

Notas de Aula de Laboratório de curvas e superfícies

Asla Medeiros e Sá e alunos da turma 2017

February 13, 2018

1 Introdução

A visualização dos conceitos formalizados em geometria diferencial podem ser de grande auxílio didático e também para a dar suporte "concreto" à intuição geométrica de conceitos como curvatura e torção. O propósito destas notas de aula é o de compilar o material abordado nas aulas de laboratório computacional da disciplina de curvas e superfícies. O intuito é documentar o que foi abordado em laboratório para consulta futura.

2 Ambientes gráficos

Para o desenho de curvas e também de superfícies, se faz necessário lançar mão de ferramentas computacionais especializadas para esta finalidade. Para curvas planas temos a disposição muitas opções. A primeira classe de ferramentas da qual podemos lançar mão são as calculadoras gráficas. Neste caso optamos por explorar o uso da calculadora gráfica Desmos (<https://www.desmos.com/>). Outra classe de ferramentas são os ambientes de geometria dinâmica, dentre os quais, o Geogebra (<https://www.geogebra.org/>) é de longe o mais popular.

Utilizamos tanto o Desmos quanto o Geogebra com a finalidade de explorar conceitos como parametrização de uma curva. Ao passar para a visualização de curvas no espaço, quisemos visualizar os vetores tangente, normal e binormal, para esclarecer a intuição do que vem a ser a curvatura e a torção de uma dada curva no espaço. Neste momento lançamos mão do Geogebra 3D para fazer diversas visualizações.

Ao estudarmos curvas cuja descrição é mais complexa, percebemos a vantagem de se trabalhar em ambientes computacionais que tratam as equações que definem as curvas de forma simbólica. Em tais ambientes, operações como derivação são executadas simbolicamente, proporcionando clareza aos códigos e evitando erros de cálculo e de transcrições de fórmulas, no caso de a derivação ser feita em um ambiente - WolframAlpha por exemplo - e o resultado ser transcrito para um ambiente de desenho.

Para o desenho de superfícies o ambiente do Geogebra 3D foi utilizado para os casos mais simples. Uma sugestão interessante de software que permite explorar famílias de superfícies de forma eficiente é o k3DSurf (<http://k3dsurf.sourceforge.net/>). Seguimos então com a abordagem simbólica para produção de desenhos de superfícies em Matlab.

Passaremos agora a descrever alguns exercícios que foram feitos no decorrer do curso, definindo os conceitos abordados no exercício e mostrando a implementação que foi feita para produção das imagens.

3 Curvas Planas

Aa seguintes curvas possuem o mesmo traço: $f(x; y) \in R^2 : x^2 + y^2 = 1$:

$$\alpha(t) = (\cos(t); \sin(t)); t \in [0, 2];$$

$$\beta(t) = (\cos(2t); \sin(2t)); t \in [0, 2]$$

No entanto, a velocidade de β é o dobro da velocidade de α . No mesmo período de tempo, a curva β percorre duas vezes a circunferência e a curva α só uma vez. Quando falamos de velocidade, ver a dinâmica do objeto de estudo em movimento é de grande utilidade cognitiva.

Pela simplicidade do exemplo proposto, vamos observar a dinamica do movimento inserindo as equações das curvas α e β no ambiente da calculadora gráfica Desmos.

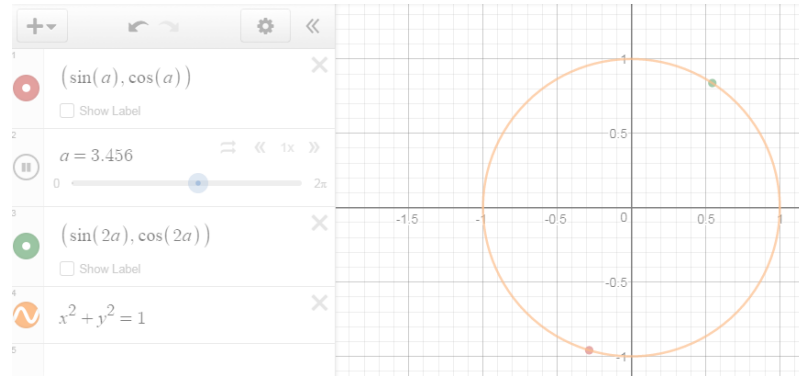


Figure 1: Exemplo de traço de circunferência no ambiente Desmos.

Como parametrizamos um segmento de uma parábola? Fazer desenho.

Como parametrizamos uma circunferência? Mostrar o vetor velocidade. Repetir o exemplo acima no geogebra.

3.1 Ambiente de geometria dinâmica: Geogebra

3.2 Matlab

4 Ambiente Simbólico do Matlab

5 Curvas no espaço

5.1 Geogebra

5.2 Matlab

First you have to upload the image file from your computer using the upload link the project menu. Figure 2 example.

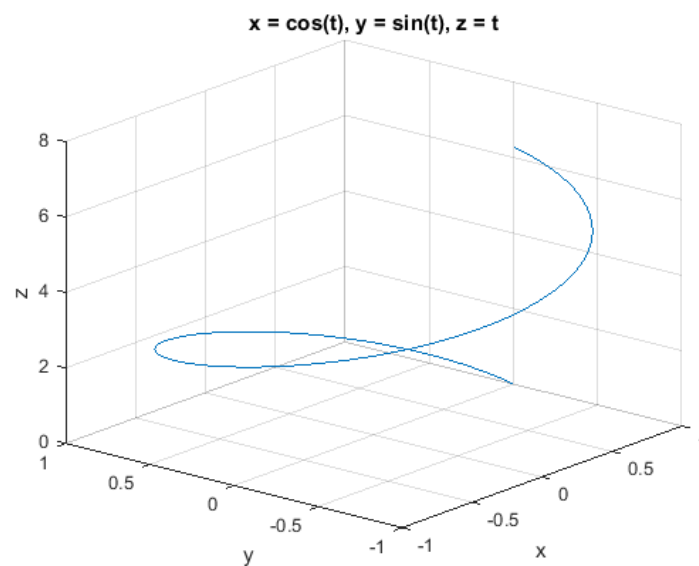


Figure 2: Exemplo de curva no espaço.

