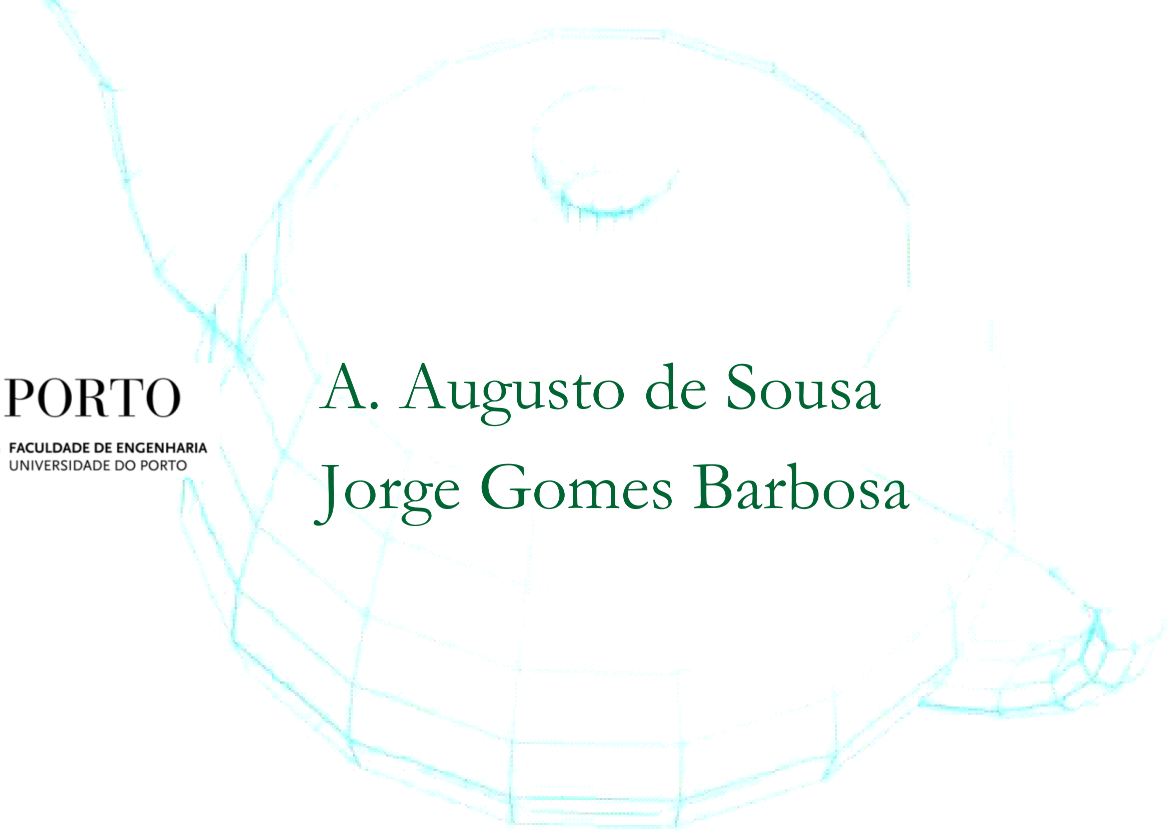
OpenGL - Evaluators



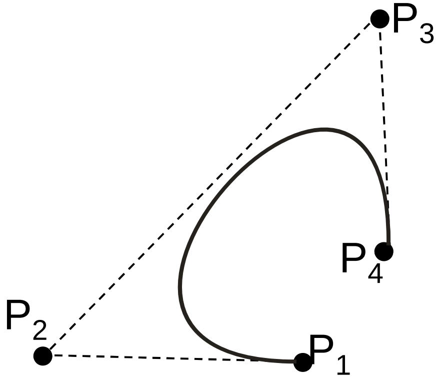
U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

A. Augusto de Sousa
Jorge Gomes Barbosa

Definitions

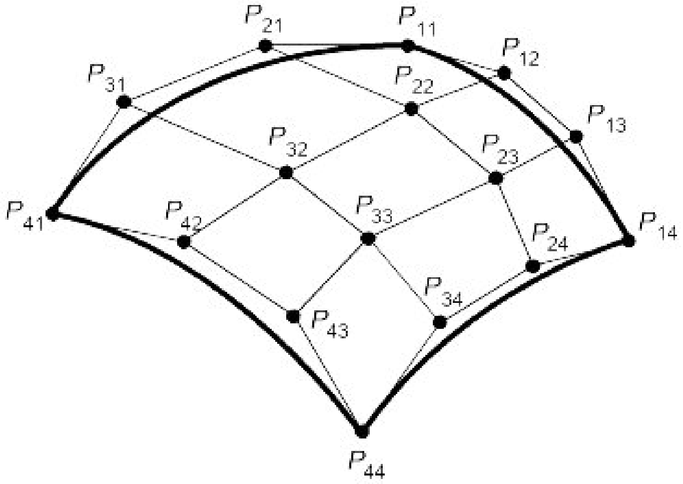
- Curve (one dimension problem)
  - Ex.: Bezier Curve

