# Teselace základních 3D objektů

Petr Felkel, Jaroslav Sloup
Katedra počítačové grafiky a interakce, ČVUT FEL
místnost KN:E-413 (Karlovo náměstí, budova E)
E-mail: felkel@fel.cvut.cz

Poslední změna: 25.2.2015

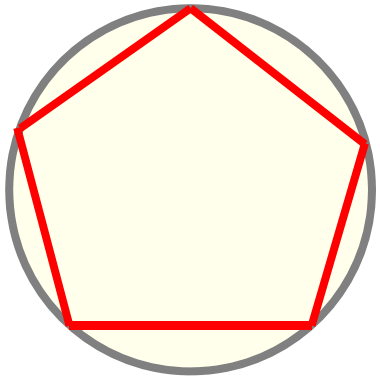
# Teselace

- Máme omezenou množinu geometrických primitiv, v podstatě trojúhelníky.
- Jak generovat povrch geometrických těles, např. koule?
- Jak to zařídit, abychom mohli zobrazit složité geometrické objekty?

# Příklad teselace:kružnice

Kružnice jako lomená čára(in 2D)
(bude se kreslit pomocí GL_LINE_LOOP)

```
#define PI 3.1415926535897;

int circlePoints = 100;
float angle;

for (int i = 0; i < circlePoints; i++) {
        angle = 2*PI*i / circlePoints;
        VytvořVrchol( cos(angle), sin(angle) );
}
```



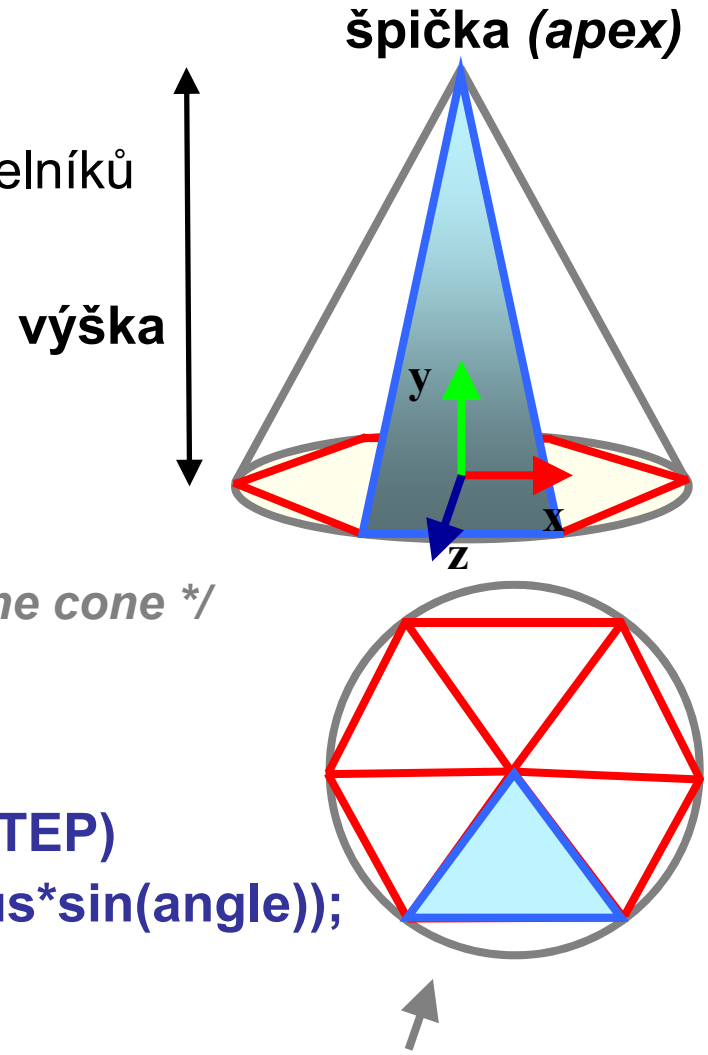
Approximace pro circlePoints = 5

**Note:** *Toto je přímočarý příklad, jak namalovat kružnici.- ne tedy efektivní varianta*

# Příklad teselace: plášť kužele

Vytvoř vrcholy pro plášť kužele z trojúhelníků
(t.j. budou se vykreslovat pomocí vějíře trojúhelníků
GL_TRIANGLE_FAN)

```
#define STEP 2*PI / circlePoints

int circlePoints = 10;
float angle;

  CreateVertex(0, height, 0);   /* the apex of the cone */

    /* approximation of the circle */
  angle = 0;
  for (int i=0; i<=circlePoints; i++, angle+=STEP)
    CreateVertex(radius*cos(angle), 0, radius*sin(angle));
```