5.3 Triedro de Frenet(Rafael Katz)

As fórmulas de Frenet–Serret descrevem as propriedades cinemáticas de uma partícula que se move ao longo de uma curva contínua e diferenciável, num espaço euclidiano tridimensional \mathbb{R}^3 , ou as propriedades geométricas da própria curva independentemente do movimento.

Dada: $\alpha : I \rightarrow \mathbb{R}^3$ parametrizada pelo comprimento de arco sem pontos singulares, definimos o triedro de Frenet como:

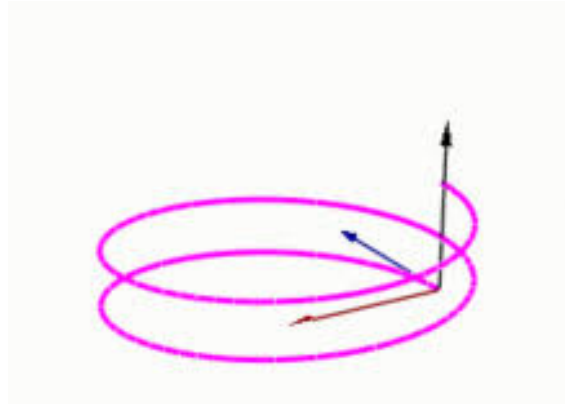
$$\{t(s); n(s); b(s)\}$$

Onde t é o vetor unitário tangente à curva, n é a derivada de t , e b é o produto vetorial de t e n .

Chamamos de fórmulas de Frenet-Serret o conjunto de equações

$$\begin{cases} t'(s) = \kappa(s)n(s), \\ n'(s) = -\kappa(s)t(s) - \tau(s)b(s), \\ b'(s) = \tau(s)n(s) \end{cases}$$

Figure 3: Triedro de Frenet

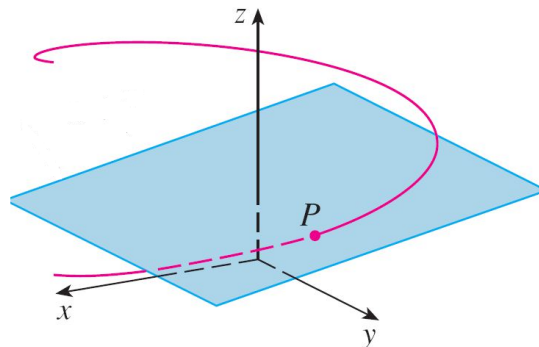


5.4 Plano Osculador(Rafael Katz)

O plano osculatório (gerado por T e N) é um plano em um espaço euclidiano ou espaço afim que atende uma subvariedade em um ponto de forma a ter uma segunda ordem de contato no ponto. A palavra osculate é do osculatus latino que é um particípio passado de osculari, que significa beijar. Um plano osculatório é, portanto, um plano que "beija" uma subvariedade.

O plano osculatório na geometria das curvas espaciais euclidianas pode ser descrito em termos das fórmulas Frenet-Serret como o espaço vectorial gerado dos vetores tangente e normal

Figure 4: Plano Osculador



Um exemplo de um Plano Osculador feito em Matlab:

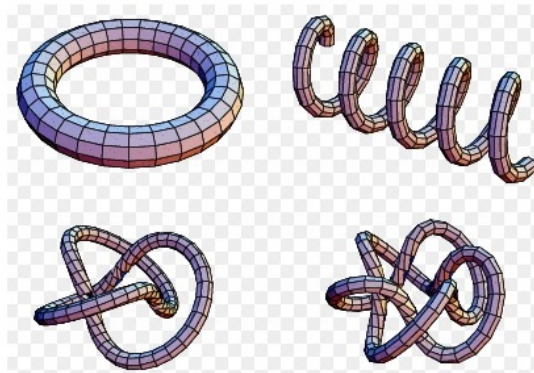
```
1 syms t;  
2 a = 1;  
3 r = 1;  
4 figure(1);  
5 alpha1 = [t, t^2, t^3];  
6 ezplot3(alpha1(1), alpha1(2), alpha1(3), [0, 2*pi, 0.01, pi]);  
7 axis([-5 5 -5 5]);  
8 for k = 0:0.1:2  
9     alpha1 = [t, t^2, t^3];  
10    ezplot3(alpha1(1), alpha1(2), alpha1(3), [0, k*pi, 0.01, pi]);  
11    axis([-5 5 -5 5]);  
12    pause(0.05);  
13 end  
14 hold on  
15 [x y] = meshgrid(-1:0.1:1); % Generate x and y data  
16 z = 6*(2-x)+3*(y-4)+8; % Solve for z data  
17 surf(x,y,z) %Plot the surface  
18 hold off
```

6 Superfícies

6.1 Tubos (Pedro, Henrique e Matheus)

Intuição gráfica

Antes de entrarmos nos detalhes formais da definição de *Tubos*, gostaríamos de dar uma intuição geométrica do conceito. As figuras abaixo, retiradas do site *Wolfram*, são exemplos de diferentes tipos de tubos:



A primeira das quatro figuras, em que um tubo é construído ao redor de um círculo e que lembra a forma do doce *donut*, é chamada de *Toro*.

Definição Formal

Ao longo de todo o curso usamos o livro *Apontamentos de Geometria Diferencial* do português Jorge Picado. Nesta obra, o autor define os tubos da seguinte forma:

Seja: $\gamma : (\alpha, \beta) \rightarrow \mathbb{R}^3$ uma curva parametrizada por comprimento de arco, para a qual existe um raio $r > 0$ tal que $k(s) < r^{-1}$ para qualquer $s \in (\alpha, \beta)$. A circunferência

$$\theta \rightarrow \cos\theta N(s) + \sin\theta B(s)$$

está no plano normal à curva em $\gamma(s)$, plano este perpendicular a tangente à curva em $\gamma(s)$. Quando esta circunferência se move ao longo de γ define uma superfície, chamada tubo de raio $r > 0$ em torno de γ , parametrizada por

$$\sigma(s, \theta) = \gamma(s) + r(\cos\theta N(s) + \sin\theta B(s));$$

com $s \in (\alpha; \beta)$, $\theta \in (0, 2\pi)$ ou $s \in (\alpha, \beta)$, $\theta \in (-\pi, \pi)$.