$$C(u) = \left[X(u) \quad Y(u) \quad Z(u) \right]$$

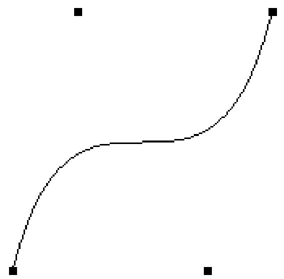


- Surface (two dimensions problem)
  - Ex.: Bezier Surface/Patch

$$S(u,v) = \left[X(u,v) \quad Y(u,v) \quad Z(u,v) \right]$$



One dimension Exemple



- Bezier Curve(cúbica)

```
void init(void)
{
...
glMap1f(GL_MAP1_VERTEX_3, 0.0, 1.0, 3, 4, &ctrlpoints[0][0]);
glEnable(GL_MAP1_VERTEX_3);
}

...

void display(void)
{
...
glBegin(GL_LINE_STRIP);
       for (i = 0; i <= 30; i++)
              glEvalCoord1f((GLfloat) i/30.0);
glEnd();
```

Evaluator Declaration

Control points

Function equivalent to **`glVertex*()`**

The function is continuous... Discretization is done with var. “i”

Parameters of glMap1*()

GL_MAP1_VERTEX_3
GL_MAP1_VERTEX_4
GL_MAP1_INDEX
GL_MAP1_COLOR_4
GL_MAP1_NORMAL
GL_MAP1_TEXTURE_COORD_1
GL_MAP1_TEXTURE_COORD_2
GL_MAP1_TEXTURE_COORD_3
GL_MAP1_TEXTURE_COORD_4

u_1

u_0

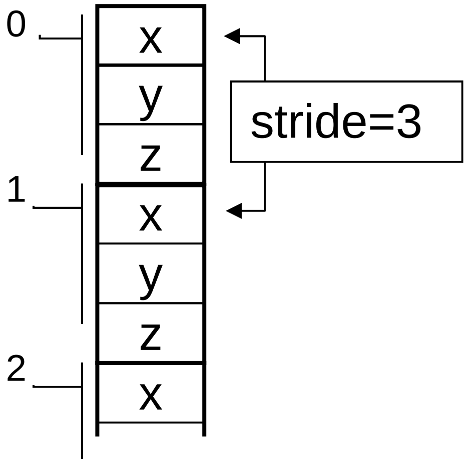
Usually:
$[u_0,u_1]=[0,1]$

```
void glMap1{fd}(GLenum target, TYPEu1, TYPEu2,
                GLint stride, GLint order, const TYPE*points)
```

1 + Curve degree

Array with the control points

x	y	z
x	y	z
x	y	z
x	y	z



```
glMap1{fd}(GL_MAP1_VERTEX_3,  0, 1,
                     3, 4,  &ctrlpoints[0][0])
```

Discrete Evaluators

- After the declaration of glMap1*():
  - A grid is declared...

```
void glMapGrid1{fd}(GLint n, TYPEu0, TYPEu1);
```

...with ***n*** steps, when ***u*** goes from $\boldsymbol{u_0}$ to $\boldsymbol{u_1}$

  - Grid is visited, drawing steps of the line...

```
void glEvalMesh1(GLenum mode, GLint p1, GLint p2);
```

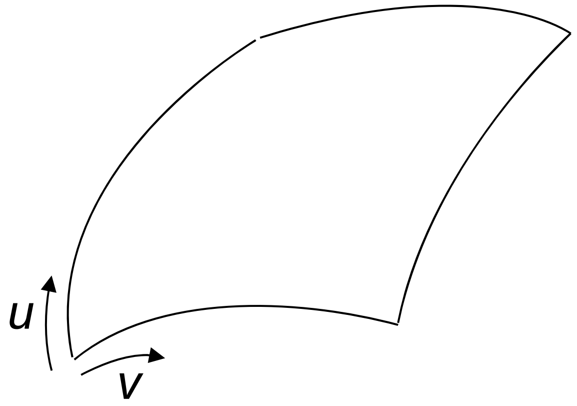
...in the proper mode, from step $\boldsymbol{p_1}$ until $\boldsymbol{p_2}$

GL_POINT
GL_LINE

Note: It is not necessary to use a cycle...
It is implicit in the function!

