Christopher Lum
lum@uw.edu

# Lecture 04b
# Scalar Functions, Vector Functions, and Vector Derivatives

The YouTube video entitled 'Scalar Functions, Vector Functions, and Vector Derivatives' that covers this lecture is located at https://youtu.be/haJVEtLN6-k .

# Outline

-Scalar Functions
-Vector Functions
-Vector Derivatives

# Scalar Functions

We can define functions which map from $\mathbb{R}^n$ to $\mathbb{R}$. These are sometimes referred to as scalar functions as the output of them are scalars.

$$f : \mathbb{R}^n \to \mathbb{R}$$

In many engineering applications we consider $n = 2$ (the function can be visualized with a planar, 2D plot) or $n = 3$ (requiring a 3D plot for visualization). For more information on 3D plotting see the following videos:

'3D Plotting in Matlab' at https://youtu.be/OUwfE_-tcfo
'3D Plotting in Mathematica' at https://youtu.be/s_ehZc5N7Lg

**Example: 2D Scalar Function**

Consider the population density of group of animals in a rectangular region to be defined as

$$p(x, y) = 3\,x^2 + y^2$$

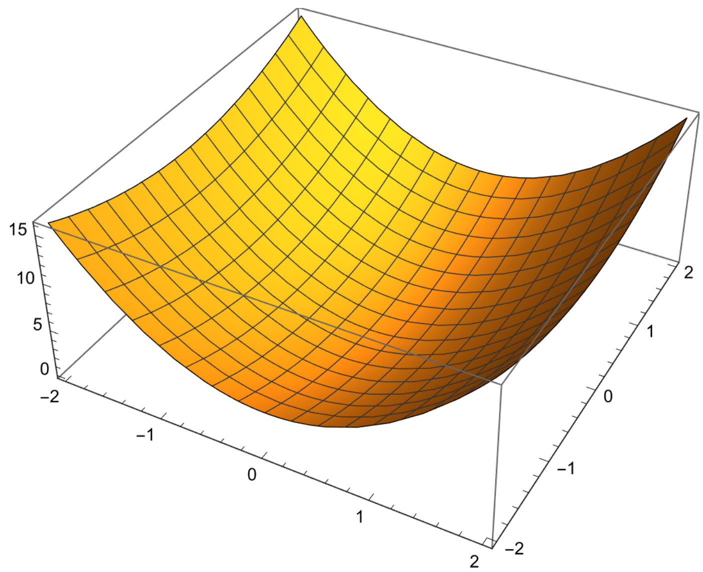
We see that this function assigns a scalar value (the population of animals) to each 2-element input vector (an $x$, $y$ position).

In[∘]:= `p[x_, y_] = 3 x^2 + y^2;`

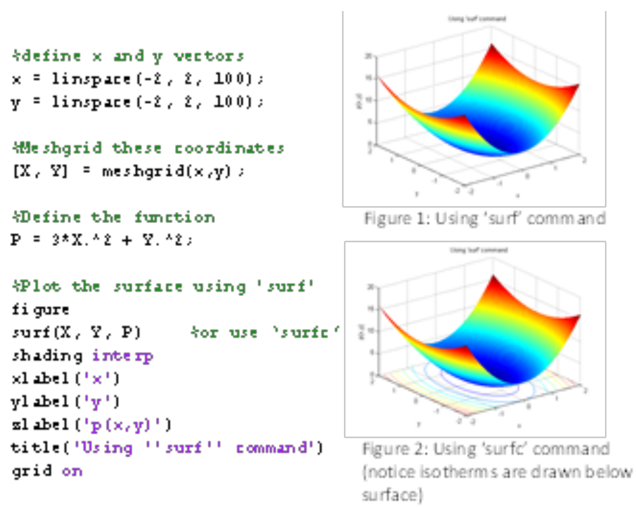
We can visualize this function using Mathematica's 'Plot3D' function. Note that in general, if Mathematica has a function that generates a 2D graphic, its equivalent in 3D simply has the suffix '3D'.