Implementação computacional da função tubo

Implementamos a definição matemática acima em Matlab com o seguinte código:

```
1 function tubo(curva, inter, r)
2     syms theta
3     t = diff(curva)
4     Naux = diff(t);
5     N = Naux/norm(Naux);
6     B = cross(N, t);
7     x_E = curva(1) + sqrt(r)*cos(theta)*N(1) + sqrt(r)*sin(theta)*B(1);
8     y_E = curva(2) + sqrt(r)*cos(theta)*N(2) + sqrt(r)*sin(theta)*B(2);
9     z_E = curva(3) + sqrt(r)*cos(theta)*N(3) + sqrt(r)*sin(theta)*B(3);
10    fsurf(x_E, y_E, z_E, [0, 2*pi, inter(1), inter(2)]);
11    camlight
12    axis equal
13 end
```

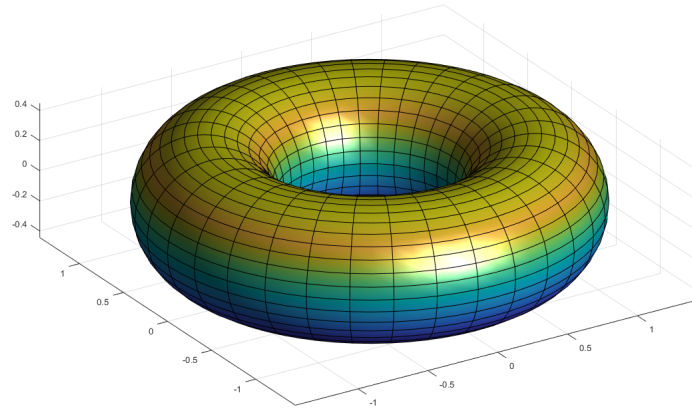
Há de ser ressaltado que essa função não é um caso particular e sim uma implementação geral. Dessa forma, ela funciona para qualquer curva parametrizada por comprimento de arco.

Exemplos

No caso do toro temos:

```
1 syms v
2 f = [cos(u), sin(u), 0]
3 tubo(f, [0, 2*pi], 0.2)
```

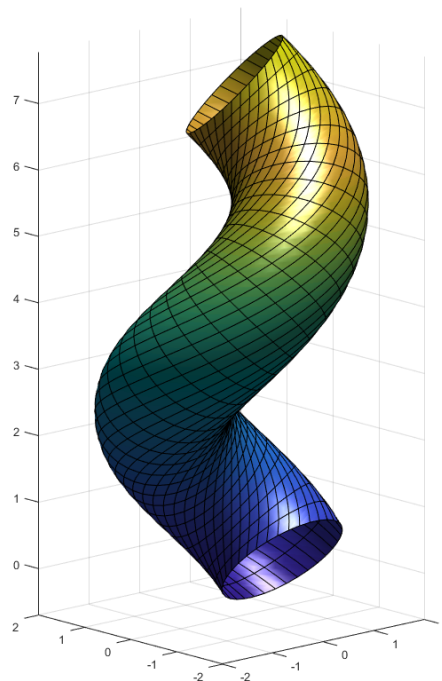
Figure 5



Outro exemplo interessante é o caso do tubo em formato de espiral. Este pode ser construído da seguinte forma:

```
1 syms v
2 f = [cos(v/sqrt(2)) sin(v/sqrt(2)) v/sqrt(2)]
3
4 tubo(f,[0,10],1)
```

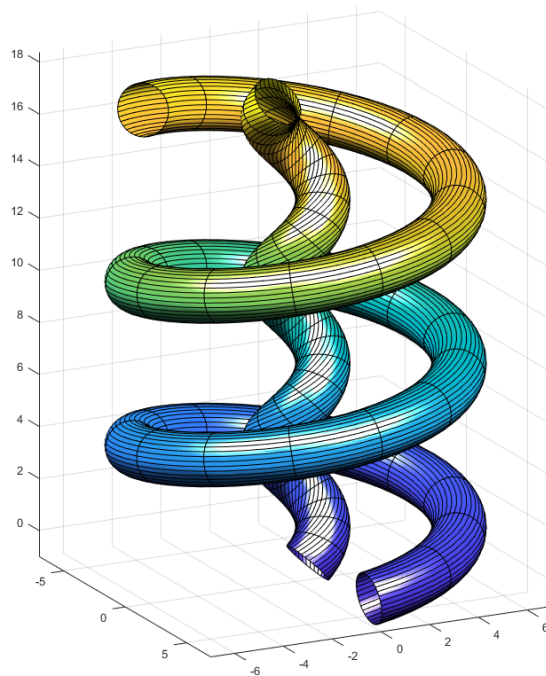
Figure 6



Por fim, cabe citar a implementação com dois tubos em formato de espiral e que possuem raios distintos:

```
1 syms v u
2
3 f = [6*cos(u/sqrt(37)) 6*sin(u/sqrt(37)) u/sqrt(37)]
4
5 g = [cos(u/sqrt(2)) sin(u/sqrt(2)) u/sqrt(2)]
6
7 tubo(f, [0 100], 1)
8 hold on
9 tubo(g, [0 25], 1)
```

Figure 7



6.2 Conchas (Bibi e Fernanda)

As conchas são um caso particular dos tubos. Ambas as superfícies são geradas a partir de um círculo ou elipse cujo centro se desloca ao longo de uma curva, porém, no caso de conchas, **o raio varia ao longo da curva**.

