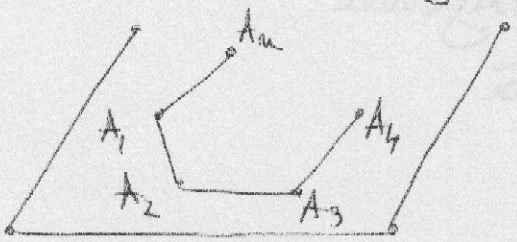
Fața / spatele unui poligon convex

Vector normal la planul unui poligon convex

Fie $A_1, A_2, \dots, A_m$ un poligon convex



Ecuația planului : $Ax+By+Cz+D=0$

unde A, B, C, D pot fi obținuți din dezvoltarea determinantului:

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{vmatrix} = 0$$

Care sunt, de fapt, coeficienții A, B, C?

De fapt : $(A, B, C) = \overrightarrow{A_1A_2} \times \overrightarrow{A_2A_3}$

⚠ Oricum am alege trei puncte consecutive, vectorii obținuți mai sus au același sens.

Vectorul $n = \dfrac{1}{\sqrt{A^2+B^2+C^2}} (A, B, C)$ s.n. normala la plan

⚠ Este independent de alegerea a 3 puncte consecutive

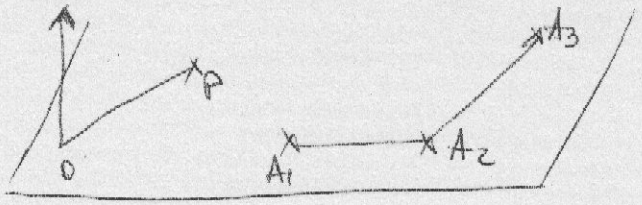
Notăm $\Pi(P) = \Pi(x, y, z) = \dfrac{1}{\sqrt{A^2+B^2+C^2}} (Ax+By+Cz+D)$

Def. $P = (x, y, z)$ se află în fața poligonului $\Leftrightarrow \Pi(P) > 0$

$P =$ __ spatele $\Leftrightarrow \Pi(P) < 0$

De semnificație are această definiție?

Translatăm totul în 0



P ∈ fata, poligonului $\Leftrightarrow Ax+By+Cz>0 \Leftrightarrow$ produs scalar $\langle n, \overrightarrow{OP} \rangle = (x,y,z)$

Concluzie: vectorul normal indică fata poligonului

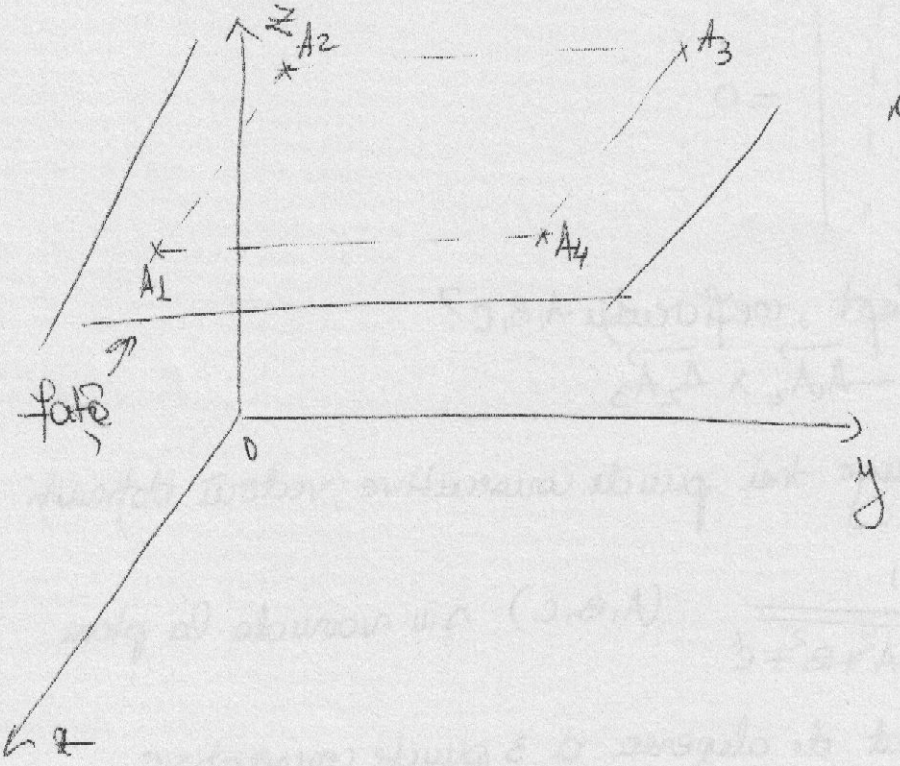
⚠ Ordinea punctelor este fundamentală

Exemplu: $A_1 = (100, -100, 100)$

$A_2 = (-100, -100, 100)$

$A_3 = (-100, 100, 100)$

$A_4 = (100, 100, 100)$

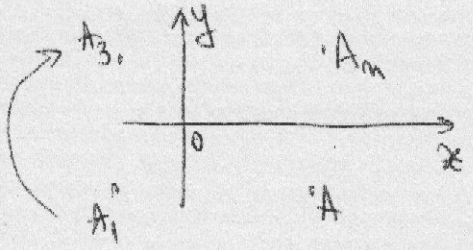


cu regula șurubului

Verificați că ec planului det de A_1, A_2, A_3 este $0x+0y-40000z+\ldots=0 \Rightarrow$

