Numărul 57 / 1998

Hello CAD FANS

REVISTA DE PROIECTARE ASISTATĂ DE CALCULATOR

Consideraţii privind reprezentarea funcţiilor de două variabile

DE PETRU-AURELIAN SIMIONESCU,
UNIVERSITATEA POLITEHNICA DIN BUCURESTI,
CATEDRA DE TEORIA MECANISMELOR SI A ROBOŢILOR

Numeroase activităţi ştiinţifice şi inginereşti presupun manipularea unui mare volum de date numerice a căror corectă interpretare depinde esenţial de utilizarea unor variate reprezentări grafice. Funcţii reale de două variabile reale apar frecvent în astfel de situaţii şi prin urmare vizualizarea lor este o utilitate comună multor softuri specializate (Matlab, MatCAD, Mathematica, Maple, Excel etc.).

Funcţiile reale, continui, de două variabile reale de forma z=f(u,v) definesc suprafeţe 3D, notate în continuare cu S_f, pentru a căror reprezentare au fost dezvoltaţi algoritmi deosebit de eficienţi ca timp de calcul [1], [2], [4], [9], [11], [12]. De regulă, aceste funcţii se aproximează prin matrice $Z_{m\times n}$ cu componente $z_{ij}=f(u_i,v_j)$ i=1,m şi j=1,n distribuite regulat în interiorul domeniului de reprezentare, astfel încât fiecare coloană a matricei corespunde unei singure cordonate u iar fiecare linie corespunde unei singure coordonate v [3]. Cunoscând intervalele $u_{min}\leq u_i\leq u_{max}$ şi $v_{min}\leq v_j\leq v_{max}$ utilizate în generarea matricei Z_{mxn} se pot reconstitui tripletele (u_i,v_j,z_{ij}) fiecare dintre acestea corespunzând unui singur punct pe suprafaţa S_f.

Se cunosc câteva tipuri de reprezentări grafice pentru funcţii reale de variabilă reală, z=f(u,v), majoritatea bazându-se pe aproximări liniare cu seturi de polilinii care trec prin punctele (u_i,v_j,z_{ij}). Aceste reprezentări sunt:

1) tip familii de curbe pentru u=constant sau v=constant (un exemplu este partea superioară a Fig.1);

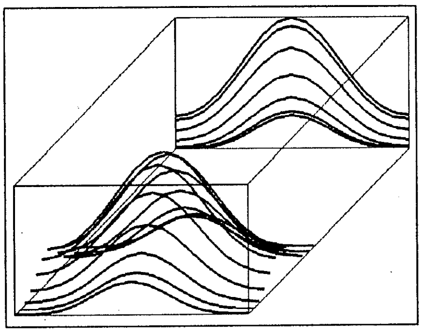

Fig.1. Familie de curbe văzută ca proiecţia frontală a unei diagrame tip „waterfall"

2) diagrame de tip „waterfall", sau mai general, reprezentări tip linii de contur (siluete) cu sau fără ascunderea porţiunilor invizibile (exemple se pot vedea în partea de jos a Fig.1 şi în Fig.6);
3) diagrame de tip „mesh" transparente (*wireframe*), sau opace, cum sunt Fig.3 şi Fig.4;
4) curbe de nivel elevate, eventual mapate pe suprafaţa S_f în reprezentare tip mesh (exemplu Fig.17);
5) suprafeţe „randate" cu diferite culori, eventual iluminate cu una sau mai multe surse

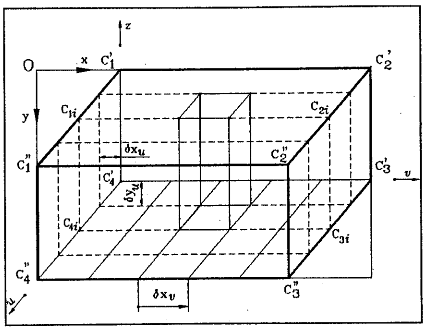

Fig.2. Linii ajutătoare pentru generarea unei proiecţii paralele-oblice

6) reprezentări de tip curbe de nivel (Fig.12).

Acestea din urmă sunt reuniuni coplanare ale curbelor rezultate prin intersecţia dintre

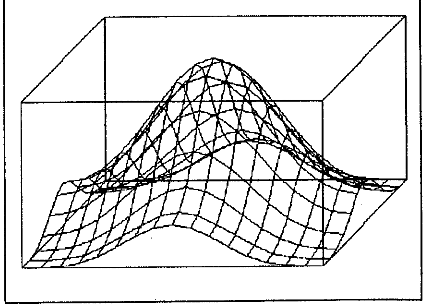

Fig.3. Reprezentare de tip „wireframe"

suprafaţa funcţiei S_f şi un număr de plane orizontale z=constant. Se obţin foarte uşor utilizând aceeaşi matrice de discretizare $Z_{m\times n}$ [9],

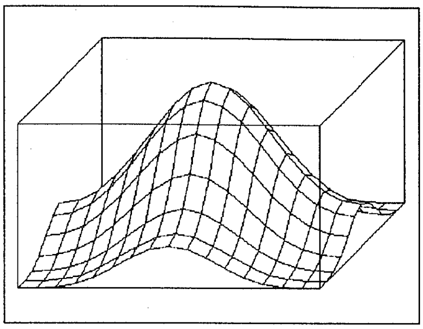

Fig.4. Reprezentare de tip „mesh"

fără a fi nevoie să se rezolve efectiv ecuația f(u,v)=constant. Figurile la care s-a făcut referire anterior sunt reprezentări ale funcției analitice $z=(\cos^2u+\cos^2v)^2$ în domeniul $[-\pi/2..\pi/2]\times[-\pi/2..\pi/2]$ pentru care în mod normal z aparține intervalului [0..4].