Segundo Picado¹, uma concha é resultado do deslocamento de uma curva C de centro P (a curva geratriz, geralmente uma elipse) ao longo de uma espiral helicoidal H (a curva estrutural), onde $P \in H$. A curva C pode ser determinada a partir dos vetores normal, N , e binormal, B , de H no ponto P :

$$C = P(t) + aN\cos(\theta) + bB\sin(\theta),$$

onde θ é o ângulo de rotação de um ponto $Q \in C$ no plano formado pelos eixos da elipse e t é o parâmetro da espiral helicoidal H .

Na natureza, podemos observar a manifestação dessas figuras nos moluscos. Essa é uma das características mais marcantes desses animais, servindo para a classificação dos mesmos em sub-grupos. Vejamos alguns exemplos:

A partir de uma elipse como curva geratriz C , podemos gerar conchas como na figura abaixo. Essa concha, chamada *Planorbis*, também dá nome a uma família de gastrópodes, os *Planorbidae*².

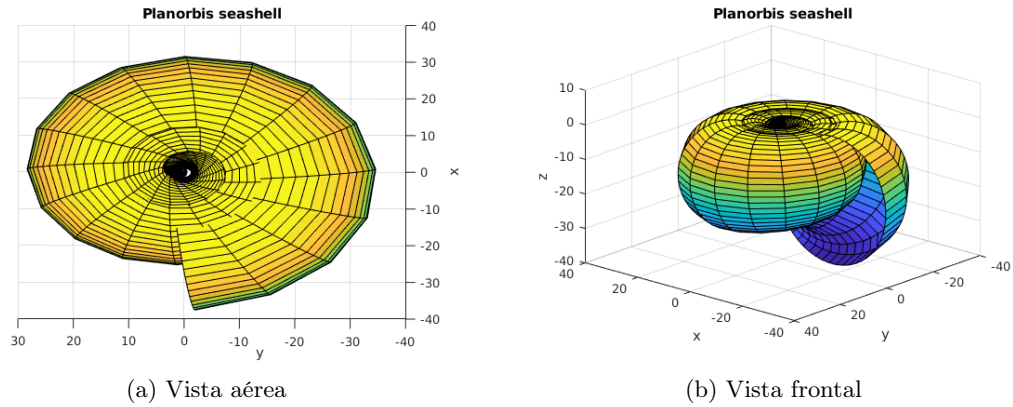


Figure 8: Concha Planorbis (fóssil)

```

1  %== Helice espiral ==%
2
3  syms t theta
4
5  x1 = t*cos(t);
6  y1 = t*sin(t);
7  z1 = t;
8
9  %ezplot3(x1,y1,z1, [-10*pi, 0])
10
11
12 %== Concha gerada a partir da helice ==%
13
14 dx1 = diff(x1);
15 dy1 = diff(y1);
16 dz1 = diff(z1);
17
18 % Vetor tangente a helice em (x1, y1, z1)
19 T = [dx1, dy1, dz1];
20
21 d2x1 = diff(dx1);

```

¹A beleza matemática das conchas marinhas. Jorge Picado

²Planorbidae. Wikipédia, a enciclopédia livre.


```

22 d2y1 = diff(dy1);
23 d2z1 = diff(dz1);
24
25 nx1 = d2x1/norm([d2x1,d2y1,d2z1]);
26 ny1 = d2y1/norm([d2x1,d2y1,d2z1]);
27 nz1 = d2z1/norm([d2x1,d2y1,d2z1]);
28
29 % Vetor normal a helice em (x1, y1, z1)
30 N = [nx1,ny1,nz1];
31
32 %Vetor binormal a helice em (x1, y1, z1)
33 B = cross(N,T);
34
35 bx1 = B(1);
36 by1 = B(2);
37 bz1 = B(3);
38
39 %Pontos da elipse de centro (x1, y1, z1)
40 X = x1 + (t*cos(theta)*nx1 + sin(theta)*bx1);
41 Y = y1 + (t*cos(theta)*ny1 + sin(theta)*by1);
42 Z = z1 + (t*cos(theta)*nz1 + sin(theta)*bz1);
43
44 ezsurf(X, Y, Z, [-6*pi, 0, 0, 2*pi])
45 title('Planorbis seashell');

```

Um outro tipo de concha pode ser encontrada em moluscos da classe *Bivalve*, que dão o nome à concha³. Ela é constituída de duas partes iguais, um exemplo de formato de cada uma das partes é o da figura abaixo. Neste caso, a curva geratriz C é uma circunferência, cujo raio varia ao longo de H .

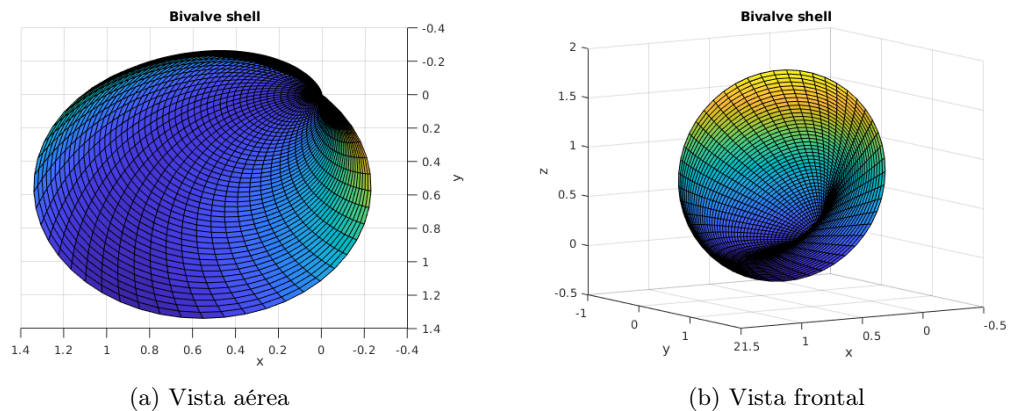


Figure 9: Concha Bivalve

```

1 %== Helice espiral ==%
2
3 syms t theta
4
5 x1 = t*cos(t);
6 y1 = t*sin(t);
7 z1 = t;
8
9 %ezplot3(x1,y1,z1, [0, 6*pi])

