

Desenarea unor curbe / suprafețe parametrizate

Cerințe:

- 1) Trasați o curbă parametrizată după modelul cercului. Exemple de curbe <https://elepa.files.wordpress.com/2013/11/fifty-famous-curves.pdf> sau <https://mathshistory.st-andrews.ac.uk/Curves/>.
- 2) Desenați un cilindru sau un con, după modelului codului sursă `08_03_sfera.cpp`. Folosiți reprezentările parametrice de mai jos, alegând intervale convenabile U și V .
 - **Cilindru** (cu raza bazei $r > 0$, fixat):

$$\begin{cases} x = r \cos(u) \\ y = r \sin(u) \\ z = v. \end{cases} \quad u \in U, v \in V$$

Normala în punctul $f(u, v)$ este coliniară cu vectorul

$$(r \cos u, r \sin u, 0).$$

- **Con** (cu raza bazei $r > 0$, fixat):

$$\begin{cases} x = v \cos(u) \\ y = v \sin(u) \\ z = v. \end{cases} \quad u \in U, v \in V$$

Normala în punctul $f(u, v)$ este coliniară cu vectorul

$$(v \cos u, v \sin u, -v).$$

- 3) Folosiți cel puțin două obiecte 3D (sfera / cilindrul / conul / etc.) pentru a schița un obiect 3D real (copac, om de zăpadă, etc.).

Coduri suport:

`08_01_cerc.cpp`

- Este utilizată reprezentarea parametrică (parametrul este θ) a cercului de centru (c_x, c_y) și de rază r

$$\begin{cases} x = c_x + r \cos \theta \\ y = c_y + r \sin \theta, \end{cases} \quad \theta \in \mathbb{R}.$$

- Se alege un interval din care va lua valori θ . Alegând valori ale lui θ (de exemplu o diviziune echidistantă a intervalului) sunt determinate, folosind reprezentarea parametrică, puncte pe cerc, obținând coordonatele vârfurilor (matricea `vf_coord`). În cod, am ales intervalul $[0, 2\pi]$ și diviziunea dată de valorile de forma $\frac{2k\pi}{n}$, cu $k \in \{0, 1, \dots, n\}$ (la funcțiile periodice trebuie atenție la periodicitate!).
- Implementare:


```
glm::vec4(cx + radius * cos(2 * PI * i / NR_POINTS),
          cy + radius * sin(2 * PI * i / NR_POINTS),
          0.0f, 1.0f);
```
- Am folosit `glm::vec4` pentru a reține coordonatele și culorile. Pentru matricea indicilor (dacă se estimează că vor fi mulți indici): `GL_UNSIGNED_SHORT`, `GL_UNSIGNED_INT`. În acest caz, dacă este utilizat un *offset* în funcția `glDrawElements()`, trebuie înmulțit cu `sizeof(GLushort)` sau `sizeof(GLuint)`.
- Informațiile despre coordonatele vârfurilor și culorile asociate sunt în același buffer, fiind utilizate sub-buffere.
- Pentru a păstra proporțiile, poate fi utilizată funcția `glutReshapeFunc(reshapeFcn)`;

08_02_cerc_cu_disc.cpp

- Pentru desenarea interiorului, am generat indici suplimentari. Schema de indexare a fost următoarea:
 - În total sunt `NR_POINTS + 3 * (NR_POINTS - 2)` indici, dintre care `NR_POINTS` pentru contur și `3 * (NR_POINTS - 2)` pentru interior (triunghiuri). Indicii sunt memorați în vectorul `vf_ind`.
 - Indexarea pentru trasarea conturului este


```
vf_ind[i] = i,
```

 deci pe poziția `i` din vectorul de indici este vârful `i`.
 - Indexarea pentru desenarea interiorului este realizată folosind grupuri de câte trei vârfuri. În total sunt `NR_POINTS-2` triunghiuri. Pentru triunghiul `i` sunt folosite vârfurile `(0, i+1, i+2)`, fiind alocați indicii `NR_POINTS+3*i, NR_POINTS+3*i+1, NR_POINTS+3*i+2`.


```
for (int i = 0; i < NR_POINTS - 2; i++)
{
    vf_ind[NR_POINTS + 3 * i] = 0;
    vf_ind[NR_POINTS + 3 * i + 1] = i + 1;
    vf_ind[NR_POINTS + 3 * i + 2] = i + 2;
}
```
- Același efect putea fi obținut aplicând `glDrawElements()`; pentru mulțimea inițială de indici.

08_03_sfera.cpp

- Elemente pentru reprezentarea suprafeței.

- (1) *Intervalele $[U_MIN, U_MAX]$, $[V_MIN, V_MAX]$ pentru parametrii considerați (u și v), indicând marginile acestora.*

Am considerat $u \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, $v \in [0, 2\pi]$.

- (2) *Numărul de paralele/meridiane, de fapt numărul de valori pentru parametri.*

Am ales $NR_PARR=10$, $NR_MERID=20$.

- (3) *Pasul cu care vom incrementa u , respectiv v .*

Se calculează folosind alegerile anterioare:

$step_u = (U_MAX - U_MIN) / NR_PARR$, $step_v = (V_MAX - V_MIN) / NR_MERID$.