In[∘]:=
```
Plot3D[p[x, y], {x, -2, 2}, {y, -2, 2}]
Clear[p]
```

Out[∘]=




Matlab also provides the functions to create 3 dimensional plots of 2D scalar functions. Pseudo-code using 'surf', 'surfc', and the resulting output is shown below.




#### Example: 3D Scalar Function

Consider the temperature in Fahrenheit inside a 3D object to be described by

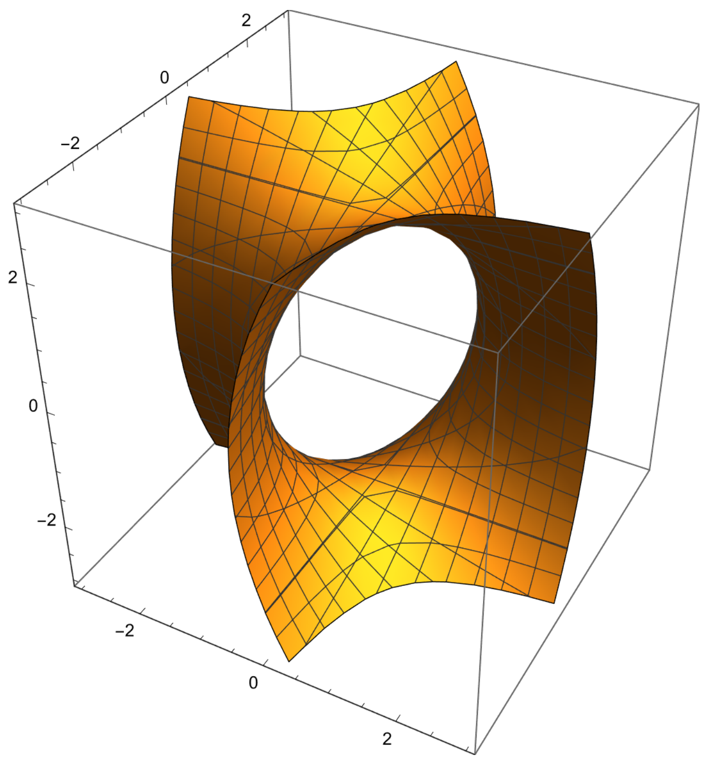
$$T(x, y, z) = 4\,x\,y\, + z^2$$

In[∘]:= `T[x_, y_, z_] = 4 x y + z^2;`

In this scenario, it becomes somewhat difficult to visualize this function because at each point in 3D space, we need to visualize a scalar value. Instead, it may be easier to visualize isotherms, or surfaces where $T(x, y, z)$ = constant. To do this, Mathematica provides a function 'ContourPlot3D' which can be used to visualize these isotherm surfaces. For example, to visualize the surface where the temperature is $5^o\,F$, we can use

In[ ]:=
```
ContourPlot3D[T[x, y, z] == 5, {x, -3, 3}, {y, -3, 3}, {z, -3, 3}]
```

