINES: BEZOUT, OPFIL, OPCOL
C.......................................................................
      SUBROUTINE IRESOL (N,M,MT,V,U,F)
      IMPLICIT INTEGER*2 (A-Z)
      DIMENSION MT(200,200),V(200,200),U(200,200)
      DO I=1,M
      DO J=1,M
      V(I,J)=0
      ENDDO
      ENDDO
      DO I=1,M
      V(I,I)=1
      ENDDO
      DO II=1,MIN0(M,N)
      IF(MT(II,II).NE.0)GOTO 2
      DO I=II,N
      DO J=II,M
      IF(MT(I,J).NE.0)GOTO 1
      ENDDO
      ENDDO
      RETURN
1     CALL OPCOL (0,1,1,0,MT,N,II,J)
      CALL OPCOL (0,1,1,0,V,N,II,J)
      CALL OPFIL(0,1,1,0,MT,M,II,I)
      CALL OPFIL(0,1,1,0,U,F,II,I)
2     IF(II.GE.M)GOTO 3
      DO 3 J=II+1,M
      IF(MT(II,J).EQ.0)GOTO 3
      CALL BEZOUT (MT(II,II),MT(II,J),A,B,D)
      RP=-MT(II,II)/D
      SP=MT(II,J)/D
      CALL OPCOL (A,B,SP,RP,MT,N,II,J)
      CALL OPCOL (A,B,SP,RP,V,N,II,J)
      GO TO 2
3     CONTINUE
      IF (II.GE.N)GOTO 4
```

```
      DO 4 I=II+1,N
      IF(MT(I,II).EQ.0)GOTO 4
      CALL BEZOUT (MT(II,II),MT(I,II),A,B,D)
      RP=-MT(II,II)/D
      SP=MT(I,II)/D
      CALL OPFIL (A,B,SP,RP,MT,M,II,I)
      CALL OPFIL (A,B,SP,RP,U,F,II,I)
      GO TO 2
4     CONTINUE
      ENDDO
      RETURN
      END

C.................................................................
C.    RUTINA IDAGO:
C.                       DIAGONALITZA UNA MATRIU ENTERA MT; FACTORITZA EN:
C.       U.MT.V=L        I CONSERVA V (INVERSIBLE ENTERA)
C.
C.   SUBRUTINES: BEZOUT, OPFIL, OPCOL
C.................................................................
      SUBROUTINE IDAGO (N,M,MT,V)
      IMPLICIT INTEGER*2 (A-Z)
      DIMENSION MT(200,200),V(200,200)
      DO I=1,M
      DO J=1,M
      V(I,J)=0
      ENDDO
      ENDDO
      DO I=1,M
      V(I,I)=1
      ENDDO
      DO II=1,MIN0(M,N)
      IF(MT(II,II).NE.0)GOTO 2
      DO I=II,N
      DO J=II,M
      IF(MT(I,J).NE.0)GOTO 1
      ENDDO
      ENDDO
      RETURN
1     CALL OPCOL (0,1,1,0,MT,N,II,J)
      CALL OPCOL (0,1,1,0,V,N,II,J)
      CALL OPFIL(0,1,1,0,MT,M,II,I)
2     IF(II.GE.M)GOTO 3
      DO 3 J=II+1,M
      IF(MT(II,J).EQ.0)GOTO 3
      CALL BEZOUT (MT(II,II),MT(II,J),A,B,D)
      RP=-MT(II,II)/D
      SP=MT(II,J)/D
      CALL OPCOL (A,B,SP,RP,MT,N,II,J)
      CALL OPCOL (A,B,SP,RP,V,N,II,J)
      GO TO 2
3     CONTINUE
      IF (II.GE.N)GOTO 4
      DO 4 I=II+1,N
      IF(MT(I,II).EQ.0)GOTO 4
      CALL BEZOUT (MT(II,II),MT(I,II),A,B,D)
      RP=-MT(II,II)/D
      SP=MT(I,II)/D
      CALL OPFIL (A,B,SP,RP,MT,M,II,I)
      GO TO 2
```