# Příklad teselace: koule

Vytvoř vrcholy pro povrch koule z trojúhelníků (budou se vykreslovat jako pásy trojúhelníků - GL_TRIANGLE_STRIP)

**Parametrická rovnice koule:**

$x(u, v) = r \cos(u) \sin(v)$

$y(u, v) = r \sin(v)$

$z(u, v) = r \sin(u) \cos(v)$

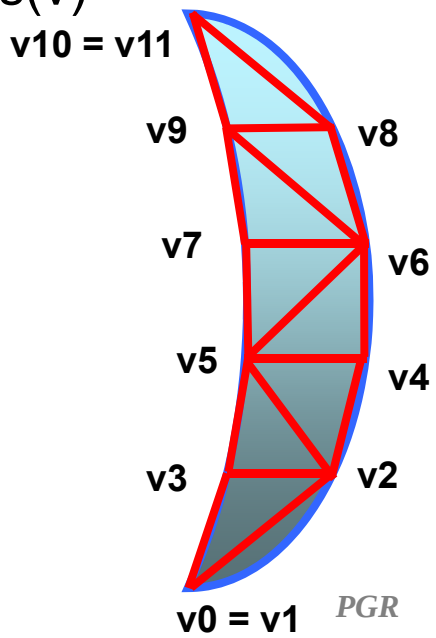
kde

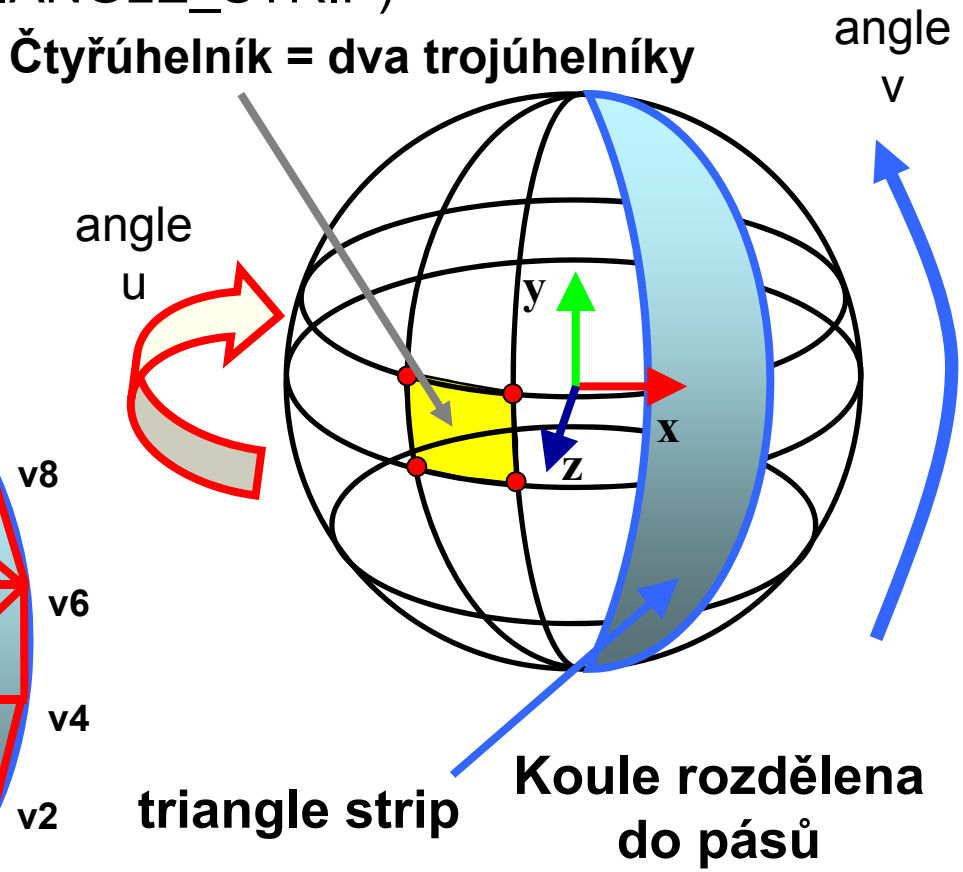
r ... poloměr

u, v ... úhly

$0 \leq u \leq 2\pi$

$-\pi/2 \leq v \leq \pi/2$



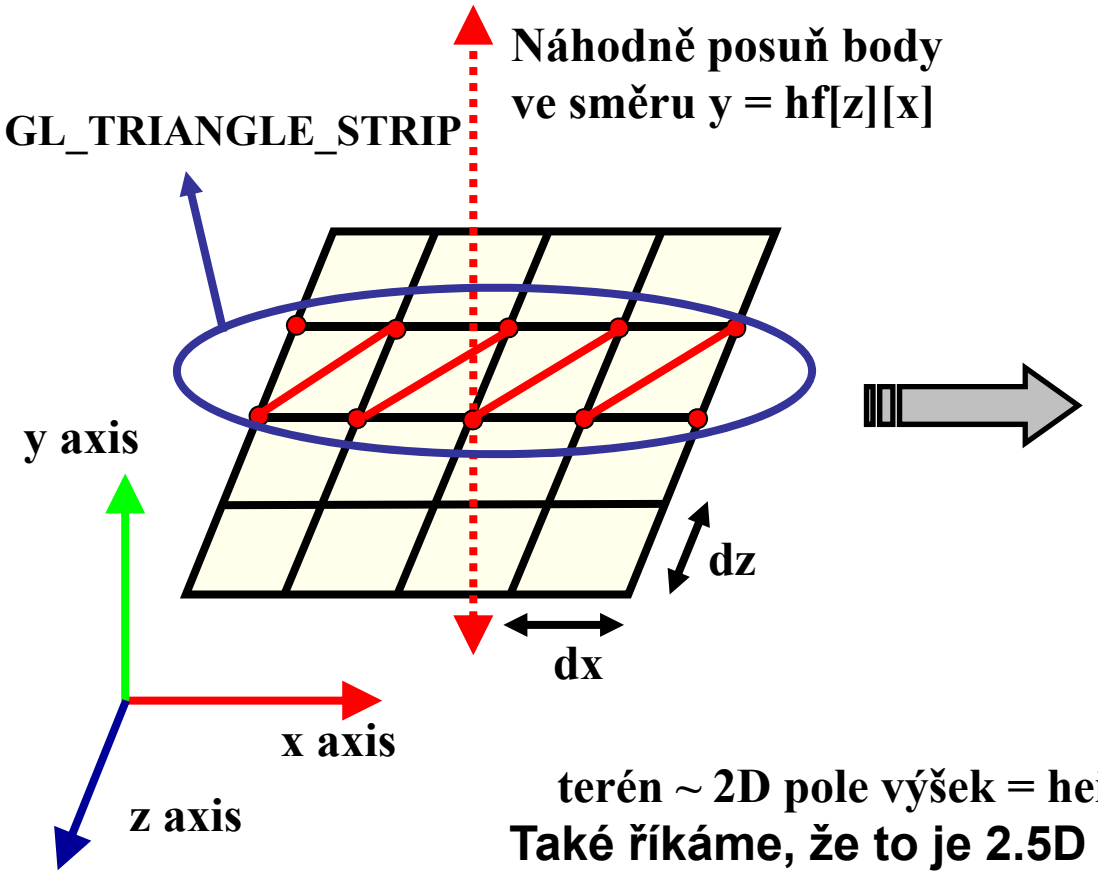



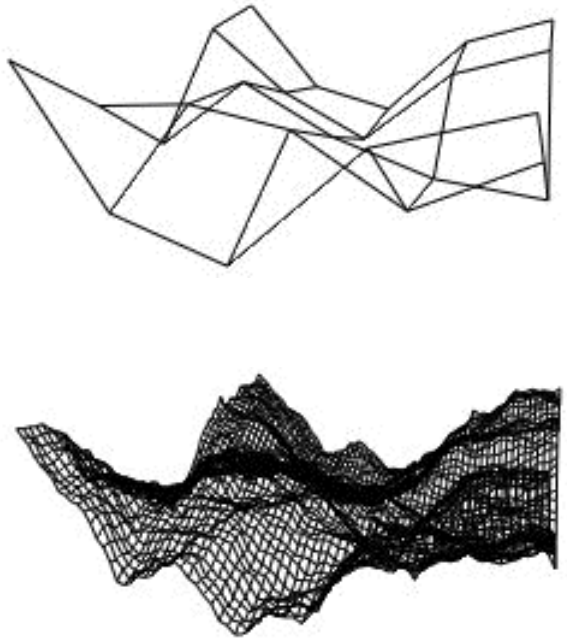
# Teselace koule (pokračování.)

```
#define STRIPS_COUNT 10
#define QUADS_IN_STRIP 5
#define Ku 2*PI/STRIPS_COUNT            /* angle step */
#define Kv PI/QUADS_IN_STRIP            /* angle step */
#define R 0.5                           /* sphere radius */

