Computação Gráfica – Modelagem Geométrica

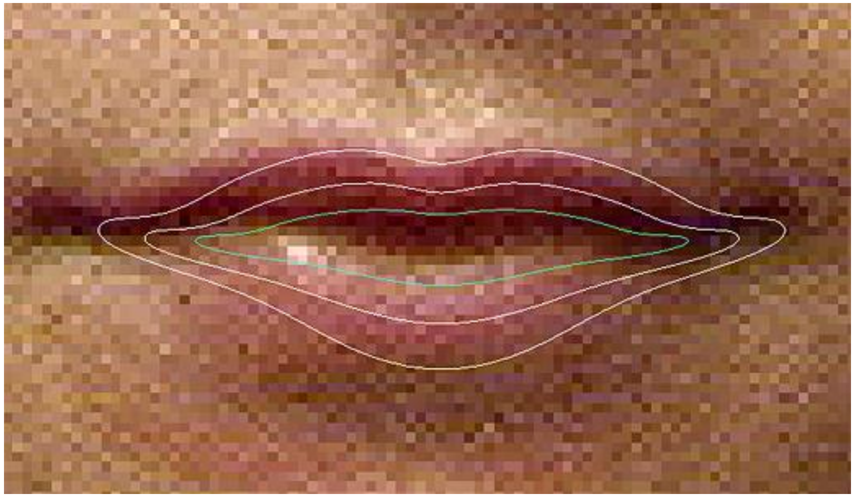
Profa. Mercedes Gonzales Márquez

Tópicos

- Curvas
- Superfícies
- Técnicas principais de Modelagem Geométrica

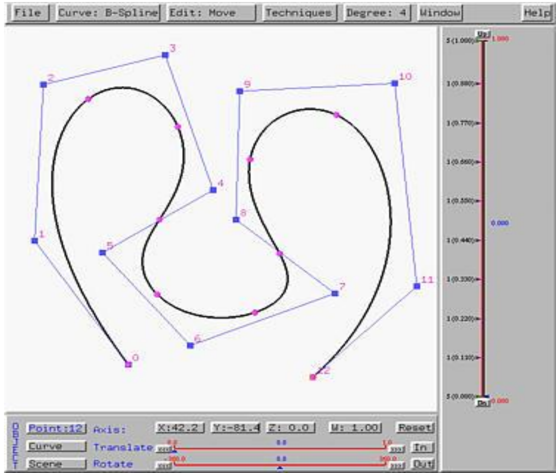
Curvas e Superfícies - Introdução

- Curvas são a base, tanto da geração de formas simples, como círculos e elipses, quanto na criação de projetos complexos como automóveis, navios, aeronaves ou até mesmo faces e corpos humanos.



Curvas e Superfícies - Introdução

- Desempenham um papel importante em várias áreas, como criação de objetos e visualização de fenômenos científicos.
- Uma curva pode ser representada:
  - Por uma sucessão de linhas retas que ligam pontos específicos.
  - Por um conjunto de pontos de controle que determina através de uma equação uma curva que passe por eles.



Representação de Curvas

- A representação analítica de curvas pode usar ou não parâmetros, se subdividindo então em:

  B.1. paramétricas

  B.2. não-paramétricas

- As formas não-paramétricas podem ser, ainda,

  B.2.1. explícitas

  B.2.2. implícitas.

Representação de Curvas

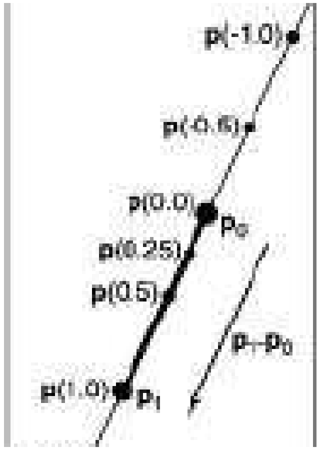
B.1 Formas paramétricas

– As coordenadas são dadas em termos de um ou um conjunto de parâmetros (t, θ, etc.).

Exemplo de curvas no plano: P(t) = (x(t), y(t))

➢ Segmentos

$$p(t) = p_0 + t(p_1 - p_0)$$



Representação de Curvas

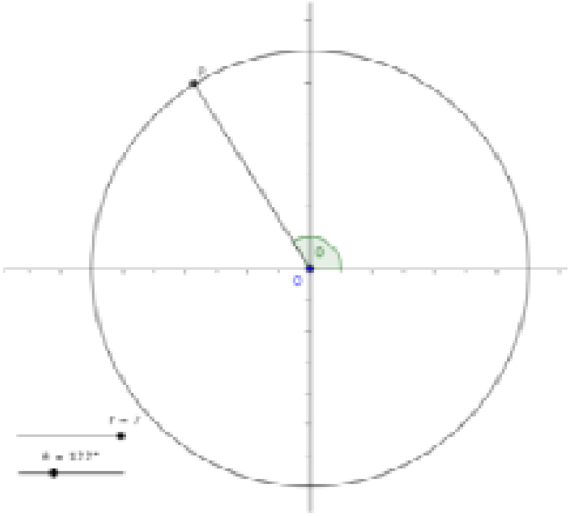
B.1 Formas paramétricas

$$P(t) = (x(t), y(t))$$

Exemplos:

Circuferências

$$p(t) = (r \cos t, r \operatorname{sen} t)$$



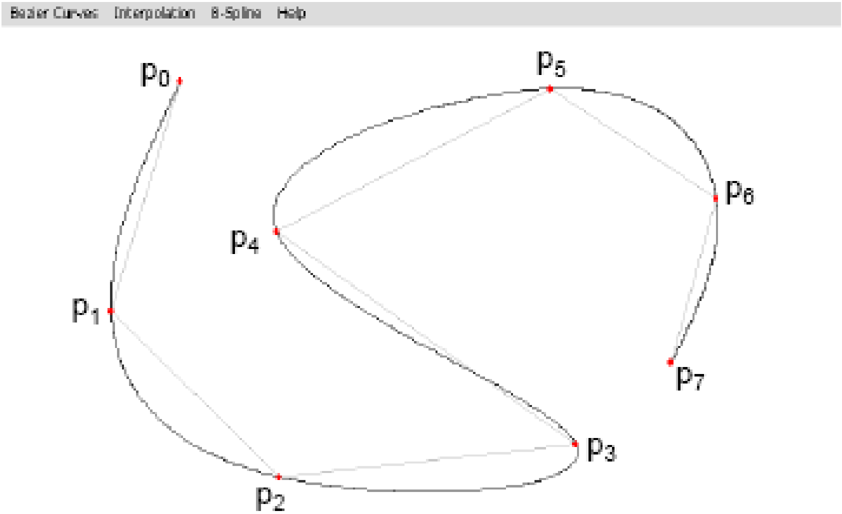
Representação de Curvas

B.1. Formas paramétricas

$$P(t) = (x(t), y(t))$$

Exemplos:

Curvas de Bézier



$$P(t) = \Sigma B_{n,i}(t)\, p_i \quad \text{onde } B_{n,i}(t) = \begin{bmatrix} n \\ i \end{bmatrix} t^i - (1-t)^{n-i}$$

Combinação convexa de p_i

Representação de Curvas

B.2. Formas não-paramétricas

Não há parâmetros e *uma coordenada* da curva é dada em função da outra, ou seja

$y = f(x)$ ou $x = f(y)$

– Exemplos:

1) Equação de um semi-círculo de raio 2

$$y = \sqrt{2^2 - x^2} \qquad ou \qquad x = \sqrt{2^2 - y^2}$$

2) Equação de uma reta

$y = 2x - 1$ ou $x = \frac{1}{2}(y + 1)$

Representação de Curvas

B.2.1 Forma *não-paramétrica explícita* : É dada por uma equação do tipo $y = f(x)$, ou seja uma das coordenadas é explicitamente dada em função das outras. Exemplos:

1) Equação genérica explícita de uma parábola:

$$y = ax^2 + bx + c$$

2) Equação de uma reta

$$y = mx + b$$

3) Polinômios:

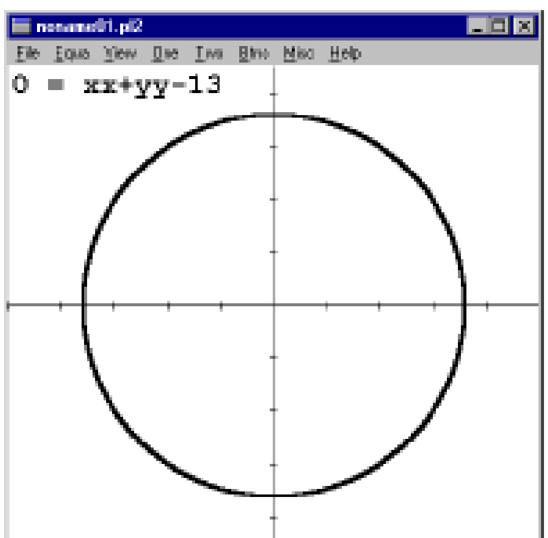
$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x^1 + a_0$$

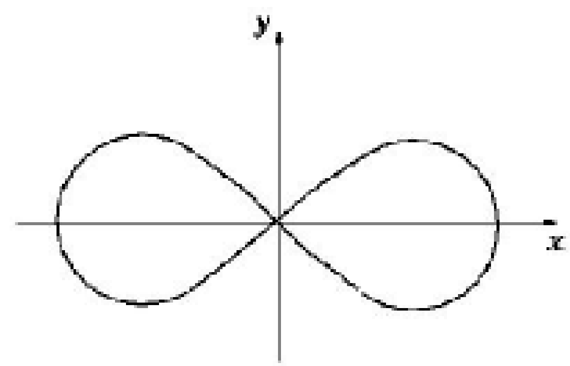
Representação de Curvas

- Obtém-se um valor de *y* para cada valor de *x* dado.

$$y = f(x) \quad \text{ou} \quad z = f(x,y)$$

Não permite representar correspondências sobrejetivas!





Lemniscate: $(x^2+y^2)^2 - (x^2-y^2) = 0$

Representação de Curvas

- B.2.2. A Forma *não-paramétrica implícita* não tem essa limitação. Nela as coordenadas são relacionadas por uma função. A sua forma é f(x,y)=0 ou f(x,y,z)=0.