În cazul în care parametrii u și v descriu curbe închise (de exemplu cercuri), trebuie atenție la manevrarea parametrilor u și v la capetele intervalelor.

- (4) *Matricele pentru vârfuri, culori, indici.*

- Din datele de mai sus și ținând cont de geometria figurii, deducem numărul de vârfuri. În acest caz avem $(NR_PARR+1) * NR_MERID$ vârfuri (nu sunt distincte, cei doi poli sunt obținuți pentru mai multe vârfuri). Avem un contor **parr** pentru paralele și un contor **merid** pentru meridian.

- Explicit indicii pentru vârfuri sunt:

$0, 1, 2, \dots, NR_PARR$	meridianul 0
$NR_PARR + 1, \dots, 2 \cdot NR_PARR + 1$	meridianul 1
$2 \cdot NR_PARR + 2, \dots, 3 \cdot NR_PARR + 2$	meridianul 2
.....	
$m \cdot NR_PARR + m, \dots, (m+1) \cdot NR_PARR + m$	meridianul m (merid)
.....	
ultimul este meridianul $NR_MERID-1$.	

Primii indici vor genera Polul Sud, ultimii Polul Nord. Contorul generic este $index = merid * (NR_PARR + 1) + parr$;

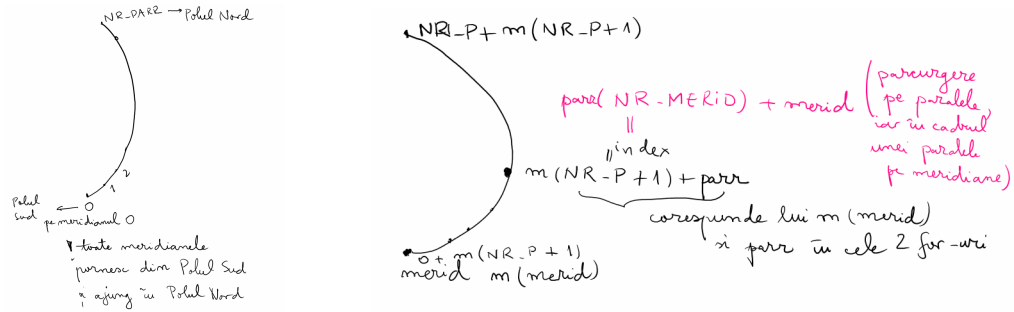
- Pentru coordonate, sunt calculate valorile parametrilor care corespund paralelei **parr** și meridianului **merid**, folosind

```
float u = U_MIN + parr * step_u;
float v = V_MIN + merid * step_v;
```

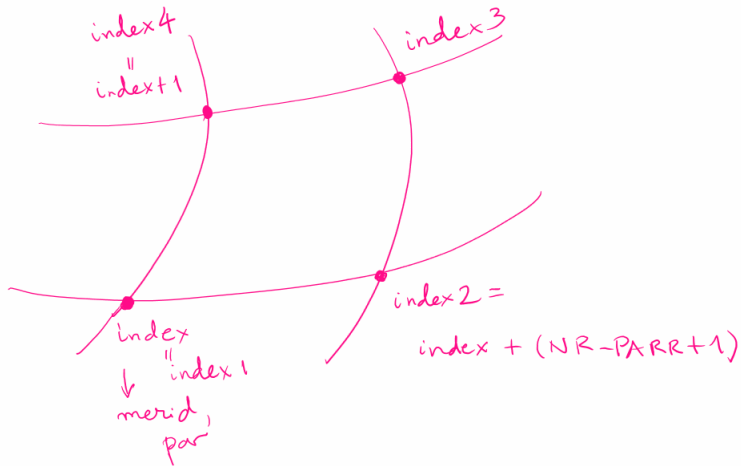
apoi este implementată reprezentarea parametrică a suprafeței:

```
float x_vf = radius * cosf(u) * cosf(v);
float y_vf = radius * cosf(u) * sinf(v);
float z_vf = radius * sinf(u);
```

- Trei grupe de indici:
 - Pentru trasarea meridianelor. Având în vedere modul în care au fost parcurși indicii-contor, întâi **merid**, apoi **parr**, aceștia sunt obținuți implicit ($\text{index} = \text{merid} * (\text{NR_PARR} + 1) + \text{parr};$).
 - Pentru trasarea paralelelor. Vârful dat de contoarele **merid** și **parr** este pe poziția $\text{parr} * (\text{NR_MERID}) + \text{merid}$ la parcurgerea cercurilor paralele.



- Pentru trasarea fețelor. Fiecărui vârf (indice) **index** care nu corespunde Polului Nord (deci **parr + 1** nu este multiplu de **NR_PARR + 1**) îi sunt asociate 4 vârfuri (4 indici) - în cod **index1**, **index2**, **index3**, **index4**. În plus, la ultimul meridian trebuie indicați indici corespunzători meridianului 0.



(5) Desenarea punctelor/muchiilor/fețelor, pe baza indexării considerate.