Agoritmii din lucrările citate mai sus, destinați reprezentărilor de funcții reale de două

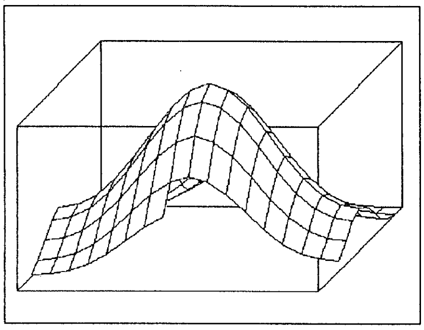

Fig.5. Succesiunea desenării și ascunderii patch-urilor într-o reprezentare de tip mesh

variabile reale, sunt orientați către dispozitive de desenare cu trasare continuă (plottere). Cu excepția algoritmului lui Kubert și col. [4], problema eliminării liniilor ascunse este rezolvată în spațiul imagine, utilizând cu succes faptul ca suprafețele patrulatere elementare (*patch*-uri) rezultate din discretizarea funcției sunt deja ordonate după profunzime.

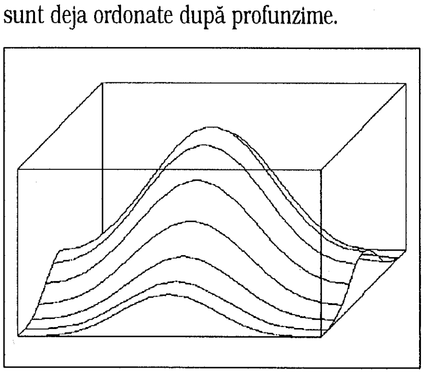

Fig.6. Reprezentare de tip"floating horizon"

În continuare se va face referire la un algoritm care utilizează pentru eliminarea liniilor ascunse metoda sortării după profunzime, propusă de Newell și col. [5] și implementat în modulul *Chart wizard* al softului Microsoft EXCEL. Metoda este cunoscută și sub denumirea de „metoda pictorului" problema vizibilității rezolvându-se prin desenarea completă a obiectelor componente ale unui ansamblu, începând cu cel mai din spate până la cel mai apropiat. Acest algoritm se pretează în mod particular unor vizualizări pe dispozitive grafice de tip raster, folosind conversia prin scanare a poligoanelor (*poligon scan conversion*, pe care se bazează de exemplu procedura *„fillpoly"* din Turbo Pascal). Plecând de la această metodă cunoscută de ascundere a zonelor invizibile în reprezentarea grafică a obiectelor 3D, autorul propune o extindere care să permită determinarea cu acuratețe a formei intersecției dintre orice suprafață S_f definită ca mai sus, și un plan orizontal *z*=constant. Problema apare atunci când se dorește vizualizarea numai a unei porțiuni a funcției studiate, delimitată prin plane orizontale paralele. Fiind integral rezolvată în spațiul imagine, această facilitate nu afectează rapiditatea intrinsecă a algoritmului de ascundere a liniilor invizibile.

Metodă de reprezentare a suprafețelor S_f

Proiecțiile paralele ale obiectelor 3D pot fi realizate manual cu instrumente clasice de desenare și asigură prelevarea comodă a datelor numerice. Diagramele tip „waterfall" folosesc astfel de proiecții și sunt utilizate de multă vreme în vizualizarea seturilor de date numerice. Diagramele de tip „waterfall" se pot interpreta ca deplasări diagonale ale unor familii de curbe așa cum se sugerează prin Fig.1. În Fig.2 se arată cum un punct C'_4 situat în interiorul unei ferestre de vizualizare (linia punctată) definită pe display-ul calculatorului, poate determina un paralelipiped prin care să se mărginească suprafața funcției S_f în proiecție oblică, generală. Este acceptat că în cazul reprezentării de funcții, păstrarea unei izotropii dimensionale în lungul celor trei axe este neesențială. Modificând poziția punctului C'_4 în interiorul ferestrei de vizualizare se poate simula schimbarea punctului din care se privește. În plus, combinând această deplasare a punctului C'_4 cu o schimbare a raportului lățime/înălțime a ferestrei de vizualizare se poate obține și o modificare a factorul de adâncime a diagramei.