- Exemplo: $x^2 + y^2 = R^{2,}$ $f(x,y)= x^2 + y^2 - R^2= 0$

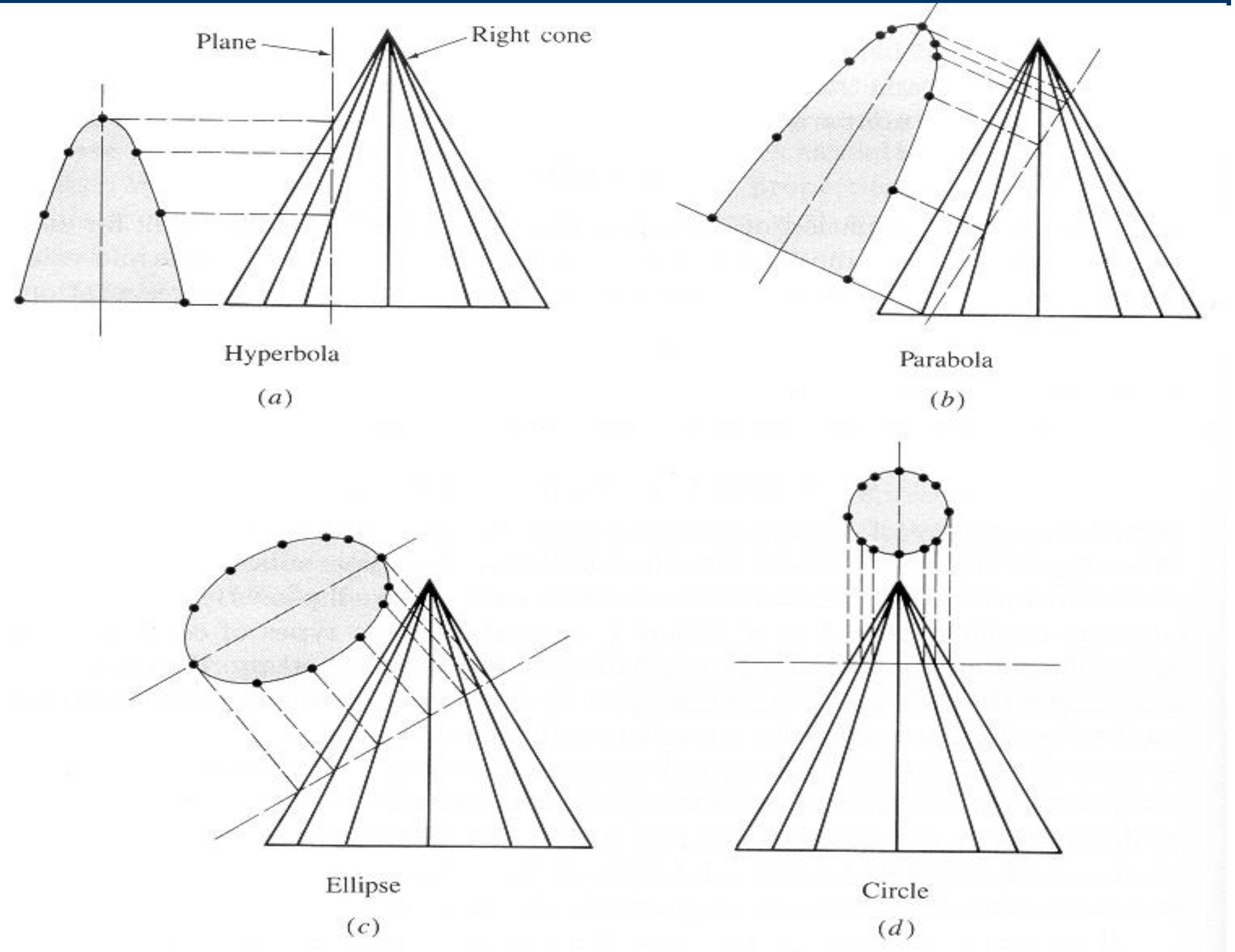
Representação de Curvas

- Exemplo: seções cônicas.

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

- Essa expressão representa a variedade de curvas planas denominadas seções cônicas. Essas curvas (cinco) são obtidas pelo corte de um cone por um plano, resultando em: círculo, elipse, parábola, hipérbole, reta.

Representação de Curvas



Representação de Curvas

Cônica	Forma Paramétrica	Forma Implícita
Elipse	$x = a \cos \theta$ $y = b \operatorname{sen} \theta$	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0$
Parábola	$x = at^2$, $y = 2at$	$y^2 - 4ax = 0$
Hipérbole	$x = a \cosh \theta$ $y = b \operatorname{senh} \theta$	$\frac{x^2}{a^2} - \frac{y^2}{b^2} - 1 = 0$

Exercício

Veja programas circle.cpp, parabola.cpp, helix.cpp

No programa circle.cpp uma circunferência de raio R foi modelada através da sua forma paramétrica. Acrescente com outra cor a circunferência também de raio R, modelada através da sua forma não paramétrica.

Curvas de Bézier

- É uma técnica de aproximação de curvas.
- Uma curva de Bézier pode ser gerada por 3, 4, até $n + 1$ pontos de controle (ajuste para um polinômio de grau n).
- Geralmente utiliza-se quatro pontos de controle (*forma cúbica*).
- A curva passa pelo primeiro e pelo último ponto de controle.

Curvas de Bézier

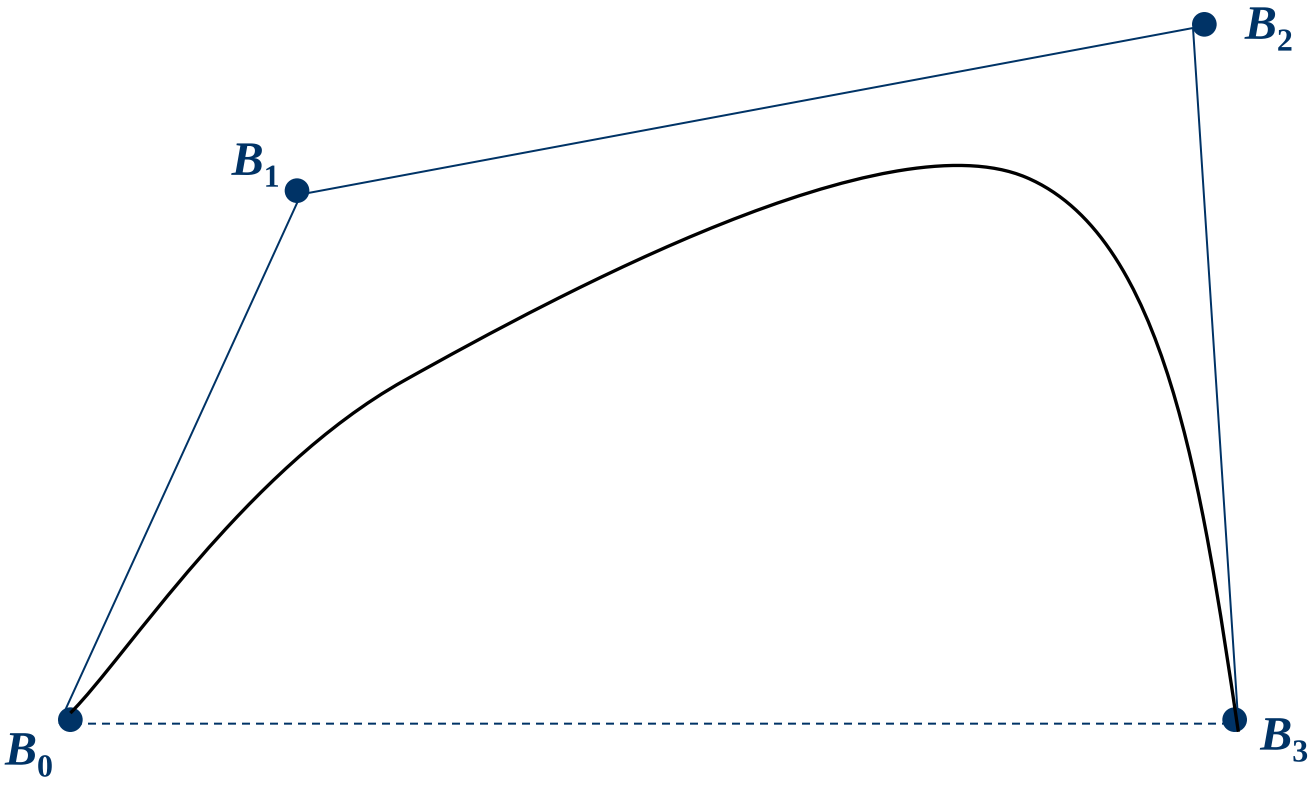


Figura 1

Curvas de Bézier

- A curva paramétrica de Bézier é definida como:

$$P(t) = \sum_{i=0}^{n} B_i J_{n,i}(t), \qquad 0 \le t \le 1$$

- Onde B_i representa cada um dos n+1 pontos de controle considerados e $J_{n,i}$ (t) são as funções que combinam a influência de todos os pontos (*blending functions*).