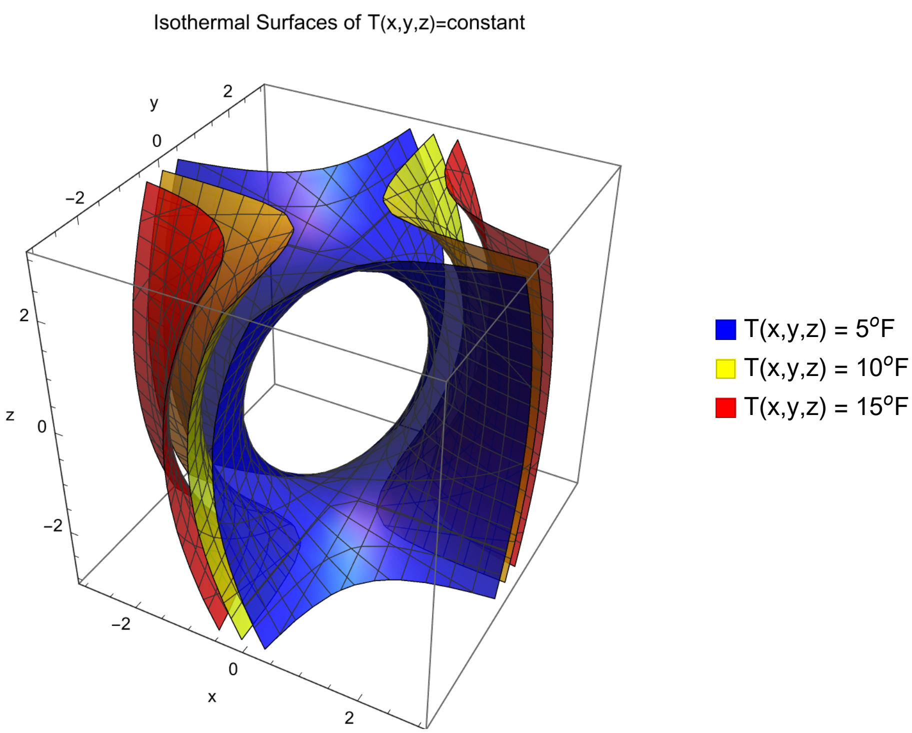
Out[ ]=




We can use multiple plots to show different temperatures

In[∘]:=
```
Legended[
 Show[
  (*T=5°F*)
  ContourPlot3D[T[x, y, z] == 5, {x, -3, 3}, {y, -3, 3}, {z, -3, 3},
   ContourStyle → Directive[Blue, Opacity[0.8], Specularity[White, 30]]
  ],

  (*Plot 2*)
  ContourPlot3D[T[x, y, z] == 10, {x, -3, 3}, {y, -3, 3}, {z, -3, 3},
   ContourStyle → Directive[Yellow, Opacity[0.8], Specularity[White, 30]]
  ],

  (*Plot 3*)
  ContourPlot3D[T[x, y, z] == 15, {x, -3, 3}, {y, -3, 3}, {z, -3, 3},
   ContourStyle → Directive[Red, Opacity[0.8], Specularity[White, 30]]
  ],

  (*Plot Options*)
  PlotLabel → "Isothermal Surfaces of T(x,y,z)=constant",
  AxesLabel → {"x", "y", "z"}
 ],

 (*Add legend information*)
 SwatchLegend[{Blue, Yellow, Red},
  {"T(x,y,z) = 5°F", "T(x,y,z) = 10°F", "T(x,y,z) = 15°F"}]
]
```

Out[ ]=




In[ ]:= **Clear[T]**

# Vector Functions

We can also define functions which map from $\mathbb{R}^2$ to $\mathbb{R}^2$ or from $\mathbb{R}^3$ to $\mathbb{R}^3$. These are sometimes referred to as vector functions or vector fields.

$f : \mathbb{R}^2 \to \mathbb{R}^2$ (planar vector function)

$f : \mathbb{R}^3 \to \mathbb{R}^3$ (3D vector function)

These are referred to as vector functions because their outputs are vectors instead of scalars.

**Example: 2D Vector Function (Velocity Field)**

Consider the vector field described by

$$\overline{v} = 50\cos(x)\,\hat{i} + x\,y^2\,\hat{j}$$

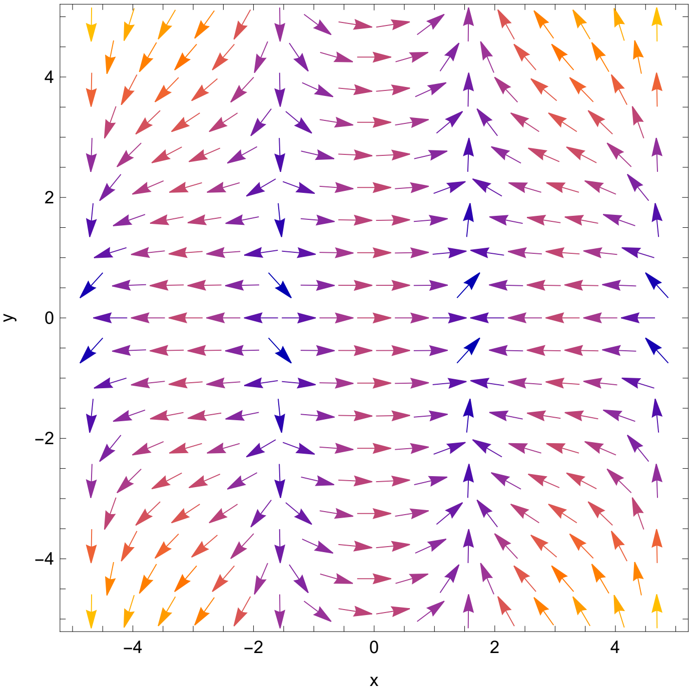
In[ ]:= $\mathbf{v[x\_,\ y\_]} = \begin{pmatrix} \mathbf{50\,Cos[x]} \\ \mathbf{x\,y^2} \end{pmatrix};$