Cunoscând coordonatele punctului C'_4 relativ la sistemul de axe Oxy atașat ferestrei de vizualizare (Fig. 2), se pot calcula mărimile δx_u și δy_u cu ajutorul cărora se poziționează un dreptunghi curent $C_{1i}C_{2i}C_{3i}C_{4i}$ ce mărginește un *patch* din interiorul suprafeței S_f. Similar se poate calcula mărimea intervalului δx_v care desparte fiecare două vertex-uri succesive ale unei polilinii u_i=constant:

$$\delta x_u = \frac{x'_{C4} - x''_{C4}}{m};\quad \delta y_u = \frac{y'_{C4} - y''_{C4}}{m} \quad \text{and} \quad \delta x_v = \frac{y'_{C3} - y'_{C4}}{n} \tag{1}$$

Prin urmare, coordonatele din spațiul imagine (x_{ij}, y_{ij}) ale uni vertex j al poliliniei i ce corespunde unei componente z_{ij} a matricei Z va fi:

$$\begin{aligned} x_{ij} &= x'_{C4} + \delta x_u \cdot (i-1) + \delta x_v \cdot (j-1) \\ y_{ij} &= y'_{C4} + \delta y_u \cdot (i-1) - (z_{ij} - z_{min})\frac{y'_{C4} - y'_{C1}}{z_{max} - z_{min}}. \end{aligned} \tag{2}$$

Reprezentarea unor diagrame de tip „waterfall" poate fi ușor extinsă către producerea de suprafețe „mesh" asemeni celei din Fig.3: Prin trasarea succesivă a poliliniilor corespunzătoare unui u_i constant și conectarea cu segmente de dreaptă a fiecărui vertex curent (x_{ij}, y_{ij}) cu vertex-ul $(x_{i-1,j}, y_{i-1,j})$ al poliliniei anterioare, va rezulta o a doua serie de polilinii de coordonată v_j constantă. Este necesară pentru aceasta utilizarea unui vector dinamic cu *n* componente în care să se memoreze coordonatelor vertex-urilor ce vor trebui

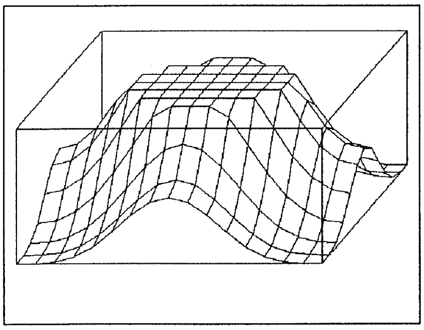

Fig.7. Rezolvarea clasică a problemei intersecției cu planul superior

unite conform procedurii descrise anterior.

Pentru generarea unei reprezentări cu liniile invizibile ascunse a suprafeței S_f (Fig.4) utilizând metoda sortării după profunzime, fiecare *patch* trebuie umplut cu culoarea de fond (sau o altă culoare) astfel încât toate segmentele aflate în spatele acestuia să devină ascunse. Succesiunea în care trebuie desenate și umplute cu culoare fiecare din aceste *patch*-uri trebuie să fie din spate spre în față, începând cu cel mai îndepărtat punct față de observator. În Fig.5, unde punctul de vedere este situat în primul octant, reprezentarea suprafeței va trebui să fie făcută de la stânga la dreapta începând cu punctul C'_4. Dacă punctul de vedere se mută în octantul patru, fiecare *patch* trebuie desenat în ordine inversă, de la dreapta spre stânga, începând cu punctul C'_3 - păstrând aceeași succesiune din spate spre în față.

Umplerea cu culoare de fond a fiecărui *patch* utilizând conversia prin scanare a poligoanelor poate fi utilizată și în rezolvarea problemei vizibilității în reprezentări de tip siluete multiple, cunoscute și sub numele de reprezentări de tip orizont flotant (*floating horizon*) a acelorași tipuri de funcții [7], [8] (vezi Fig.6). Rezolvarea problemei intersecțiilor poliliniilor succesive pentru eliminarea liniilor ascunse, ce stă la baza algoritmilor cunoscuți [9], [10], poate în acest caz fi înlocuită cu umplerea poligoanelor formate cu vertex-urile $(x_{i-1,j}, y_{i-1,j})$ în ordinea j=n,1 și a vertex-urilor $(x_{ij}; y_{ij})$ în ordinea j=1,n respectând de asemenea o succesiune de desenare din spate spre în față. O altă soluție ar fi să se șteargă toate segmentele unei categorii (u sau v) ale unei

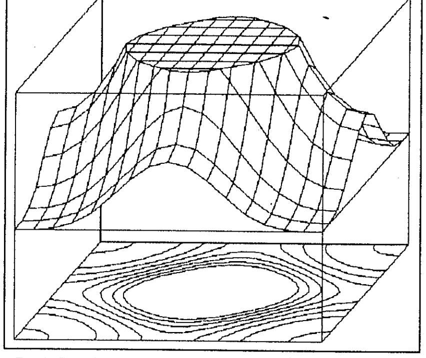

Fig.8. Rezolvarea mai exactă a problemei intersecției cu un plan orizontal

reprezentări de tip *mesh*, prin trasarea peste acestea a unor linii de aceeași culoare cu cea de umplere. Metoda de ascundere este avantajoasă deoarece nu vor mai apărea defecte de tip „margini zdrențuite" și nici cele cauzate de evaluarea vizibilității poliliniilor la un nivel mai crescut de rezoluție decât a fost funcția discretizată, defecte menționate în lucrarea [7].

Problema intersecției suprafeței S_f cu plane orizontale

Sunt numeroase situațiile când interesul în vizualizarea formei suprafeței S_f se reduce numai la o prțiune a domeniului de valori inițial $z_{min}...z_{max}$. Este, de exemplu, cazul funcțiilor obiectiv penalizate, întâlnite în problemele de optimizare, care înregistrează valori mari în afara domeniului nefezabil al spațiului de proiectare. Un alt caz este acela al funcțiilor analitice cu limite infinite în anumite porțiuni din interiorul domeniului $[u_{min},u_{max}]\times[v_{min}, v_{max}]$ pe care se dorește a fi vizualizate.