- Essas funções são descritas pelos polinômios de Bernstein como:

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

Curvas de Bézier

- onde n é o grau dos polinômios e:

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

(i = 0, 1, ..., n) são os coeficientes binomiais.

- Essas funções $J_{n,i}$ (t) devem satisfazer as condições: $J_{n,i}$ (t) ≥ 0 para todo i entre 0 e 1, isto é 0 ≤ t ≤ 1 e também:

$$\sum_{i=0}^{n} J_{n,i}(t) = 1, \qquad 0 \leq t \leq 1$$

Curvas de Bézier

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

n=2, vejamos os J para i=0,1,2

$$J_{2,0}(t) = \binom{2}{0} t^0 (1-t)^2 = (1-t)^2$$

$$J_{2,1}(t) = \binom{2}{1} t^1 (1-t)^1 = 2t(1-t)$$

$$J_{2,2}(t) = \binom{2}{2} t^2 (1-t)^0 = t^2$$

Curvas de Bézier

$$\sum_{i=0}^{n} J_{n,i}(t) = 1, \qquad 0 \le t \le 1$$

1 = t + (1-t)

$$(a+b)^2 = a^2 + 2ab + b^2$$

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

$$(t+(1-t))^2 = t^2 + 2t(1-t) + (1-t)^2$$

$$(t+(1-t))^3 = t^3 + 3t^2(1-t) + 3t(1-t)^2 + (1-t)^3$$

Curvas de Bézier

					1					
				1		1				
			1		2		1			
		1		3		3		1		
	1		4		6		4		1	
1		5		10		10		5		1

$$(a+b)^2 = a^2 + 2ab + b^2$$

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

$$(a+b)^5 = a^5 + 5a^4b + 10a^3b^2 + 10a^2b^3 + 5ab^4 + b^5$$

Curvas de Bézier

- Expressões que definem as curvas de Bézier:
  - Para três pontos de controle $\Rightarrow$ polinômios com grau 2.

$$P(t) = (1 - t)^2 B_0 + 2t(1 - t) B_1 + t^2 B_2,$$

onde *t* inicialmente é 0.

$$P(t) = [t^2 \quad t \quad 1] \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \end{bmatrix}$$

Curvas de Bézier

- Expressões que definem as curvas de Bézier:
  - Para quatro pontos de controle $\Rightarrow$ polinômios com grau 3.

$$P(t) = (1 - t)^3 B_0 + 3t\,(1 - t)^2 B_1 + 3t^2 (1 - t) B_2 + t^3 B_3,$$

onde t inicialmente é 0.

$$P(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 3 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

Curvas de Bézier - Algoritmo

Exercício:

(1) O programa superficies.cpp permite o ingresso interativo (pelo cliques do mouse) de n+1 pontos de controle e constrói a curva de Bézier de grau n, correspondente. Responda as seguintes questões:

a. Qual trecho de código constrói as funções de Bernstein (Jni)
b. Qual trecho de código constrói a curva de Bézier.
c. Em qual trecho de código a curva é desenhada
d. O programa funciona, mas pode ser melhorado. Identifique quais melhoras podem ser feitas.

Curvas de Bézier - Algoritmo

Exercício:

(2) Rode e entenda os programas bezierCurves.cpp e bezierCurveWithEvalMesh.cpp que desenham curvas de Bézier usando comandos de OpenGL. Responda as seguintes questões:

a. Qual trecho de código constrói as funções de Bernstein (Jni)
b. Qual trecho de código constrói a curva de Bézier.
c. Em qual trecho de código a curva é desenhada
d. O programa apresenta uma proposta de interação, você teria outra proposta.

Curvas de Bézier - Problemas

1. Falta de controle local : Uma alteração em um ponto no polígono de Bézier acarreta alterações em toda a curva de Bézier. Indesejável quando desejamos fazer ajustes finos.
2. O grau do polinômio cresce com o número de pontos de controle do polígono de controle.