$\Rightarrow$ fata poligonului e indicată de $(0,0,-1)$ (în jos)

„privit de sus" (dinspre fata)



Obs! Poligoanele pot fi colorate/umplute folosind șabloane

- definim un tablou de 32×32 biți

GLubyte sablon[] = {0x70, 0xEA, ...};

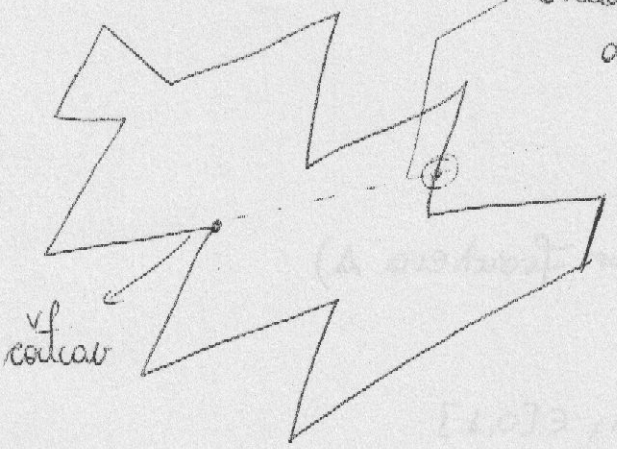
- activare / apelare / dezactivare

glEnable(GL_POLYGON_STIPPLE);
glPolygonStipple(sablon);
glBegin(GL_POLYGON);
...
glEnd();
glDisable(GL_POLYGON_STIPPLE);

Cum reprezentăm un poligon concav?

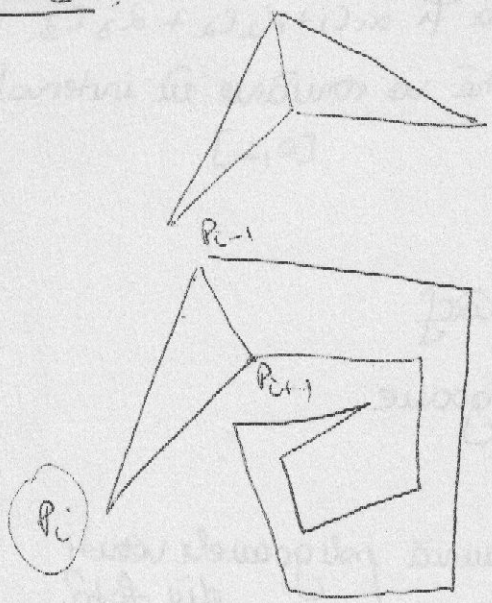
Răspuns: Triangulându-l

Met I.



această preumpere determină o descompunere a poligonului inițial în 2 poligoane fără un nr mai mic de concavități

Met II.



- Selectăm 3 vârfuri consecutive $P_{i-1}\ P_i\ P_{i+1}$ a.î. P_i convex
- $P_{i-1}\ P_{i+1}$ este „diagonală veritabilă" $\Leftrightarrow$ nici unul din celelalte vf. nu este în interiorul sau pe frontiera $\triangle P_{i-1} P_i P_{i+1}$

⚠ Orice poligon admite (cel puțin) o diagonală veritabilă.

Convenții referitor la GL_FILL:

poligonul este "umplut" (filled)

Cum?

glShadeModel (GL_SMOOTH), (implicit)

→ combinate culorile vârfului:

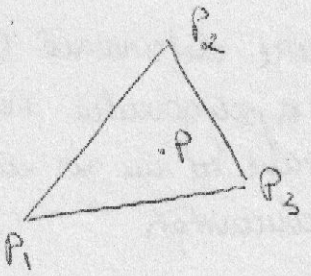
- descompunere în triunghiuri
- fiecare triunghi este colorat folosind algoritmul Gouraud

Algoritmul Gouraud pt triunghiuri

Fie $P_1P_2P_3$ vârfurile (pixel) unui triunghi colorate cu culorile c_1, c_2, c_3

($c_1, c_2, c_3 \rightsquigarrow$ RGB în $[0,1]$)

Fie P



P = punct în interiorul (sau pe frontiera Δ)

$\Rightarrow$ P se scrie unic

$P = \alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3$ cu $\alpha_1, \alpha_2, \alpha_3 \in [0,1]$

(P = combinație convexă) ⟶ culoarea sa va fi $\alpha_1 c_1 + \alpha_2 c_2 + \alpha_3 c_3$

(fiecare componentă va rămâne în intervalul $[0,1]$)

glShadeModel (GL_FLAT);

→ poligonul este umplut cu culoarea ultimului vârf

Se poate renunța la trasarea fețelor unor poligoane

```
glEnable (GL_CULL_FACE);
glCullFace (GL_FRONT);
   ... lista pol convexe
glDisable (GL_CULL_FACE);
```

→ elimină poligoanele văzute din față