```

³Bivalvia. Wikipédia, a enciclopédia livre.

```

10
11
12 %==== Concha gerada a partir da helice ====
13
14 dx1 = diff(x1);
15 dy1 = diff(y1);
16 dz1 = diff(z1);
17
18 %Vetor tangente a helice em (x1, y1, z1)
19 T = [dx1, dy1, dz1];
20
21 d2x1 = diff(dx1);
22 d2y1 = diff(dy1);
23 d2z1 = diff(dz1);
24
25 nx1 = d2x1/norm([d2x1, d2y1, d2z1]);
26 ny1 = d2y1/norm([d2x1, d2y1, d2z1]);
27 nz1 = d2z1/norm([d2x1, d2y1, d2z1]);
28
29 %Vetor normal a helice em (x1, y1, z1)
30 N = [nx1, ny1, nz1];
31
32 %Vetor binormal a helice em (x1, y1, z1)
33 B = cross(N,T);
34
35 bx1 = B(1);
36 by1 = B(2);
37 bz1 = B(3);
38
39 %Pontos da circunferencia de centro (x1, y1, z1)
40 X = x1 + t*(cos(theta)*nx1 + sin(theta)*bx1);
41 Y = y1 + t*(cos(theta)*ny1 + sin(theta)*by1);
42 Z = z1 + t*(cos(theta)*nz1 + sin(theta)*bz1);
43
44 ezsurf(X, Y, Z, [0, pi/4, 0, 2*pi]);
45 title('Bivalve shell')

```

O formato de C e H nos permite gerar os mais variados tipos de conchas. Alguns exemplos do Picado⁴, utilizando também rotações da elipse C em torno de seus próprios eixos, ilustram essa diversidade.

Tonel, oliva e concha cavalo:

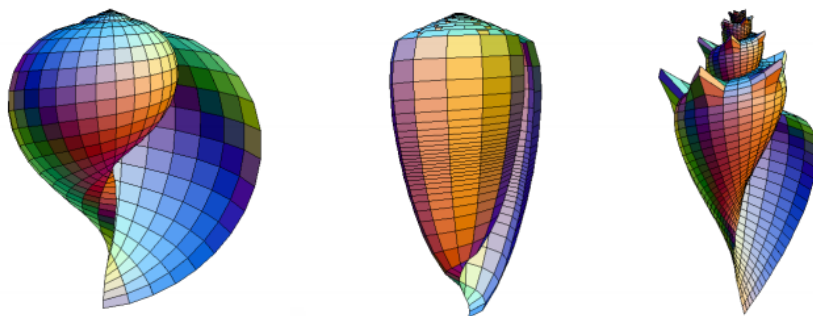


Figure 10: Outros exemplos de conchas

⁴A beleza matemática das conchas marinhas. Jorge Picado

6.3 Superfícies Regradas (Paula e Carla, Fifi)

Uma superfície regradada S é uma superfície que em cada ponto $p \in S$, existe um segmento de reta contido em S passando por p .

Observação: Uma forma de obter superfícies regradadas é tomar uma parametrização da forma $f(u, v) = g(u) + v.h(u)$, onde g e h são curvas diferenciáveis.

1. A função $f(u, v) = (\cos(u), \sin(u), 1) + v(0, 0, 1)$ define uma superfície regradada.

```
1 syms t u
2
3 x_E = cos(t);
4 y_E = sin(t);
5 z_E = 1+u;
6
7 ezsurf(x_E, y_E, z_E, [-pi, pi, -1, 1])
8 title('f(u,t)=(cos(t),sin(t),1)+u(0,0,1)')
```

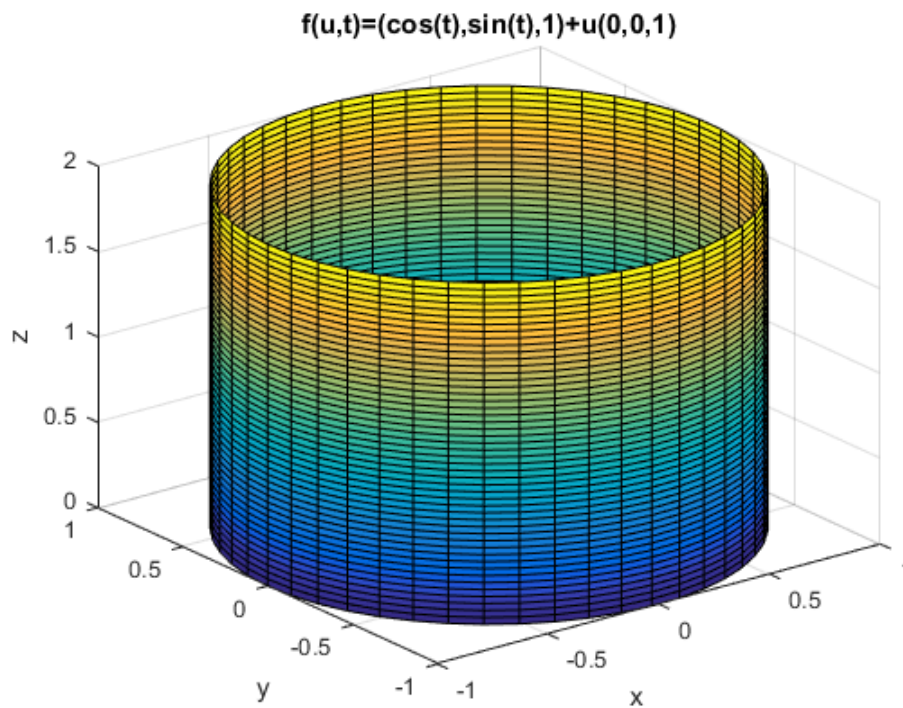


Figure 11: Superfície similar a um bracelete

2. O helicóide definido por $f(u, v) = g(u) + v.h(u)$ onde $g(u) = (\cos(u), \sin(u), u)$ e $h(u) = (-\cos(u), -\sin(u), 0)$, é uma superfície regradada.

```
1 %helicóide
2
3 syms t u
4
5 x_E = u*sin(t);
6 y_E = u*cos(t);
7 z_E = t;
8
9 ezsurf(x_E, y_E, z_E, [-pi, pi, -1, 1])
10 title('Helicóide')
```