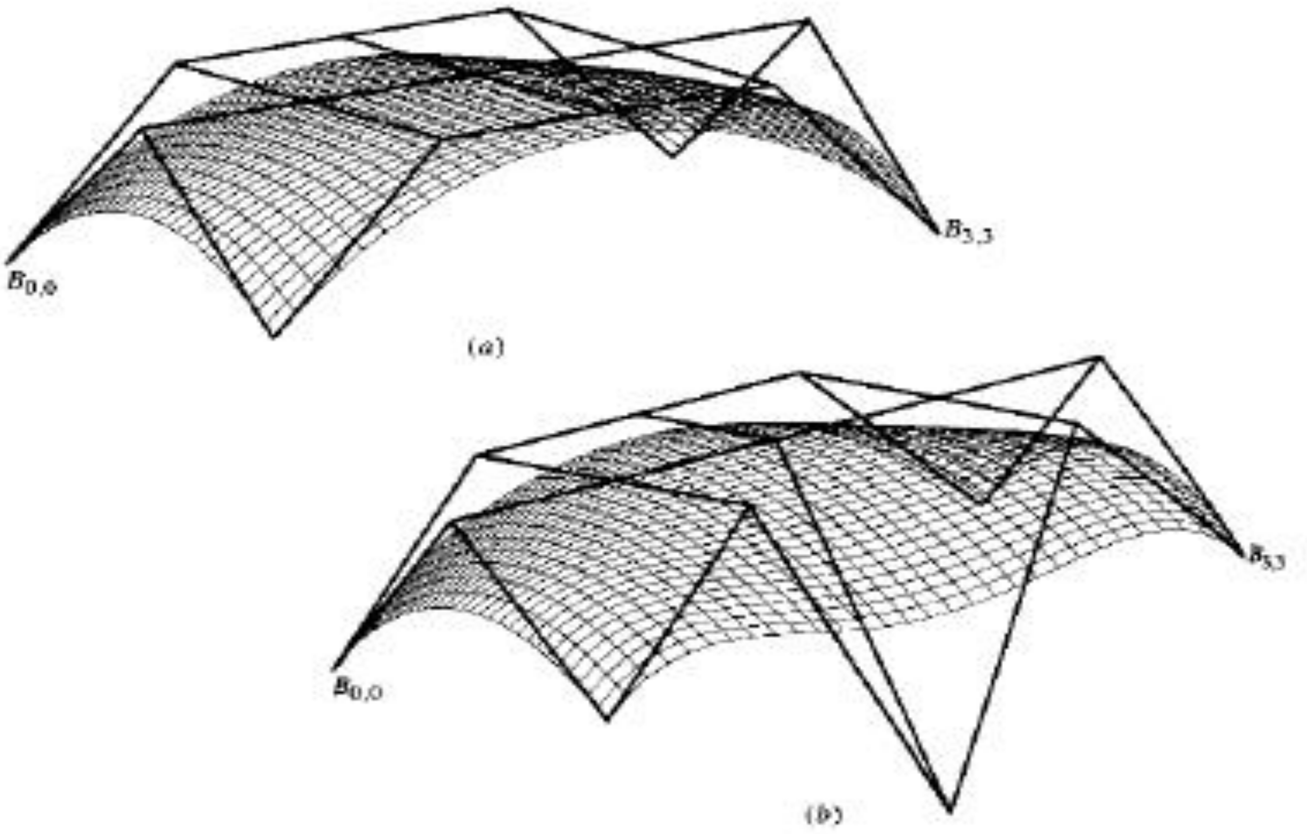
Superfícies Bézier

- Generalização da idéia de curva de Bézier.
- Sejam B_{ij}, $i=0,\ldots,m$, $j=0,\ldots,n$, um conjunto de pontos no R^3 de tal forma que sua projeção no plano *x0y* seja formada pelos vértices de *mn* retângulos de mesmas dimensões. A superfície de Bézier definida no domínio [0,1]x[0,1] é

$$Q(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{ij} J_{ni}(u) K_{mj}(v)$$

Onde J_{ni} e K_{mj} são os polinômios de Bernstein.

Superfícies Bézier



Curvas e Superfícies Bézier em OpenGL

1. Leia e entenda os programas bezierSurface.cpp e bezierCanoe.cpp que desenham superfícies de Bézier usando comandos de OpenGL. Explique os comandos que geram as superfícies em ambos programas.

Quádricas da GLU

A biblioteca GLU provê a renderização automática de objetos tridimensionais como esferas, cilindros e discos.

- Esferas:

gluSphere(GLUquadricObj *obj, GLdouble radius, GLint slices, Glint stacks)

- Cilindros:

gluCylinder(GLUquadricObj *obj, GLdouble baseRadius, Gldouble topRadius, GLdouble height, GLint slices, GLint stacks)

topRadius == zero, permite criar cone.

- Discos:

gluDisk(GLUquadricObj *obj, GLdouble innerRadius, GLdouble outerRadius, GLint slices, GLint loops)

innerRadius != zero, permite criar discos com furos.

Veja e rode o programa gluQuadrics.cpp da pasta Code

Modelagem usando Superfícies Bézier e primitivas

1. Leia e entenda o programa torpedo.cpp que desenha um torpedo composto de diferentes pedaços:

(i) Corpo: cilindro da GLU

(ii) Nariz: hemisferio

(iii) Três barbatanas: discos parciais da GLU

(iv) Disco traseiro : disco da GLU

(v) Haste da hélice: cilindro da GLU

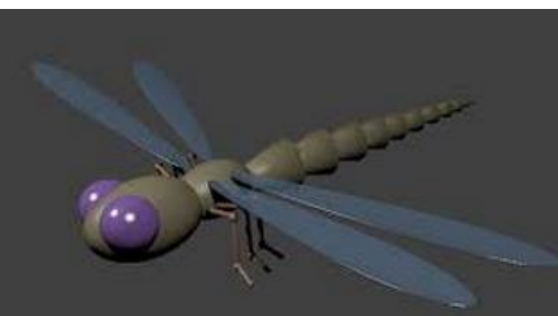
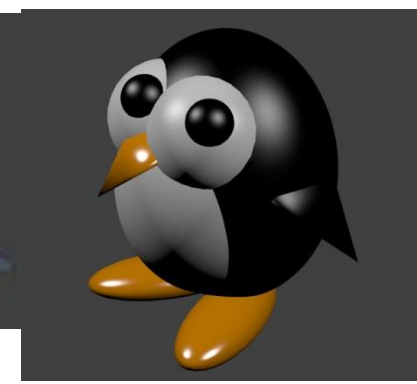
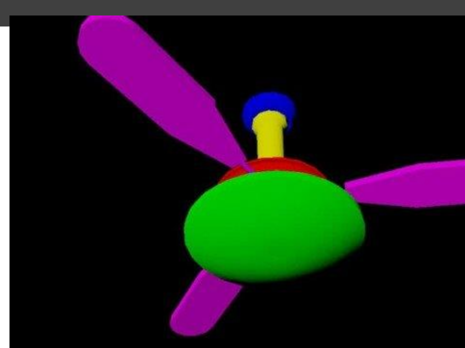
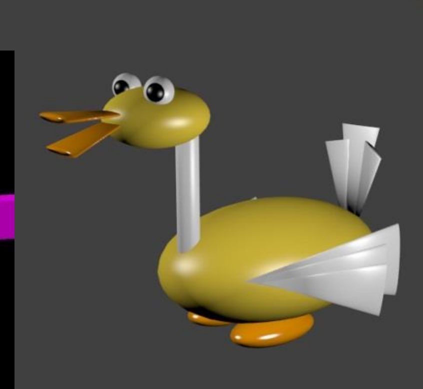
(vi) Três pás da hélice: pedaços bicúbicos Bezier