In this case, the function assigns a 2-element vector, $\overline{v}$, for each input vector (also a 2-element vector).

For example we can physically interpret the input vector $(x \quad y)^T$ as a 2D position and the output vector, $\overline{v} = (v_x \quad v_y)^T$ as the velocity of a fluid at this location $(x \quad y)^T$.

Mathematica provides the ‘VectorPlot’ function to visualize 2D vector fields

In[ ]:=
```
VectorPlot[{v[x, y][[1, 1]], v[x, y][[2, 1]]}, {x, -5, 5}, {y, -5, 5},
 FrameLabel → {"x", "y"}]
```

Out[ ]=



Matlab also provides the function' quiver' to produce similar output. Pseudo-code and Matlab output is shown below

```
%define the points where we
%would like to draw the vectors
x = linspace(-5, 5, 15);
y = linspace(-5, 5, 15);

%Meshgrid these coordinates
[X, Y] = meshgrid(x,y);

%Define the function
U = 50*cos(X);
V = X.*Y.^2;

%Plot the vector field using 'quiv
figure
quiver(X, Y, U, V)
xlabel('x')
ylabel('y')
title('Using ''quiver'' command')
grid on
```

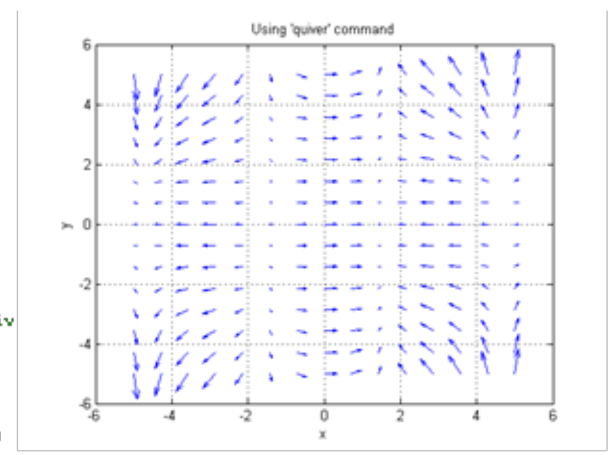
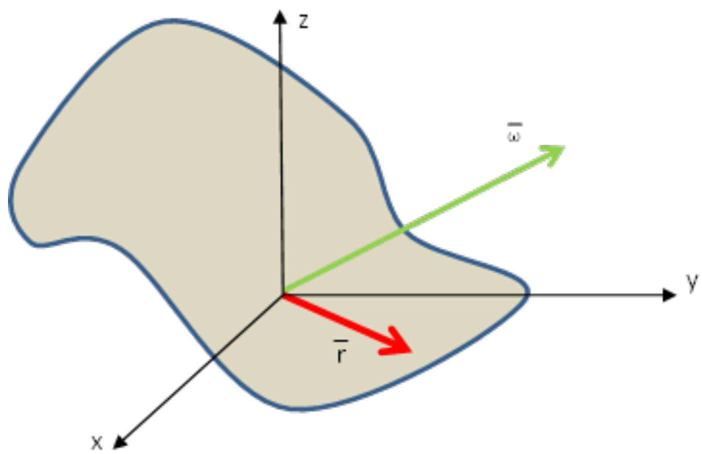



Figure 1: Using 'quiver' command

**Example: 3D Vector Function (Velocity Field)**

Consider a rigid body such as an asteroid rotating in space. We fix a coordinate system to the body as shown below