Soluția tradițională la o astfel de problemă [13] (vezi Fig.7 și 17) este de a opera asupra componentelor matricei $Z_{m\times n}$, prin înlocuirea tuturor valorilor z_{ij} mai mici decât limita modificată z_{min} cu exact z_{min} și, asemănător, înlocuirea tuturor valorile z_{ij} mai mai mari decât noul z_{max} cu acestă din urmă valoare. Metoda are dezavantajul că forma curbei de intersecție dintre suprafața funcției și planul de secțiune este aproximată numai grosier (Fig.7). Precizia poate fi îmbunătățită într-o oarecare masură prin creșterea numărului de puncte de discretizare, adică a lui *m* și *n*, conducând însă la alte neajunsuri, ușor de intuit.

În cazul proiecției paralele oblice (Fig.8) a fost găsită o soluție mult mai precisă, care rezolvă intersecția fiecărui *patch* al suprafeței S_f cu cele două plane orizontale ce formează noile limite $z_{min}..z_{max}$ de reprezentare după axa z. Vor rezulta astfel două până la cinci *subpatch*-uri, numărul maxim apărând atunci când atât planul superior cât și cel inferior intersectează același *patch*.

În Fig.9 sunt schematizate toate variantele posibile care ar putea apare la intersecția dintre unul dintre planele orizontale și un *patch*

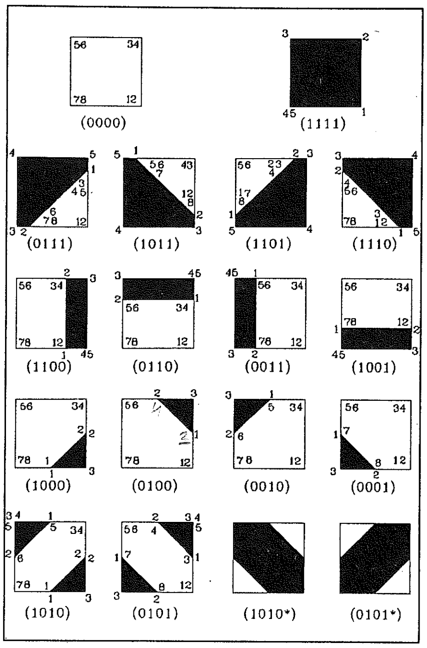

Fig.9. Variante de dispunere a unui „patch" relativ la unul din planele de secțiune

inițial notat cu $P_1P_2P_3P_4$ în figurile următoare, unde varianta (1111) semnifică un *patch* situat total în afara paralelipipedului care mărginește graficul funcției iar (0000) semnifică un *patch* care nu intersectează planul

jos \ sus	1111	0000	0111	1011	1101	1110	1100	0110	0011	1001	1000	0100	0010	0001	1010	0101	1010*	0101*
1111																		
0000		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0111		X									X							
1011		X										X						
1101		X											X					
1110		X												X				
1100		X							X				X	X				
0110		X								X	X			X				
0011		X					X				X	X						
1001		X						X				X	X					
1000		X	X					X	X			X	X	X		X		X
0100		X		X					X	X	X		X	X	X		X	
0010		X			X		X			X	X	X		X		X		X
0001		X				X	X	X			X	X	X		X		X	
1010		X										X		X		X		X
0101		X									X		X		X		X	
1010*		X										X		X		X		X
0101*		X									X		X		X		X	

Tabelul 1

orizontal considerat, dar care ar putea fi intersectat de cel de-al doilea plan orizontal (în cazul în care atât z_{min} cât și z_{max} au fost modificate față de valorile avute inițial).

În cazul intersecței cu planul orizontal inferior (proiectat în Fig.2 prin $C'_4C'_3C''_3C''_4$), zonele de culoare neagră reprezintă porțiunile *patch*-ului $P_1P_2P_3P_4$ situat dedesubt, în timp ce pentru intersecția cu planul orizontal superior (proiectat în Fig.2 prin $C'_1C'_2C''_2C''_1$) porțiunea de culoare neagră semnifică regiunea situată deasupra acestui plan. De exemplu, în varianta (0111) de intersecție cu planul orizontal inferior, vertex-urile P_2, P_3 și P_4 vor fi situate dedesubt și numai vertexul P_1 va fi situat deasupra. La rândul său, acest vertex P_1 poate să se găsească dedesubtul sau deasupra planului orizontal superior, în acest din urmă caz rezultând o variantă de intersecție (1000). Toate cazurile de intersecție cu un singur plan orizontal au fost reunite în Fig.9.

Modul cum aceste variante (0000)..(0101) de intersecție cu planul inferior, respectiv superior se pot asocia pe suprafața aceluiași *patch* a fost sistematizate într-un graf reprezentat prin Tabelul 1.

Revenind la Fig.2, paralelipipedul proiectat $C'_1..C'_4C''_1..C''_4$ poate fi considerat împărțit în $m{\times}n$ paralelipipede elementare, cu ajutorul cărora problema intersecției atât cu planul orizontal superior cât și cu cel inferior poate fi soluționată integral în spațiul imagine (bidimensional).