2. Desenhe um avião composto de vários pedaços. Use superfícies de Bézier e quádricas.

Superfícies Bézier e quádricas em OpenGL

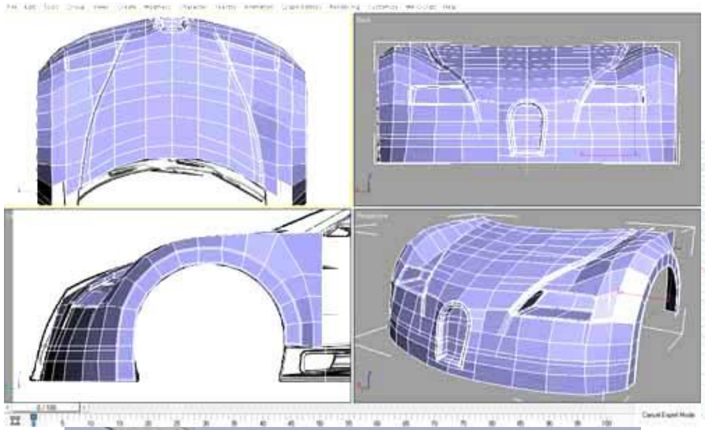
3. Modele os seguintes objetos.

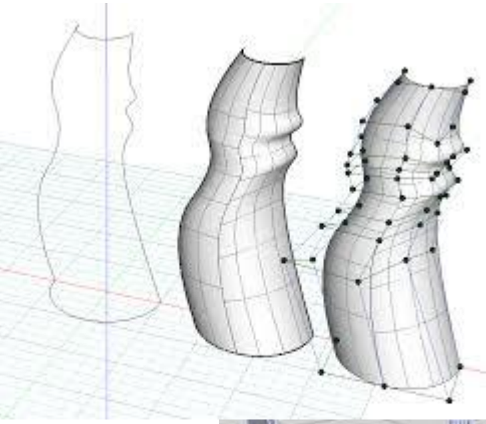


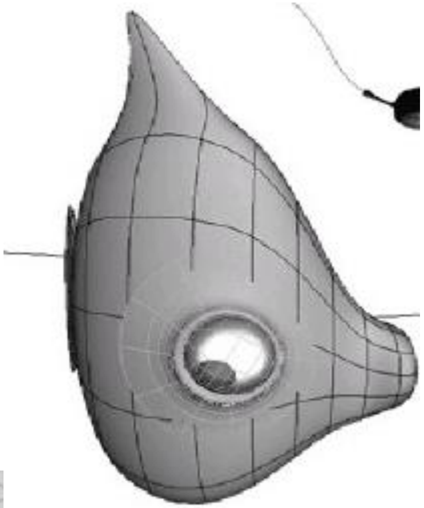





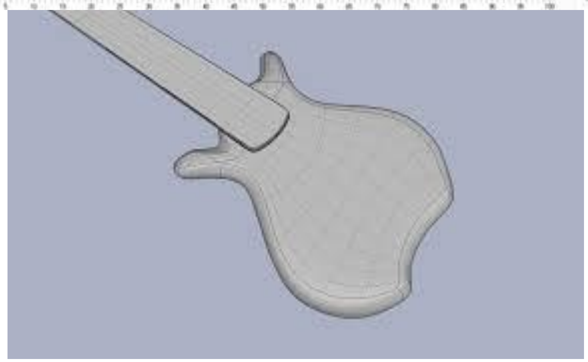
Superfícies Bézier em OpenGL

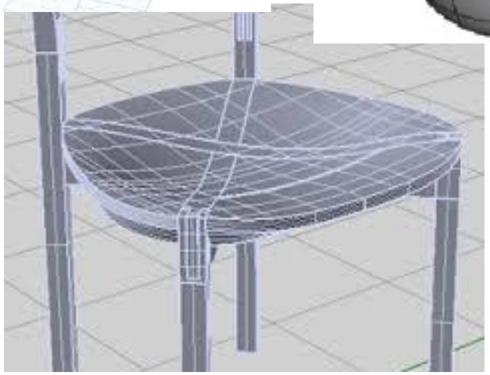
4. Modele os seguintes objetos: carro, vestido, peixe, violão, cadeira com uso de superfícies de Bézier.







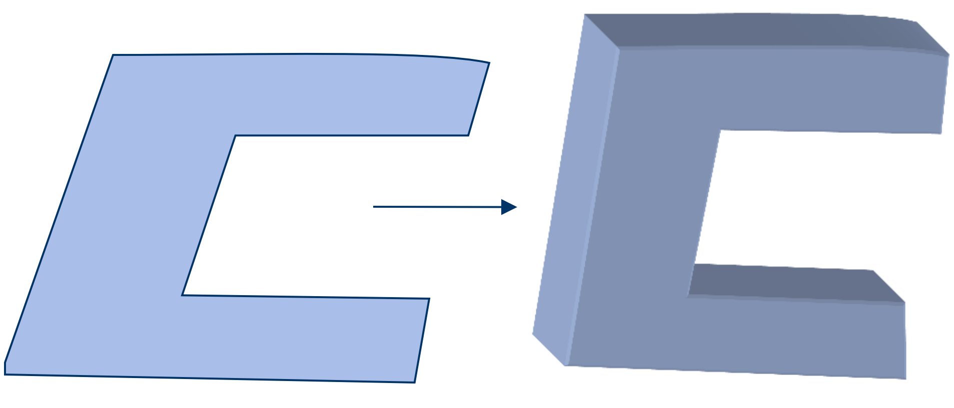


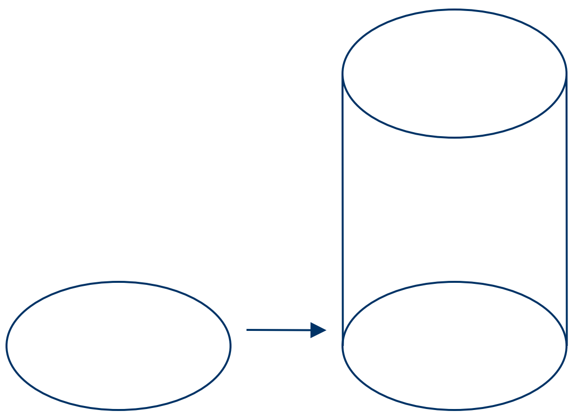


Varredura (*Sweeping*)