GLfloat u, v, x, y, z;

 u = 0;
 for(int strip=0; strip<=STRIPS_COUNT; strip++) {        /* smyčka přes poledníky */
   v = -PI / 2;
   for(int quad=0; quad<=QUADS_IN_STRIP; quad++) {  /*smyčka přes jeden poledník */
       x = R*cos(u)*cos(v);  y = R*sin(v);  z = R*sin(u)*cos(v);
       VytvořVrchol(x, y, z);                    /* první vrchol podle poledníku 1*/
       x = R*cos(u+Ku)*cos(v);  z = R*sin(u+Ku)*cos(v);
       VytvořVrchol (x, y, z);                   /* druhý vrchol podle poledníku 2 */
       v += Kv;
       if (quad  == 0) VytvořJedenTrojúhelník(...)          /* severní čepička */
       else if (quad  == QUADS_IN_STRIP) VytvořJedenTrojúhelník(...) /* jižní čepička */
       else VytvořDvaTrojúhelníky(...)           /*  jeden čtyřúhelník na dva trojúhelníky */
   }
   u += Ku;
 }
```

# Příklad: výšková mapa terénu

Příklad: zobrazení terénu pomocí trojúhelníků (GL_TRIANGLE_STRIP)



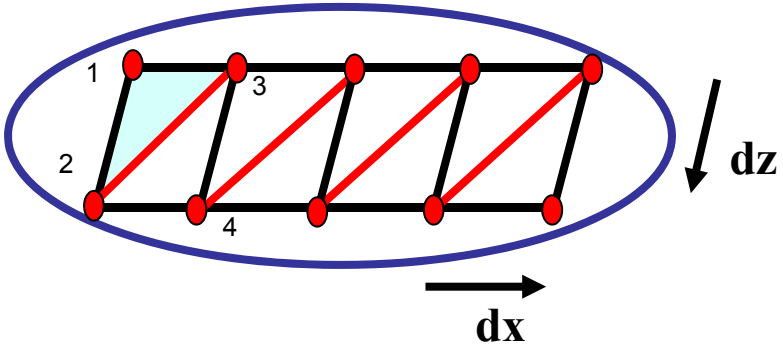
terén ~ 2D pole výšek = height field hf[]

**Také říkáme, že to je 2.5D reprezentace.**

# Příklad: výšková mapa terénu (pokrač.)

```
float px, pz;

pz = -0.75;
for(int z=0; z<N-1; z++) {                    /* loop over strips */

   /* start a new triangle strip */

  px = -0.75;
  for(int x=0; x<N; x++) {

    VytvořVrchol( px, hf[z][x], pz);          /* first vertex */
    VytvořVrchol( px, hf[z+1][x], pz+dz);/* second vertex */

    px += dx;
  }

  /* end of triangle strip */

  pz += dz;
}
```