Există patru categorii de puncte ce pot rezulta în urma intersecției unui *patch* inițial $P_1P_2P_3P_4$ cu planul orizontal inferior și similar la intersecția cu planul orizontal superior. Coordonatele colțurilor *patch*-ului inițial vor fi notate cu $P_1(x_{ij}, y_{ij})$, $P_2(x_{ij-1}, y_{ij-1})$, $P_3(x_{i-1,j-1}, y_{i-1,j-1})$ și $P_4(x_{i,j-1}, y_{i,j-1})$, iar punctele de intersecție cu M, N, Q și R așa cum se vede în Fig.10 precum și în Fig.11. Aceeași literă va fi folosită când intersecția are loc pe aceeași latură orizontală a paralelipipedului elementar.

Varianta (1001), schematizată în Fig.10, va da naștere la punctele de intersecție M și Q, iar varianta (0011) reprezentată în Fig.11 la punctele N și R. *Patch*-ul inițial cu numai patru laturi $P_1P_2P_3P_4$ se va transforma în două *subpatch*-uri adiacente cu o linie comună MQ respectiv RN. Așa cum se poate observa, regiunea situată sub planul de intersecție $C'_4C''_4C''_3C'_3$ trebuind să fie „aliniată" respectivului plan. Aceste transformări de intersecție și aliniere au fost generalizate pentru toate variantele reunite în Fig.9.

După cum reiese din Fig.10 și 11, coordonatele punctelor M, N, Q și R relativ la sistemul de coordonate Oxy al ferestrei de afișare vor fi:

$$x_M = x_{P1} - \frac{y_M - y_{14}}{\delta y_u} \cdot (x_{P2} - x_{P1}) \quad y_M = \frac{y_{P1} \cdot y_{23} - y_{P2} \cdot y_{14}}{y_{P1} - y_{P2} - \delta y_u} \tag{3}$$

$$x_N = x_{P2} - \frac{y_M - y_{P2}}{y_{P3} - y_{P2}} \cdot \delta x_v \quad y_N = y_{23} \tag{4}$$

$$x_Q = x_{P4} - \frac{y_Q - y_{14}}{\delta y_u} \cdot (x_{P3} - x_{P4}) \quad y_Q = \frac{y_{P4} \cdot y_{23} - y_{P3} \cdot y_{14}}{y_{P4} - y_{P2} - \delta y_u} \tag{5}$$

$$x_R = x_{P1} - \frac{y_R - y_{P1}}{y_{P4} - y_{P1}} \cdot \delta x_v \quad y_R = y_{14} \tag{6}$$

În relațiile de mai sus s-au folosit notațiile y_{23} și y_{14} care depind de tipul planului, inferior sau superior, după cum urmează:

$$y_{23} = \begin{cases} y'_{C4} + \delta y_u \cdot (i-1) & \text{pentru planul inferior} \\ y'_{C1} + \delta y_u \cdot (i-1) & \text{pentru planul superior} \end{cases} \tag{7}$$

$$y_{14} = \begin{cases} y'_{C4} + \delta y_u \cdot i & \text{pentru planul inferior} \\ y'_{C1} + \delta y_u \cdot i & \text{pentru planul superior} \end{cases} \tag{8}$$

Cu excepția variantelor (1010*) și (0101*), care dau naștere la două *subpatch*-uri a căror orientare rămâne neschimbată (regiunile de culoare albă), toate celelalte variante din Fig.9 vor rezulta într-un singur *subpatch* „nealiniat". Prin neglijarea acestor două cazuri notate cu asterisc, care apar relativ de puține ori, se poate obține o simplificare a algoritmului de reprezentare grafică după cum urmează. Se va putea utiliza un poligon generic cu 8 vârfuri, în care fiecare două-câte-două vor fi inițial confundate, așa cum este adnotată varianta (0000) din Fig.9. În cazul în care se produce o intersecție, acest poligon se va modifica prin separarea sau reunirea în continuare a vertex-urilor, variantele extreme fiind (1111) când poligonul dispare (se reduce la 8 puncte confundate care să se reprezinte - eventual - undeva în afara spațiului vizibil), sau se extinde într-un octogon nedegenerat, cum este cazul variantelor suprapuse (1010-sus) + (0101-jos) sau (0101-jos) + (1010-sus). În paralel cu acest poligon generic cu 8 vârfuri, alte două poligoane cu 5, respectiv 3 vârfuri, vor trebui considerate pentru reprezentarea de această dată a *subpatch*-urilor aliniate. Aceste poligoane vor avea un comportament similar, putând exista:

a) ca un singur punct,

b) cu vârfurile parțial confundate (pentru poligonul cu 5 vârfuri), sau

c) ca pentagoane și triunghiuri nedegenerate (vezi zonele de culoare neagră din Fig.9).

Succesiunea prin care aceste trei poligoane vor trebui desenate și umplute cu culoare pentru a obține o ascundere corectă a liniilor invizibile este începând cu cel (cele) aliniate planului orizontal inferior și terminând cu cele aliniate planului orizontal superior. De asemenea trebuie păstrată ordinea dinspre spate spre față și de la stânga spre dreapta (sau de la dreapta spre stânga) asemeni cazului fără intersecții din Fig.5.