If the object is rotating about the origin of the coordinate system with angular velocity $\overline{\omega}$, we can find the velocity of any point on the object $\overline{v}$ by constructing a vector, $\overline{r}$, from the origin of the coordinate system to the point of interest and using

$$\overline{v} = \overline{\omega} \times \overline{r}$$

where $\overline{r} = (\,x \quad y \quad z\,)^T$ (position of point of interest)
$\overline{\omega} = (\,\alpha \quad \beta \quad \gamma\,)^T$ (rotational velocity vector)

In[ ]:=
```
r = {x, y, z};
ω = {α, β, γ};

v = Cross[{α, β, γ}, {x, y, z}];
v // MatrixForm
```

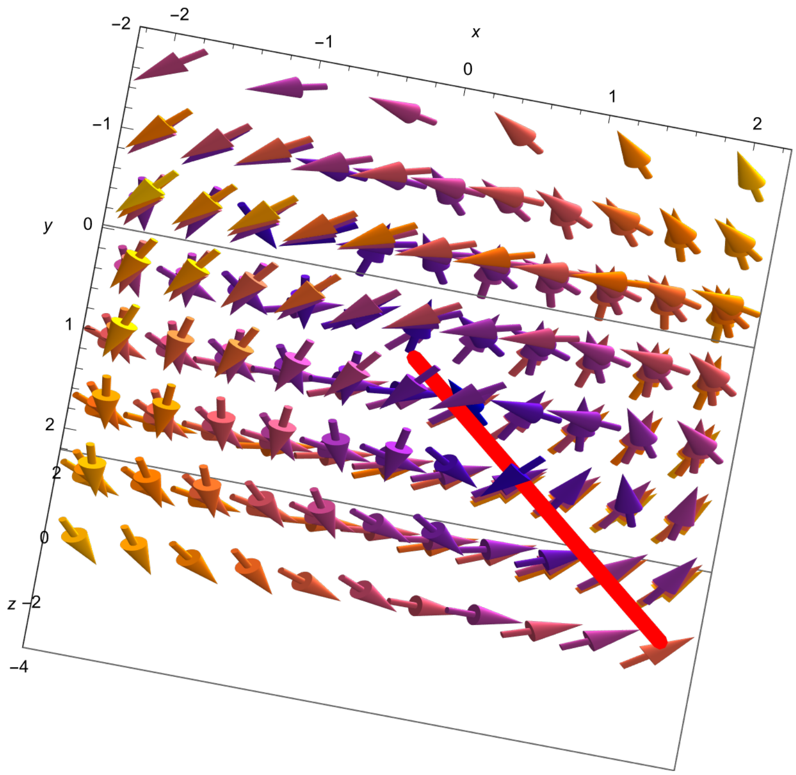
Out[ ]//MatrixForm=

$$\begin{pmatrix} z\beta - y\gamma \\ -z\alpha + x\gamma \\ y\alpha - x\beta \end{pmatrix}$$

We can plot this vector field (assuming $\alpha = 2$, $\beta = 1$, $\gamma = -4$) using the 'VectorPlot3D' function.

In[∘]:=
```
replaceString = {α → 2, β → 1, γ → -4};

(*define the vector function/field*)
vPlot = v /. replaceString;

(*Plot the vector ω*)
ωPlot = ω /. replaceString;
p1 = ListLinePlot3D[{{0, 0, 0}, ωPlot}, PlotStyle → {Red, Thickness[0.02]}];

(*Plot the vector function*)
p2 = VectorPlot3D[vPlot, {x, -2, 2}, {y, -2, 2}, {z, -2, 2}];

(*Show plots on top of each other*)
Show[p1, p2,
 AxesLabel → {x, y, z},
 PlotRange → All,
 ViewPoint → {0, -∞, 0}]
```