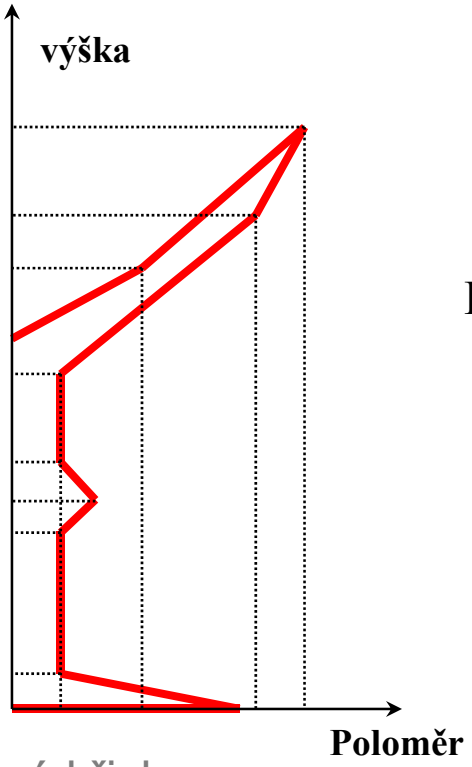
Výška v bodě [x,z]



# Příklad: Rotační těleso

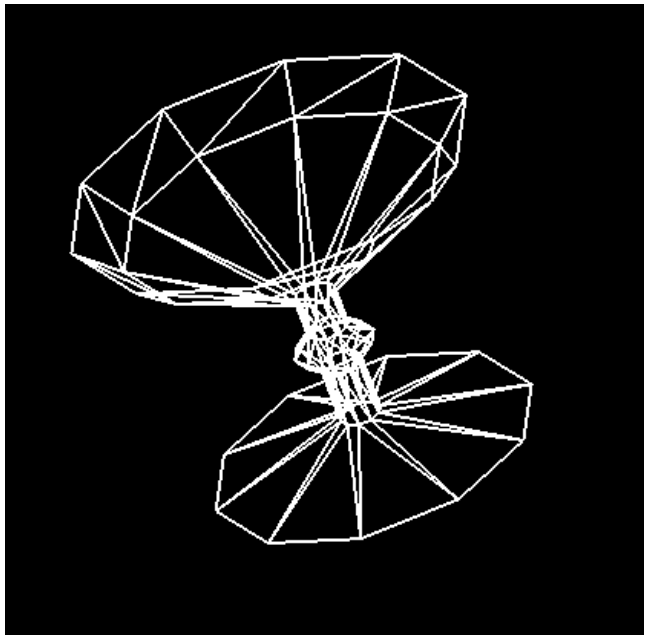
Povrch rotačního tělesa se získá otáčením dvojrozměrné křivky (profilu) okolo osy => výsledný povrch je rotačně symetrický

**Profilová křivka je zadána po částech lineární funkcí**




Profilová křivka

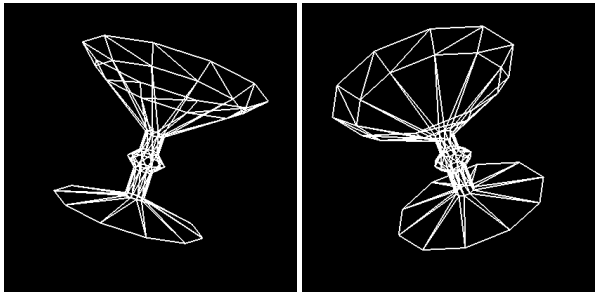
**Rotace okolo osy y**



Rotační povrch
(*surface of revolution*)