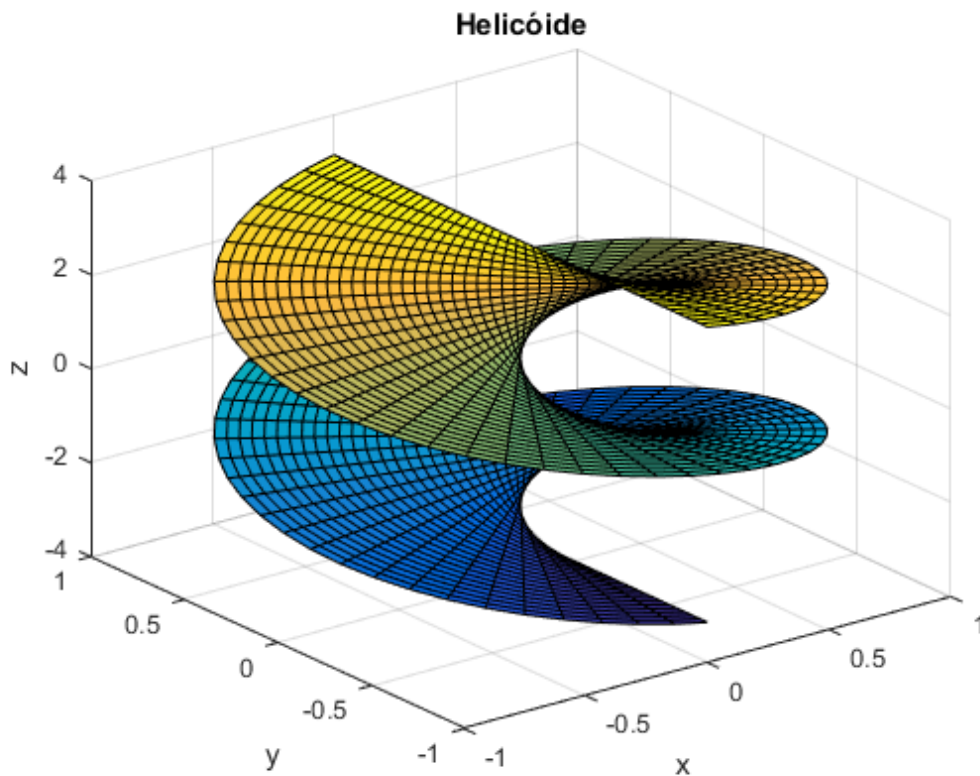


Figure 12: Helicóide

3. Outra superfície regrada conhecida é a superfície obtida a partir da curva de Viviani. Em Matemática, particularmente Geometria, a curva de Viviani, também conhecida como janela de Viviani, é uma curva no espaço figura em forma de oito nomeada em homenagem ao matemático italiano Vincenzo Viviani, a intersecção de uma esfera com um cilindro que é tangente à esfera e passa através do centro da esfera.

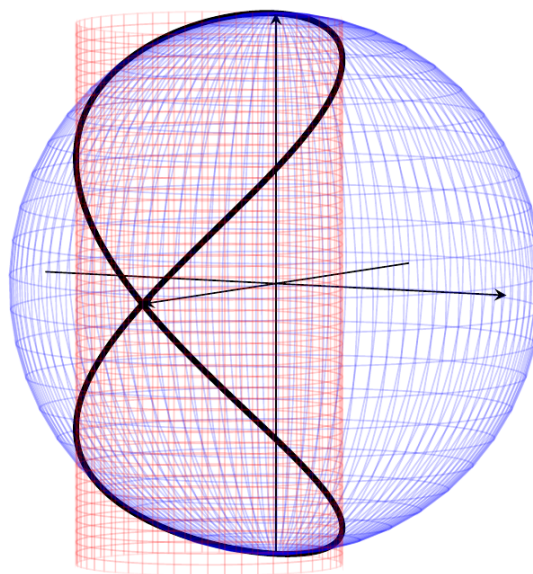


Figure 13: Curva de Viviani

A parametrização da superfície é dada pela seguinte equação:

$$f(u, t) = (1 + \cos(t), \sin(t), 2\sin(\frac{t}{2})) + u(-\sin(t), \cos(t), \cos(\frac{t}{2}))$$

```

1 syms t u
2
3 x = (1 + cos(t) - u*sin(t));
4 y = (sin(t) + u*cos(t));
5 z = (2*sin(t/2) + u*cos(t/2));
6
7 ezsurf(x, y, z, [0, 4*pi, -4, 4])
8 title('Curva de Viviani')