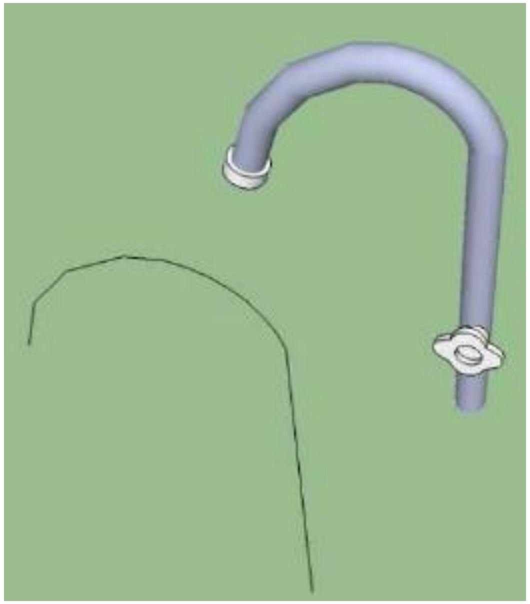
- Uma superfície é descrita quando uma curva C1 (curva geratriz) é deslocada no espaço, ao longo de uma trajetória dada por uma outra curva C2 (caminho ou diretriz).

  - Varredura translacional (Extrusão ou superfícies geradas por deslocamento)



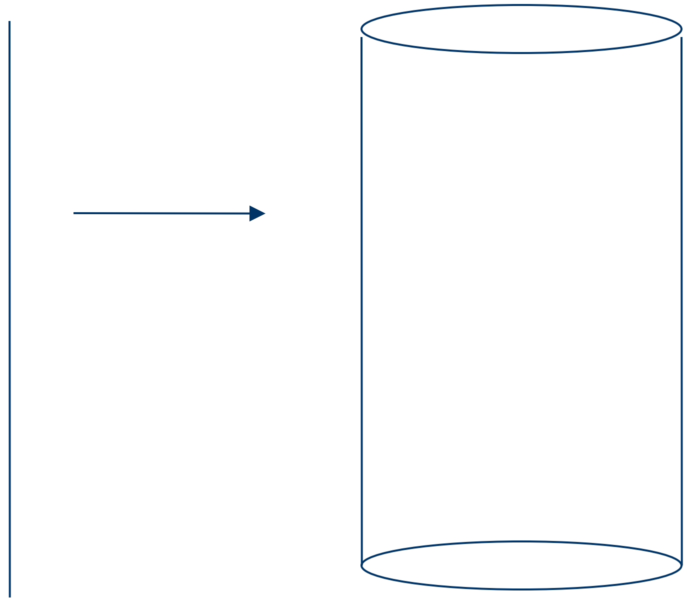


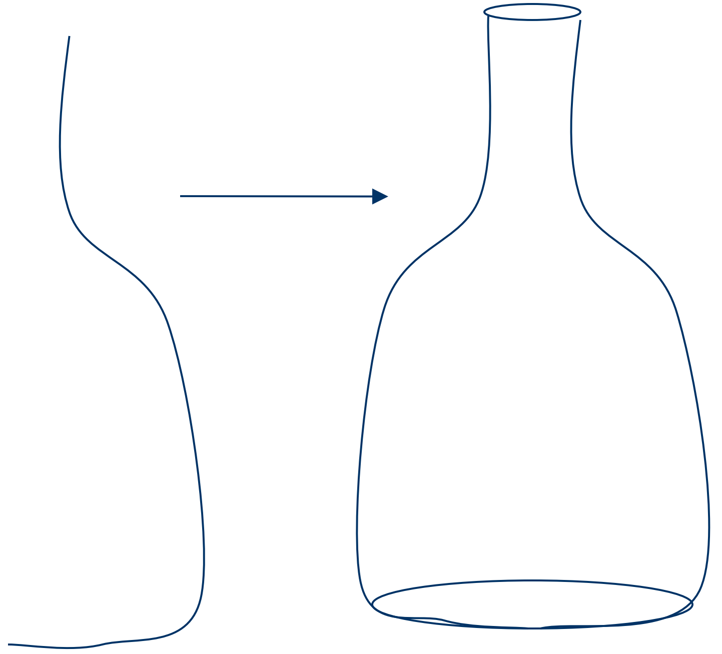
Varredura (*Sweeping*)



Varredura (*Sweeping*)

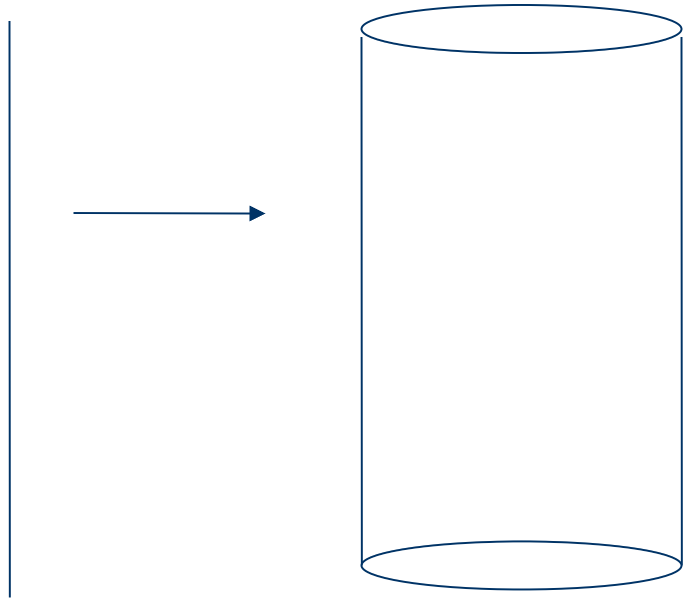
- Varredura rotacional (ou superfícies de revolução)