Out[∘]=




Matlab also provides the function' quiver3' to produce similar output. Pseudo-code and Matlab output is shown below

```
%define the points where we would
%like to draw the vectors
x = linspace(-2, 2, 5);
y = linspace(-2, 2, 7);
z = linspace(-2, 2, 9);

[X, Y, Z] = ndgrid(x,y,z);

%Define the function
alpha = 2;
beta = 1;
gamma = -4;

U = Z*beta - Y*gamma;
V = -Z*alpha + X*gamma;
W = Y*alpha - X*beta;


%Plot the vector field using 'quiver3'
figure
quiver3(X, Y, Z, U, V, W)
xlabel('x')
ylabel('y')
ylabel('z')
title('Using ''quiver3'' command')
grid on
```

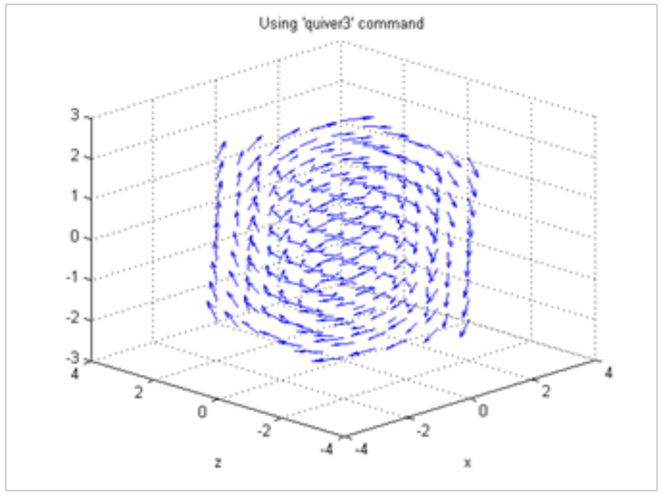


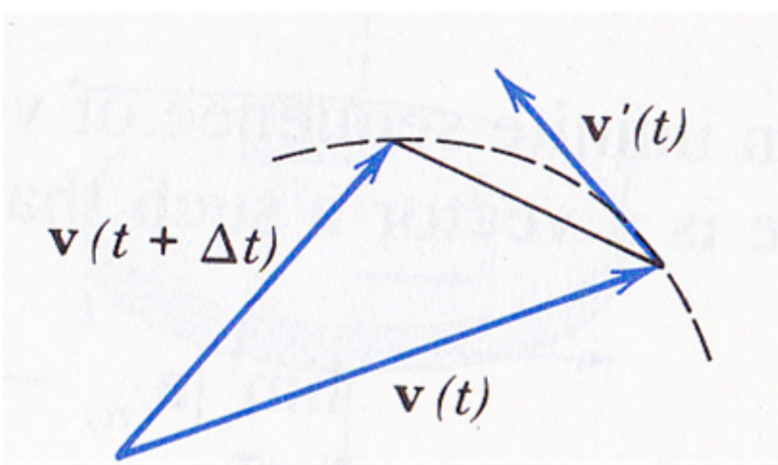

Figure 1: Using 'quiver3' command

# Vector Derivatives

The vector function defines a vector field. Similarly, a scalar function defines a scalar field. The vector and scalar functions may also depend on time or on other parameters. We can therefore define the derivative of a vector or scalar field as how this field changes as a function of the appropriate independent variable.

A vector function $\overline{v}(t)$ is said to be differentiable at a point $t$ if the following limit exists

$$\overline{v}'(t) = \lim_{\Delta t \to 0} \frac{\overline{v}(t+\Delta t) - \overline{v}(t)}{\Delta t}$$




We can differentiate each component separately to compute the derivative of the vector function

$$\bar{v}'(t) = (\, v_1'(t) \quad v_2'(t) \quad v_3'(t) \,)$$

**Differentiation Rules**

$$(\bar{u} \cdot \bar{v})' = \bar{u}' \cdot \bar{v} + \bar{u} \cdot \bar{v}'$$

$$(\bar{u} \times \bar{v})' = \bar{u}' \times \bar{v} + \bar{u} \times \bar{v}'$$