# Tessellation example: Surface of revolution (contd.)

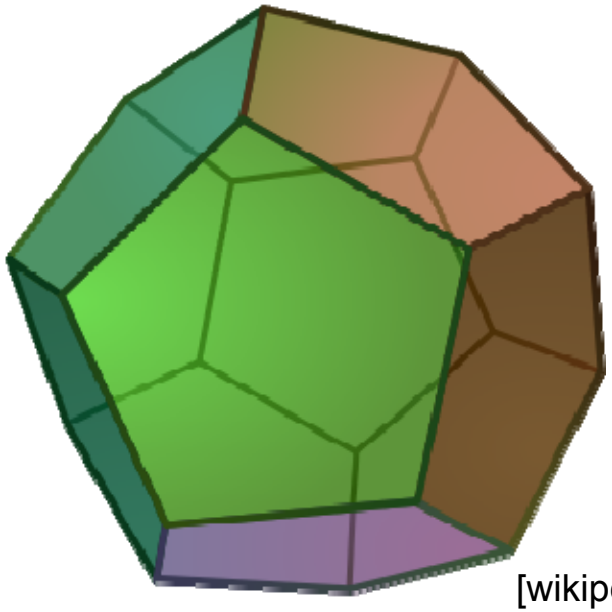
```
#define COUNT 11   /* number of points in profile */
float height[COUNT] = {0.0, 0.0, 0.2, 0.5, 0.6, 0.7, 0.9, 1.4, 1.6, 1.4, 0.9}; /* profile is defined by height and related radius */
float radius[COUNT] = {0.0, 0.8, 0.1, 0.1, 0.2, 0.1, 0.1, 0.8, 1.0, 0.75, 0.0};
#define ROT_STEPS_COUNT 10   /* number of rotation steps */
#define PI 3.1415162
#define STEP 2*PI / ROT_STEPS_COUNT
float height1, height2, radius1, radius2;
```



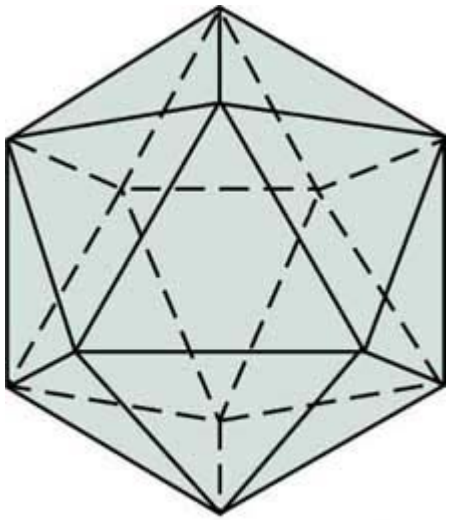
```
height2 = height[0];  radius2 = radius[0];
for(int h=1; h<COUNT; h++) {      /* loop over all strips */
    height1 = height2;    height2 = height[h];
    radius1 = radius2;    radius2 = radius[h]; angle=0.0;

    /* starrt a new triangle strip */                    /* each strip is rendered as triangle strip */
        for(int i=0; i<ROT_STEPS_COUNT; i++, angle+=STEP) {
            CreateVertex(radius1*sin(angle), height1-0.75, radius1* cos(angle));
            CreateVertex(radius2*sin(angle), height2-0.75, radius2* cos(angle));
        }
        CreateVertex(radius1*sin(0.0), height1-0.75, radius1* cos(0.0));
        CreateVertex(radius2*sin(0.0), height2-0.75, radius2* cos(0.0));
  /* end of triangle strip */
}
```

# Úkol pro chytré hlavy

- Dodekaedron – pravidelný **dvanáctistěn** složený z pravidelných pětiúhelníků
- Ikosahedron – pravidelný **dvacetistěn** složený z rovnostranných trojúhelníků
- Napište teselační rutinu pro OpenGL pro obě primitiva



[wikipedia]



[Encyclopedia of science]

# Zajímavé odkazy

- David Wolff: ***OpenGL 4.0 Shading Language Cookbook***. Packt Publishing, 2011, ISBN 978-1-849514-76-7.
- Richard S. Wright, Nicholas Haemel, Graham Sellers, Benjamin Lipchak: ***OpenGL SuperBible: Comprehensive Tutorial and Reference***. 5th ed., Addison-Wesley Professional, 2010, ISBN 0-321-71261-7.
- Ed Angel, Dave Shreiner: *An Introduction to Modern OpenGL Programming*, SIGGRAPH 2011 tutorial, http://www.daveshreiner.com/SIGGRAPH/s11/Modern-OpenGL.pptx
- Joe Groff. *An intro to modern OpenGL*. Updated July 14, 2010 http://duriansoftware.com/joe/An-intro-to-modern-OpenGL.-Table-of-Contents.html
- Vertex Array Object na OpenGL Wiki: http://www.opengl.org/wiki/Vao
- Vertex Specification na OpenGL Wiki: http://www.opengl.org/wiki/Vertex_Specification