Two dimensions Evaluators

- Now we have two control variables: (*u*, *v*)



```
void glMap2{fd}(GLenum target,
                TYPEu1, TYPEu2, GLint ustride, GLint uorder,
                TYPEv1, TYPEv2, GLint vstride, GLint vorder,
                TYPE points);

void glMapGrid2{fd}(GLint nu, TYPEu1, TYPEu2,
                    GLint nv, TYPEv1, TYPEv2);

void glEvalCoord2{fd}(TYPE u, TYPE v);

void glEvalMesh2(GLenum mode, GLint i1, GLint i2,
                              GLint j1, GLint j2);
```

Two dimensions Evaluators

- Exemple/code available in the moodle page:

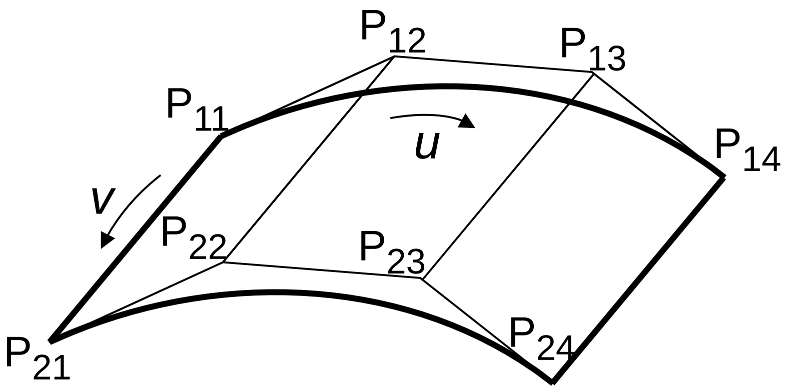
```
// In the funtion "inicializacao"
glMap2f(GL_MAP2_VERTEX_3, 0.0, 1.0, 3, 2,  0.0, 1.0, 6, 2,  &ctrlpoints[0][0]);
glMap2f(GL_MAP2_NORMAL,   0.0, 1.0, 3, 2,  0.0, 1.0, 6, 2,  &nrmlcompon[0][0]);
glMap2f(GL_MAP2_COLOR_4,  0.0, 1.0, 4, 2,  0.0, 1.0, 8, 2,  colorpoints[0][0]);

glEnable(GL_MAP2_VERTEX_3);
glEnable(GL_MAP2_NORMAL);
glEnable(GL_MAP2_COLOR_4);

// for this set of interpolators:
//    in the direction U, divisions will be made in 40 steps,
//        when variable U goes from 0 to 1
//    in the direction V, divisions will be made in 60 steps,
//        when variable V goes from 0 to 1
glMapGrid2f(40, 0.0,1.0, 60, 0.0,1.0);
```