```
4       CONTINUE
        ENDDO
        RETURN
        END
C.......................................................................
C.      RUTINA IDSM:                                                    .
C.                          DIAGONALITZA UNA MATRIU ENTERA MT           .
C.                                                                      .
C.                                                                      .
C.     SUBRUTINES: BEZOUT, OPFIL, OPCOL                                 .
C.......................................................................
        SUBROUTINE IDSM (N,M,MT)
        IMPLICIT INTEGER*2 (A-Z)
        DIMENSION MT(200,200)
        DO II=1,MINO(M,N)
        IF(MT(II,II).NE.0)GOTO 2
        DO I=II,N
        DO J=II,M
        IF(MT(I,J).NE.0)GOTO 1
        ENDDO
        ENDDO
        RETURN
1       CALL OPCOL (0,1,1,0,MT,N,II,J)
        CALL OPFIL(0,1,1,0,MT,M,II,I)
2       IF(II.GE.M)GOTO 3
        DO 3 J=II+1,M
        IF(MT(II,J).EQ.0)GOTO 3
        CALL BEZOUT (MT(II,II),MT(II,J),A,B,D)
        RP=-MT(II,II)/D
        SP=MT(II,J)/D
        CALL OPCOL (A,B,SP,RP,MT,N,II,J)
        GO TO 2
3       CONTINUE
        IF (II.GE.N)GOTO 4
        DO 4 I=II+1,N
        IF(MT(I,II).EQ.0)GOTO 4
        CALL BEZOUT (MT(II,II),MT(I,II),A,B,D)
        RP=-MT(II,II)/D
        SP=MT(I,II)/D
        CALL OPFIL (A,B,SP,RP,MT,M,II,I)
        GO TO 2
4       CONTINUE
        ENDDO
        RETURN
        END
```

TRIANGULACIO DEL TOR:
ASSIGNAT A LA UNITAT 2 DE LECTURA

1	2	4	0	0	3
2	3	5	0	0	3
1	3	6	0	0	3
2	4	5	0	0	3
3	5	6	0	0	3
1	4	6	0	0	3
4	5	7	0	0	3
5	7	8	0	0	3
5	6	8	0	0	3
6	8	9	0	0	3
4	6	9	0	0	3
4	7	9	0	0	3
1	7	8	0	0	3
1	2	8	0	0	3
2	8	9	0	0	3
2	3	9	0	0	3
3	7	9	0	0	3
1	3	7	0	0	3
2	4	4	0	0	2
1	4	5	0	0	2
1	2	6	0	0	2
3	5	5	0	0	2
2	5	6	0	0	2
2	3	6	0	0	2
3	6	7	0	0	2
1	6	8	0	0	2
1	3	8	0	0	2
4	5	9	0	0	2
5	6	9	0	0	2
4	6	9	0	0	2
5	7	8	0	0	2
4	7	8	0	0	2
7	8	9	0	0	2
5	8	9	0	0	2
6	8	9	0	0	2
8	9	7	0	0	2
6	9	0	0	0	2
4	9	0	0	0	2
7	9	0	0	0	2
1	8	0	0	0	2
1	7	0	0	0	2
2	8	0	0	0	2
2	9	0	0	0	2
3	9	0	0	0	2
3	7	0	0	0	2
4	4	4	0	0	1
2	4	5	0	0	1
1	2	6	0	0	1
5	5	5	0	0	1
3	5	6	0	0	1
6	3	6	0	0	1
7	6	7	0	0	1
8	6	8	0	0	1
9	3	8	0	0	1
0	0	0	0	0	0

```
DIMENSIO DEL COMPLEX,DIMENSIO DE LA CEL.LA,MAXIMA OCUPACIO
    2, 0, 30
HOMOLOGIA DE DIMENSIO:        2
H(XF)=
Z

HOMOLOGIA DE DIMENSIO:        1
H(XF)=
Z
Z
```

```
   TRIANGULACIO DE S1
1  2  0  0  0           2
2  3  0  0  0           2
1  3  0  0  0           2
2  2  0  0  0           1
1  3  0  0  0           1
3  3  0  0  0           1
0  0  0  0  0           0

   E2 DONANT DUES VOLTES A S1

01,02
02,03
03,01
01,02
02,03
03,01
00,00


DIMENSIO DEL COMPLEX,DIMENSIO DE LA CEL.LA,MAXIMA OCUPACIO
  1,2,30
HOMOLOGIA DE DIMENSIO:      1
ADJUNTANT UNA CEL.LA      2
H(XF)=
Z /  2
```