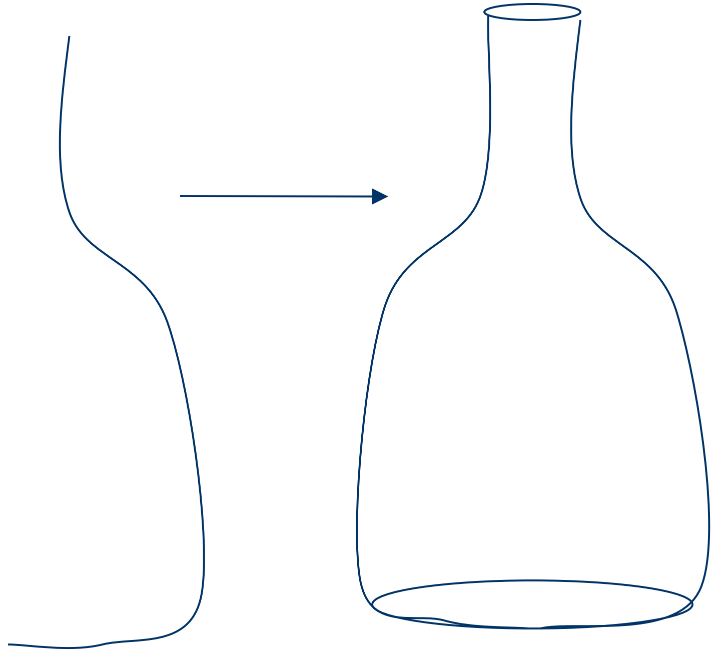




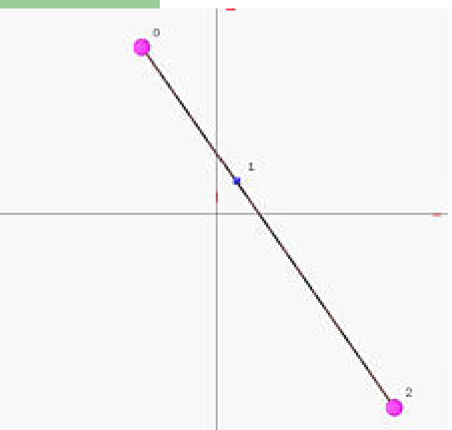
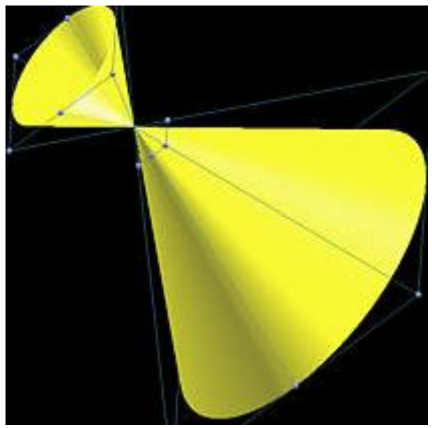
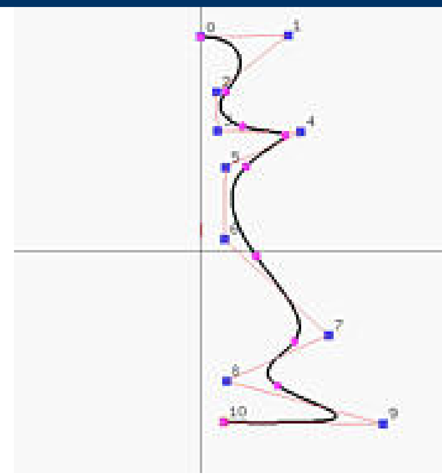
Varredura (*Sweeping*)

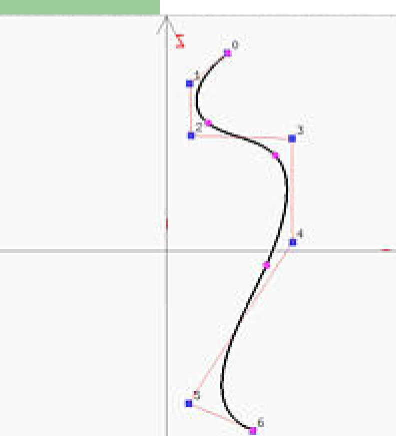
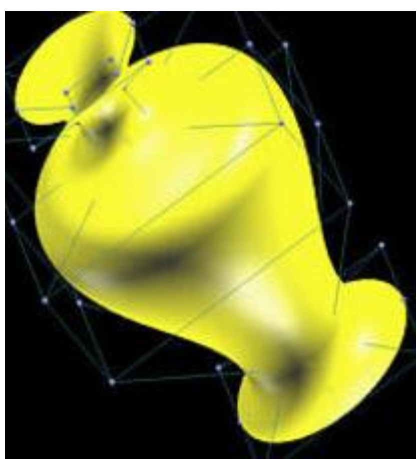
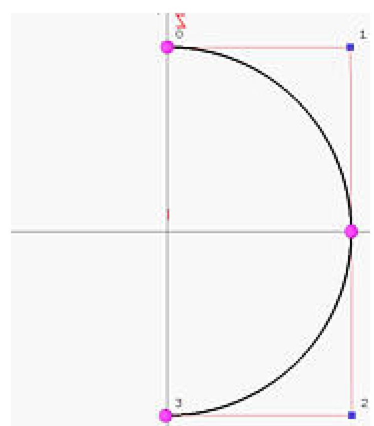
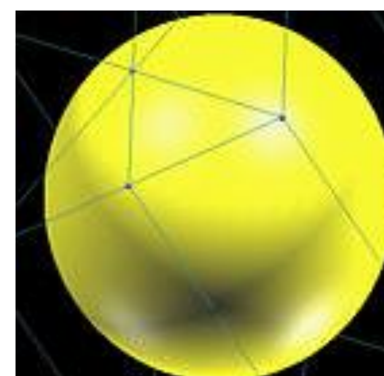
- Varredura rotacional (ou superfícies de revolução)





Varredura (*Sweeping*)

Superfície de Revolução - Tarefa

O executável swprj.exe permite desenhar uma curva geratriz que dará origem a uma superfície de revolução

O programa torus.c (disponível no site da disciplina) desenha a superfície de revolução chamada torus.

O programa superfícies.cpp permite desenhar uma curva de Bézier como curva geratriz e a partir dela obter uma superfície de revolução.

1. Veja que a superfície em superfícies.cpp é representada apenas por cortes horizontais, como você representaria ela por polígonos (quadriláteros ou triângulos)?

Superfície de Revolução - Tarefa

2. Compare o programa torus.cpp da pasta Code com o programa torus.c disponibilizado no site da disciplina. Para consulta de comandos novos use o livro RedBook disponibilizado também no site da disciplina.
3. Desenhar um cilindro como superfície de revolução usando um segmento de reta como curva geratriz.