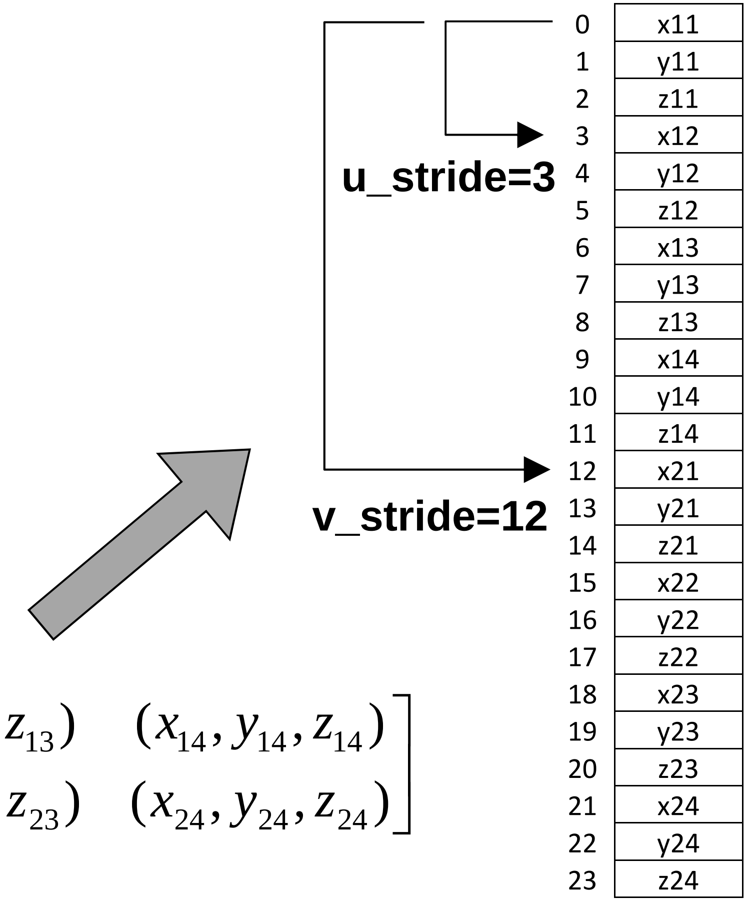
```
// In the function “display”
glShadeModel(GL_FLAT);    // GL_FLAT, GL_SMOOTH
glEvalMesh2(GL_FILL, 0,40, 0,60);          // GL_POINT, GL_LINE, GL_FILL
//glEvalMesh2(GL_FILL, 10,30, 20,40);      // polygon not complete...
//glEvalMesh2(GL_FILL, -10,50, -20,70);    // ...ou even "larger"
```

Stride and Order: example 1



4 points in direction *U* → **u_order=4**

2 points in direction *V* → **v_order=2**



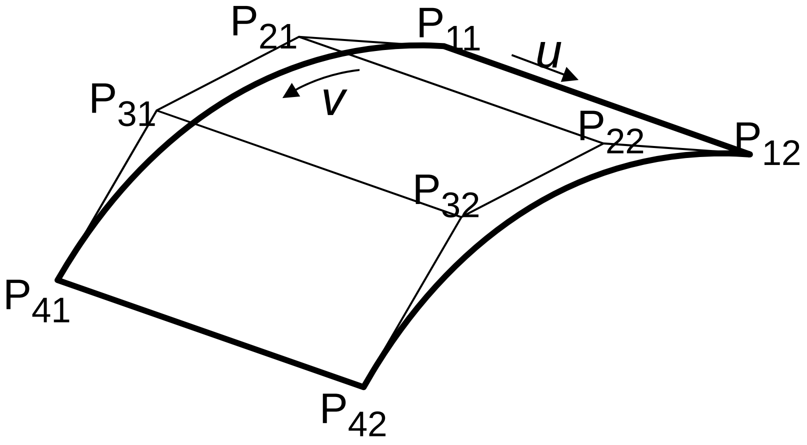
0	x11
1	y11
2	z11
3	x12
4	y12
5	z12
6	x13
7	y13
8	z13
9	x14
10	y14
11	z14
12	x21
13	y21
14	z21
15	x22
16	y22
17	z22
18	x23
19	y23
20	z23
21	x24
22	y24
23	z24

u_stride=?

$$\text{ctrlpoints} = \begin{bmatrix} (x_{11}, y_{11}, z_{11}) & (x_{12}, y_{12}, z_{12}) & (x_{13}, y_{13}, z_{13}) & (x_{14}, y_{14}, z_{14}) \\ (x_{21}, y_{21}, z_{21}) & (x_{22}, y_{22}, z_{22}) & (x_{23}, y_{23}, z_{23}) & (x_{24}, y_{24}, z_{24}) \end{bmatrix}$$

v_stride=?

Stride and Order: example 2



2 points in direction *U* → **u_order=2**
4 points in direction *V* → **v_order=4**

u_stride=?

v_stride=?

$$\text{ctrlpoints} = \begin{bmatrix} (x_{11}, y_{11}, z_{11}) & (x_{12}, y_{12}, z_{12}) \\ (x_{21}, y_{21}, z_{21}) & (x_{22}, y_{22}, z_{22}) \\ (x_{31}, y_{31}, z_{31}) & (x_{32}, y_{32}, z_{32}) \\ (x_{41}, y_{41}, z_{41}) & (x_{42}, y_{42}, z_{42}) \end{bmatrix}$$

u_stride=3

v_stride=6

0	x11
1	y11
2	z11
3	x12
4	y12
5	z12
6	x21
7	y21
8	z21
9	x22
10	y22
11	z22
12	x31
13	y31
14	z31
15	x32
16	y32
17	z32
18	x41
19	y41
20	z41
21	x42
22	y42
23	z42