**Partial Derivatives of a Vector Function**

If the components of the vector field $\bar{v}$ are a function of multiple variables, $t_1,\ t_2,\ \ldots,\ t_n$, then the partial derivative of $\bar{v}$ w.r.t. a particular variable is defined in the normal fashion

$$\bar{v}(\bar{t}) = v_1(\bar{t})\,\hat{i} + v_2(\bar{t})\,\hat{j} + v_3(\bar{t})\,\hat{k}$$

$$\frac{\partial \bar{v}(\bar{t})}{\partial t_l} = \frac{\partial v_1(\bar{t})}{\partial t_l}\,\hat{i} + \frac{\partial v_2(\bar{t})}{\partial t_l}\,\hat{j} + \frac{\partial v_3(\bar{t})}{\partial t_l}\,\hat{k}$$

where $\quad \bar{t} = (\, t_1 \quad t_2 \quad \ldots \quad t_n \,)$

**Example**

Consider a vector function $\bar{r} : \mathbb{R}^2 \to \mathbb{R}^3$.

$$\bar{r}(\bar{t}) = \begin{pmatrix} a\cos(t_1) \\ a\sin(t_1) \\ t_2 \end{pmatrix}$$

```
rx[t1_, t2_] = a Cos[t1];
ry[t1_, t2_] = b Sin[t1];
rz[t1_, t2_] = t2;
```

We can then easily calculate the partial derivative of this vector function with respect to both $t_1$ and $t_2$

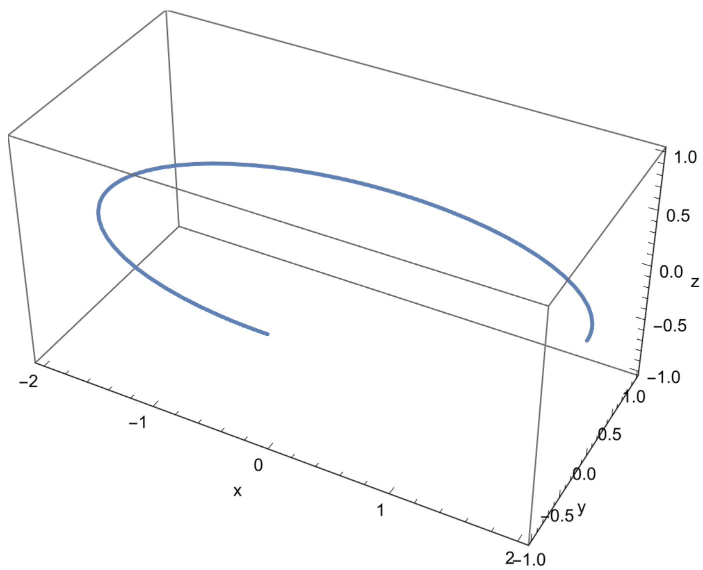
$$\frac{\partial \bar{r}(\bar{t})}{\partial t_1} = \begin{pmatrix} -a\sin(t_1) \\ a\cos(t_1) \\ 0 \end{pmatrix}$$

$$\frac{\partial \bar{r}(\bar{t})}{\partial t_2} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

If we consider $\bar{r}(t_1,\ t_2)$ to be the position vector at a given $t_1$ and $t_w$, then we can interpret $\frac{\partial \bar{r}(\bar{t})}{\partial t_1}$ as how this position vector changes as $t_1$ changes and $\frac{\partial \bar{r}(\bar{t})}{\partial t_2}$ describes how the position changes with $t_2$.

For example holding $t_2$ = constant = 4 and varying $t_1 \in [0,\ 3/2\,\pi]$ yields (with $a = 2,\ b = 1$)

```
t2constant = 4;
ParametricPlot3D[
 {rx[t1, t2constant], ry[t1, t2constant], rz[t1, t2constant]} /. {a → 2, b → 1},
 {t1, 0, 3 / 2 π},
 AxesLabel → {"x", "y", "z"}]
```




For example holding $t_1$ = constant = $\pi/2$ and varying $t_2 \in [-3, -1]$ yields

```
t1constant = π / 2;
ParametricPlot3D[
 {rx[t1constant, t2], ry[t1constant, t2], rz[t1constant, t2]} /. {a → 2, b → 1},
 {t2, -3, -1},
 AxesLabel → {"x", "y", "z"}]
```