Reprezentări tip curbe de nivel

Așa cum s-a sugerat deja prin Fig.8, curba rezultată din intersecția suprafeței funcției S_f cu oricare din planul orizontal inferior sau superior, este de fapt o curbă de nivel în proiecție oblică [3]. Considerând un număr de astfel de intersecții, de fiecare dată pentru o altă poziție a suprafeței S_f translatată vertical, și prin aplicarea unei transformări „shearing" adecvate, se pot genera diagrame tip curbe de nivel asemenei celor din Fig.12. Evident rezoluția reprezentării va depinde de densitatea cu care funcția f(u,v) a fost discretizată,

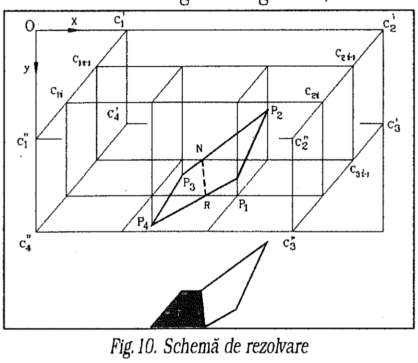

Fig.10. Schemă de rezolvare a variantei de intersecție (1001)

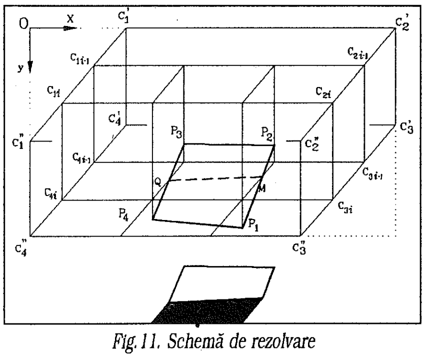

Fig.11. Schemă de rezolvare a variantei de intersecție (0011)

după cum se poate constata comparând exemplele a și b din Fig.12.

Variantele (1010*) și (0101*) din Fig.9, ignorate la reprezentarea în perspectivă a suprafeței S_f, pot da rezultate incorecte cum sunt de exemplu „punțile" din partea superioară a suprafeței 3D reprezentată în Fig.13. Această figură reprezintă (cu o rezoluție m=25 și n=40) funcția z=sin((cos$u$-sin$v$)·π) pentru u și v în domeniul [-π/2..π/2], planul superior de intersecție fiind la z_{max}=0.975 față de valoarea inițială z_{max}=1. Defectele menționate ar putea fi corectate prin considerarea nu numai a variantelor (1010) și (0101), ci și a perechilor lor (1010*) și (0101*), distincția putându-se face prin estimarea valorii curburii locale a suprafeței S_f.

Soluția adoptată, relativ simplă, a fost aplicată numai pentru corectarea curbelor de nivel. Ea a dat rezultate satisfăcătoare în majoritatea reprezentărilor de funcții experimentate (vezi partea de jos a Fig.13), și constă în compararea valorilor z_{ij}, $z_{i-1,j}$, $z_{i-1,j-1}$ și $z_{i,j-1}$ corespunzătoare punctelor P1, P2, P3 și P4 cu valorile vecine z_{ij+1}, $z_{i-1,j+1}$, $z_{i-1,j-2}$ și $z_{i,j-2}$. În Fig.14 și 15 este ilustrat exemplul diferenței dintre variantele (1010) și (1010*), în care cotele punctelor vecine P'$_2$ and P'$_4$ au servit la estimarea semnului curburii suprafeței S_f în zona *patch*-ului $P_1P_2P_3P_4$. Dacă în particular suprafața are în zona considerată un punct șa, programul elaborat de autor trasează două segmente încrucișate prin unirea punctelor MQ și NR.