```

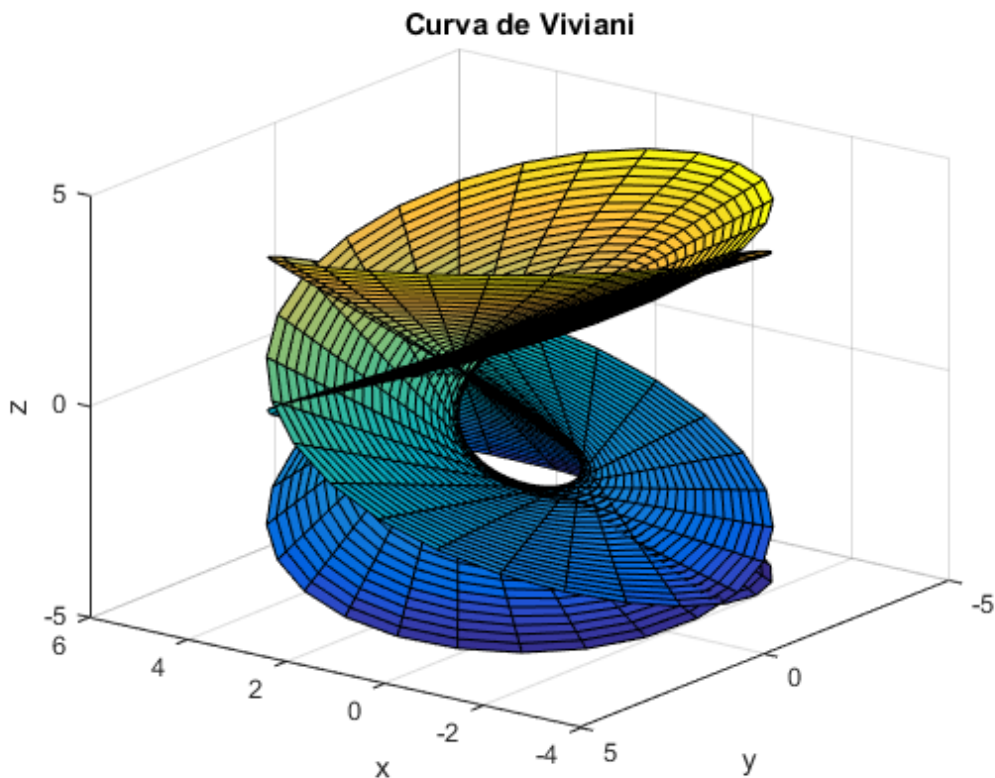


Figure 14: Curva de Viviani

4. O guarda-chuva Whitney (às vezes chamado de guarda-chuva Cayley) é uma superfície auto-intersectada colocada em três dimensões. É a união de todas as linhas retas que passam por pontos de uma parábola fixa e são perpendiculares a uma linha reta fixa, paralelamente ao eixo da parábola e deitada no seu plano de divisão perpendicular (biseção/bissetriz). A parametrização é dada por:

$$f(u, t) = (t^2, 0, 0) + u(0, \frac{1}{\sqrt{1+t^2}}, \frac{t}{\sqrt{1+t^2}})$$

```

1 syms t u
2
3 x = t^2;
4 y = u/(1+t^2)^(1/2);
5 z = t*u/(1+t^2)^(1/2);
6
7 ezsurf(x, y, z, [-1, 1, -1, 1])
8 title('Guarda-chuva de Whitney')

```

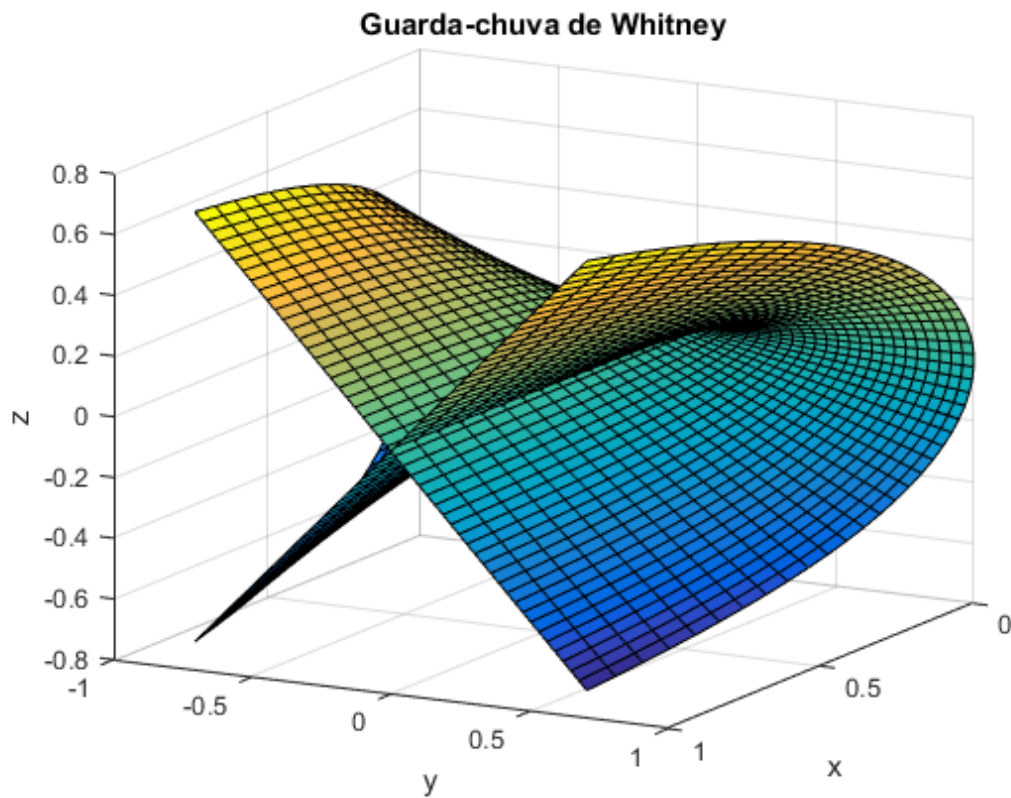


Figure 15: Guarda-chuva de Whitney

6.4 Superfícies de Revolução (Rafael Katz, Rafael Paya e Emerson Lai)

Uma superfície de revolução é uma superfície gerada por uma rotação de uma curva plana ou cônica, chamada geratriz, em torno de um dos eixos ou em torno de uma reta fixa nesse plano, chamado de eixo de revolução. Por exemplo, a esfera e o toro são superfícies de revolução representado nas imagens abaixo.