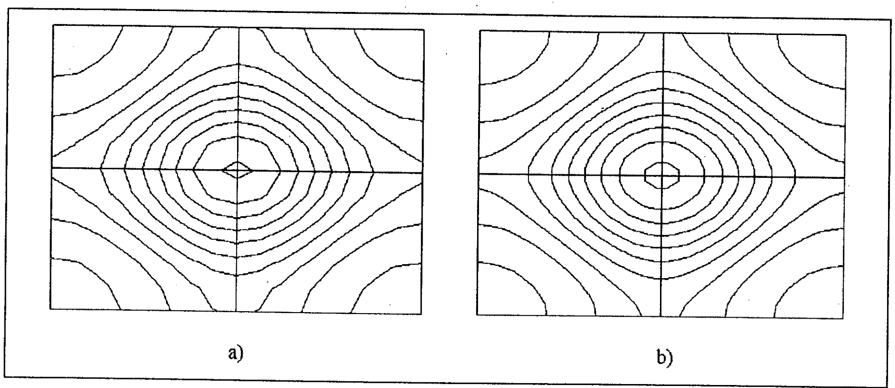

Fig.12. Curbe de nivel produse pentru o rezoluție (a) 13´17 și (b) 26´34

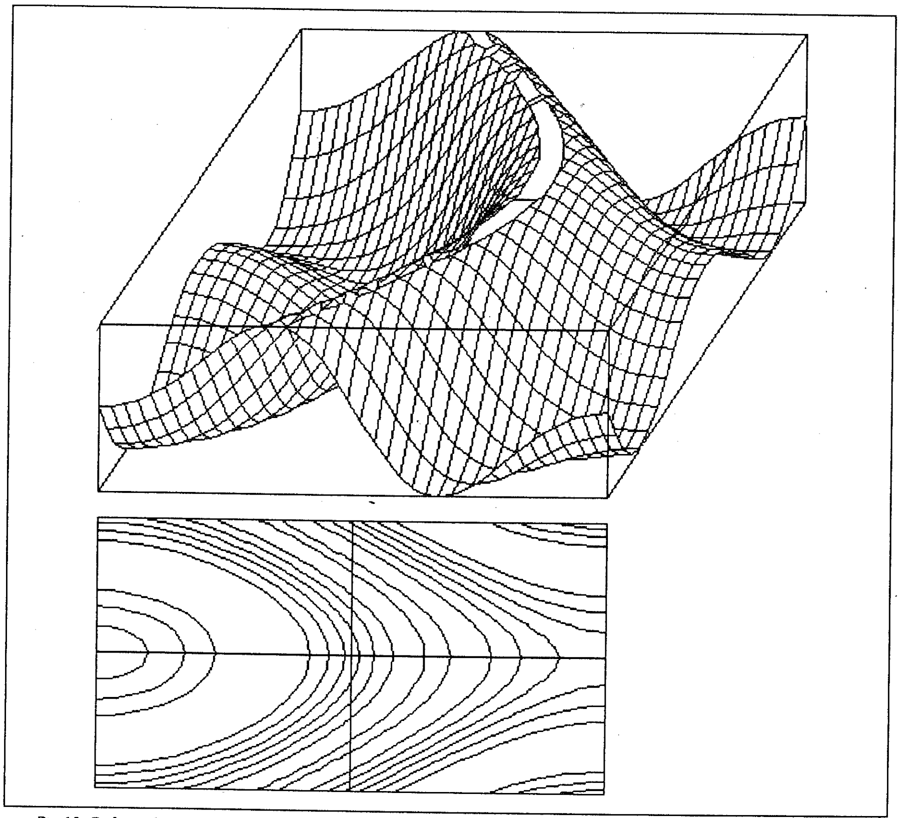

Fig.13. Defecte datorate ignorării variantelor (1010) și (0101*) și eliminarea lor într-o diagramă tip curbe de nivel, prin estimarea curburii locale*

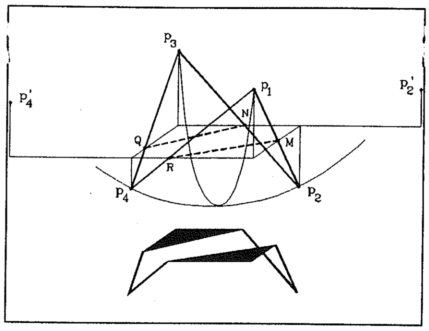

Fig.14. Varianta (0101) rezultată în condițiile unei curburi locale pozitive a suprafeței Sf

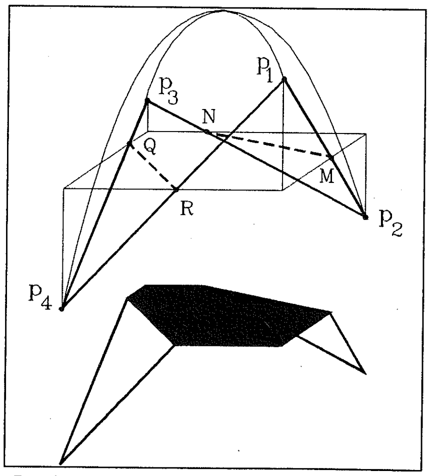

Fig.15. Varianta (0101) rezultată în condițiile unei curburi locale negative a suprafeței Sf*

Discuții și concluzii

A fost prezentată o metodă simplă pentru reprezentarea grafică a funcțiilor reale de două variabile reale într-un domeniu trunchiat de valori al funcției. Problema ascunderii liniilor

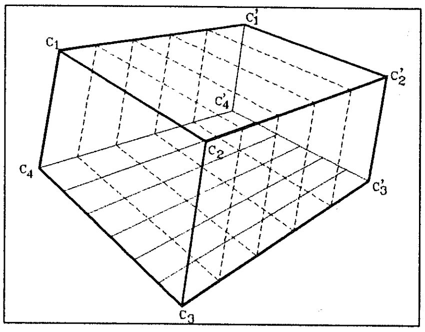

Fig.16. O generalizare a liniilor ajutătoare din Fig.2

invizibile ca și problema intersecției suprafeței funcției cu planul/planele orizontale de intersecție, a fost rezolvată cu succes în cele două dimensiuni ale spațiului imagine.

Generalizând paralelipipedul proiectat $C'_1C'_2C'_3C'_4C''_1C''_2C''_3C''_4$ care mărginește suprafață S_f ca și patrulaterele ajutătoare ce definesc direcțiile *u* sau *v* constante din Fig.2, se pot genera și alte tipuri de proiecții ale suprafeței S_f (Fig.16).

Modulul Chart wizard al softului Microsoft EXCEL utilizează triangularizarea suprafeței pentru implementarea *algoritmului pictorului* de ascundere a suprafețelor invizibile și permite alegerea și a altor tipuri de proiecții, inclusiv cea paralelă-oblică. Trunchierea domeniului valorilor *z* ale funcției este rezolvată în schimb mod clasic (Fig.17), cu toate că lucrul cu *patch*-uri triunghiulare ar reduce substanțial numărul cazurilor de intersecție față de cele enumerate în Fig.9, simplificând problema.

Graficele din Fig.18-a și b, produse folosind programul elaborat de autor, reprezintă funcția $z=(\cos^2u+\cos^2v)2$ trunchiată la

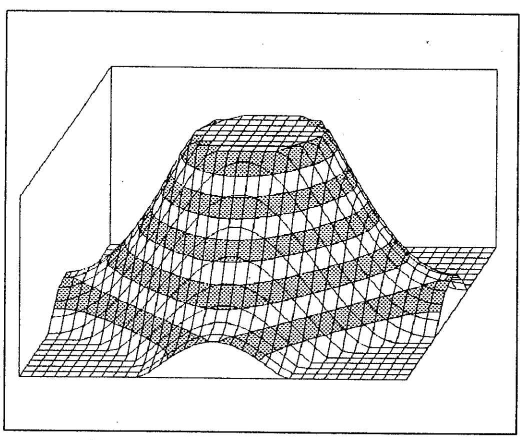

Fig.17. Reprezentare trunchiată a unei suprafețe produse cu MS EXCEL 5

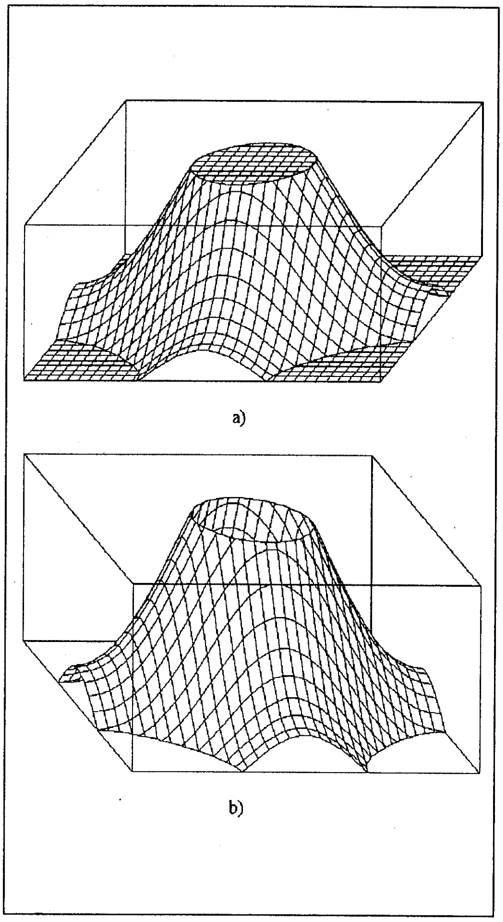

Fig.18. Reprezentări realizate cu programul elaborat de autor

domeniul de valori $z_{min}=0.5$ și $z_{max}=3$. Se poate observa o formă mai realistă a conturului intersecției cu planele orizontale inferior și superior, față de metoda clasică (Fig.17).

Conversia prin scanare a poligoanelor este o caracteristică comună dispozitivelor grafice de tip raster, făcând *algoritmul pictorului* foarte ușor de implementat în probleme de vizibilitate a suprafețelor. Algoritmul descris poate fi însă aplicat și pentru dispozitive cu trasare continuă, de tipul ploterelor, folosind algoritmul lui Patnaik și col. [6] de scanarea prin conversie a poligoanelor care lucrează cu entități de tip linie, ceea ce înseamnează că neajunsurile asociate imaginilor de tip hartă de biți vor putea fi eliminate.

Bibliografie

1. Boese, F. G., *Surface drawing made simple-but not too simple*, Computer Aided Design, Vol. 20, p. 249-258, 1988.
2. Butland, J., *Surface drawing made simple, Computer-Aided Design*, Vol. 11, p. 19-22, 1979.
3. Foley, J. D., van Dam, A., Feiner, S. K., Huges, J. F., *Computer graphics, principles and practice*, Addison-Wesley Publishing Co., Reading, Massachusetts, 1996.
4. Kubert, B, Szabo, J. Giulieri, S., *The perspective representation of functions of two variables*, J. Ass. Comput. Mach. Vol. 2, p. 193-204, 1968.
5. Newell, M. E., Newell, R. G., Sancha, T. L., *A solution to the hidden surface problem*, Proc. Ass. Comput. Mach. National Meeting, p. 443-450, 1972.
6. Patnaik, L. M., Shenoy, R. S., Krishnan, D., *Set theoretic operations on polygons using the scan-grid approach*, Computer-Aided Design, Vol. 18, p. 275-279 (1986)
7. Rogers, D. F., *Procedural Elements for Computer Graphics*, McGraw-Hill, NY, 1985.
8. Smeureanu, I., ș.a., *Grafică interactivă pe calculatoare personale*, Editura Militară, București, 1995.
9. Sowerbutts, W. T. C., *A surface-plotting program suitable for microcomputers*, Computer Aided Design, Vol. 15, p. 324-327, 1983.
10. Tănăsescu, A., ș.a. *Grafică asistată. Programe FORTRAN pentru reprezentări geometrice*, Editura Tehnică, București, 1989.
11. Williamson, H. *Hidden-line plotting program*, Commun. Ass. Comput. Mach. Vol. 15, p. 100-103, 1972.
12. Wright, T. J., *Two-space solution to the hide line problem for plotting functions of two variables*, IEEE Transaction on Computers, Vol. C-22, p. 28-33, 1973.
13. *** Displa, *Versatile display system*, *Computer-Aided Design*, Vol. 4, p. 212, 1972.