Figure 16: Esfera

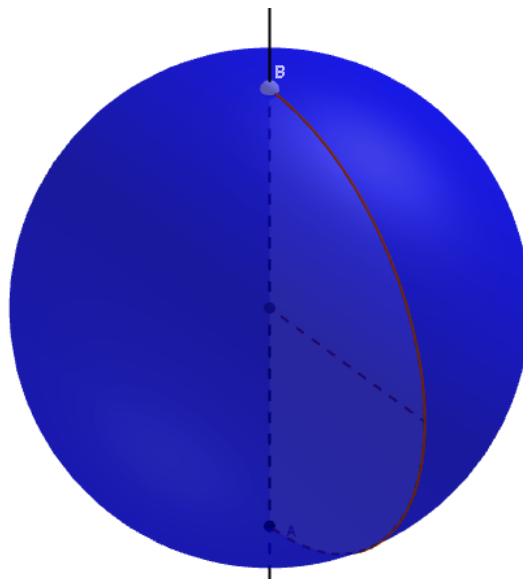
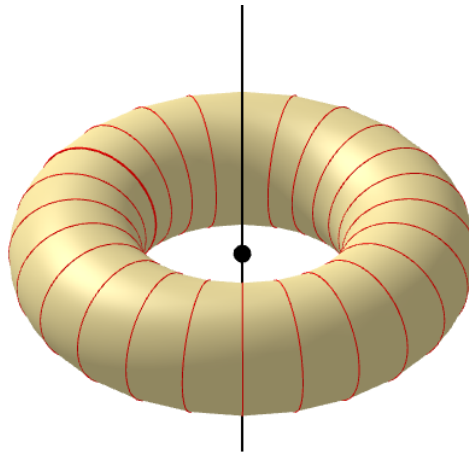


Figure 17: Toro



Construção dessas figuras usando Matlab

Esfera

```
1 syms u v;
2 a = 1;
3 figure(1);
4 alpha1 = [a*cos(u)*sin(v), a*sin(u)*sin(v), a*cos(v)];
5 ezsurf(alpha1(1), alpha1(2), alpha1(3), [0, 2*pi, 0.01, pi]);
6 axis([-1 1 -1 1]);
7 for k = 0:0.1:2
8     alpha1 = [a*cos(u)*sin(v), a*sin(u)*sin(v), a*cos(v)];
9     ezsurf(alpha1(1), alpha1(2), alpha1(3), [0, k*pi, 0.01, pi]);
10    axis([-1 1 -1 1]);
11    pause(0.05);
12 end
```

toro

```
1 syms u v;
2 R = 3;
3 r = 2;
4 colormap spring;
5 figure(1);
6 alpha1 = [(R+r*cos(u))*cos(v), r*sin(u), (R+r*cos(u))*sin(v), r*sin(u)];
7 ezsurf(alpha1(1), alpha1(2), alpha1(3), [0, 2*pi, 0.01, 2*pi]);
8
9 axis([-5, 5, -5, 5, -5, 5])
10
11
12 for k = 0:0.1:2
13     alpha1 = [(R+r*cos(u))*cos(v), (R+r*cos(u))*sin(v), r*sin(u)];
14     ezsurf(alpha1(1), alpha1(2), alpha1(3), [0, k*pi, 0.01, 2*pi]);
15     axis([-5, 5, -5, 5, -5, 5]);
16     pause(0.05);
17 end
```

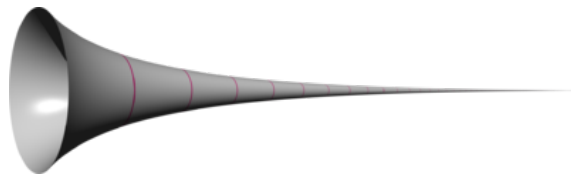

Cone Duplo

```
1 syms u v;
2 h = 3;
3 r = 2;
4 colormap spring;
5 figure(1);
6 alpha1 = [(h-v)/h*r*cos(u), (h-v)/h*r*sin(u), v];
7 ezsurf(alpha1(1), alpha1(2), alpha1(3), [0, 2*pi, 0.01, 2*pi]);
8
9 axis([-5, 5, -5, 5, -5, 5])
10
11
12 for k = 0:0.1:2
13     alpha1 = [(h-v)/h*r*cos(u), (h-v)/h*r*sin(u), v];
14     ezsurf(alpha1(1), alpha1(2), alpha1(3), [0, k*pi, 0.01, 2*pi]);
15     axis([-5, 5, -5, 5, -5, 5]);
16     pause(0.05);
17 end
```

Gabriel's Horn

Trombeteira de Gabriel, é uma superfície de revolução que se obtém girando a curva $y = x^{-1}$, com $x \in [1, \infty)$, em torno do eixo das abscissas. Tal construção tem a característica de possuir uma superfície com área infinita, envolvendo um volume finito.

Figure 18: trompeta de gabriel



```
1 syms u v;
2 a = 3;
3 colormap spring;
4 figure(1);
5 alpha1 = [u, a*cos(v)/u, a*sin(v)/u];
6 ezsurf(alpha1(1), alpha1(2), alpha1(3), [0, 2*pi, 0.01, 2*pi]);
7
8 axis([-5, 5, -5, 5, -5, 5])
9
10
11 for k = 0:0.1:2
12     alpha1 = [u, a*cos(v)/u, a*sin(v)/u];
13     ezsurf(alpha1(1), alpha1(2), alpha1(3), [0, k*pi, 0.01, 2*pi]);
14     axis([-5, 5, -5, 5, -5, 5]);
15     pause(0.05);
16 end
```

7 Comentários e bibliografia L^AT_EX

Como colocar comentários (útil para o ambiente colaborativo do overleaf):

Comments can be added to your project by clicking on the comment icon in the toolbar above. To reply to a comment, simply click the reply button in the lower right corner of the comment, and you can close them when you're done.

You can upload a `.bib` file containing your BibTeX entries, created with JabRef; or import your [Mendeley](#), CiteULike or Zotero library as a `.bib` file. You can then cite entries from it, like this: [\[Gre93\]](#). Just remember to specify a bibliography style, as well as the filename of the `.bib`.

You can find a [video tutorial here](#) to learn more about BibTeX.

We hope you find Overleaf useful, and please let us know if you have any feedback using the help menu above — or use the contact form at <https://www.overleaf.com/contact>!

References

- [Gre93] George D. Greenwade. The Comprehensive Tex Archive Network (CTAN). *TUGBoat*, 14(3):342–351, 1993.