Derivadas e Integrais

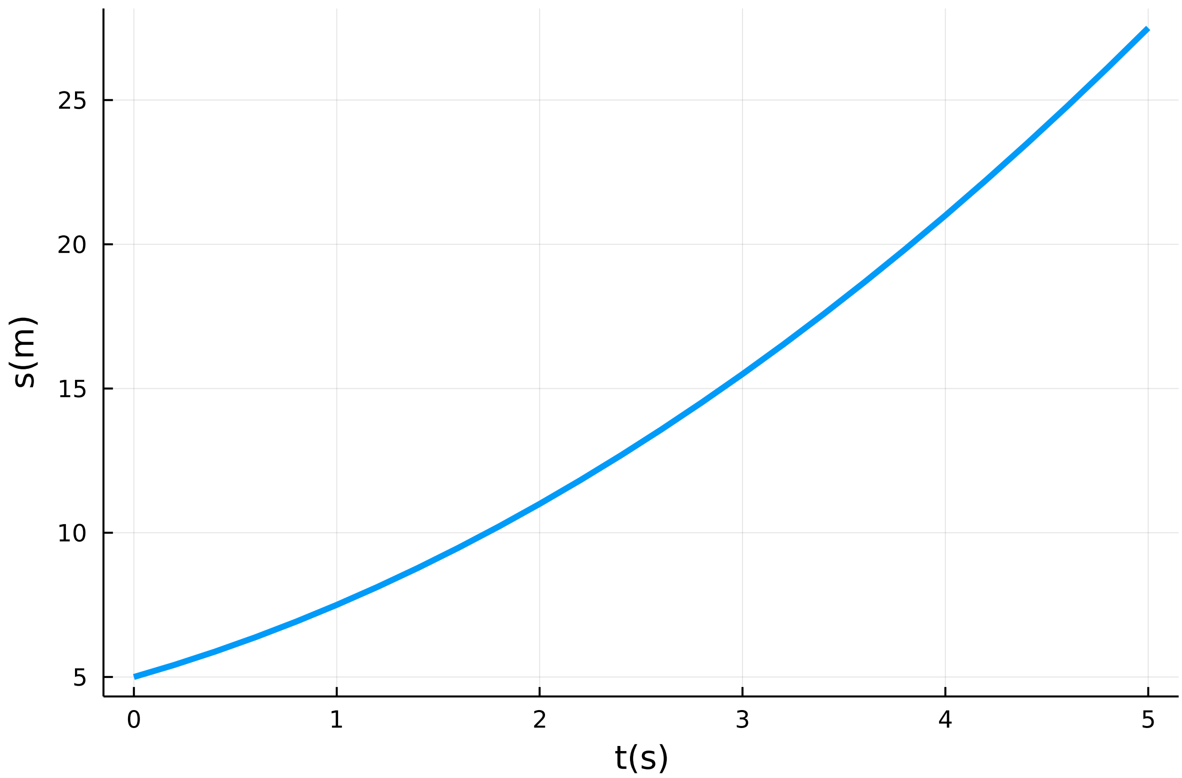
Vamos considerar o caso unidimensional em que a posiçao de uma partícula em no tempo é definida pela função:

$$s(t) = \frac{t^2}{2} + 2t + 5$$

```julia
using Plots
```

```
s (generic function with 1 method)
```

```julia
s(t) = 0.5*t^2 + 2*t +5
```



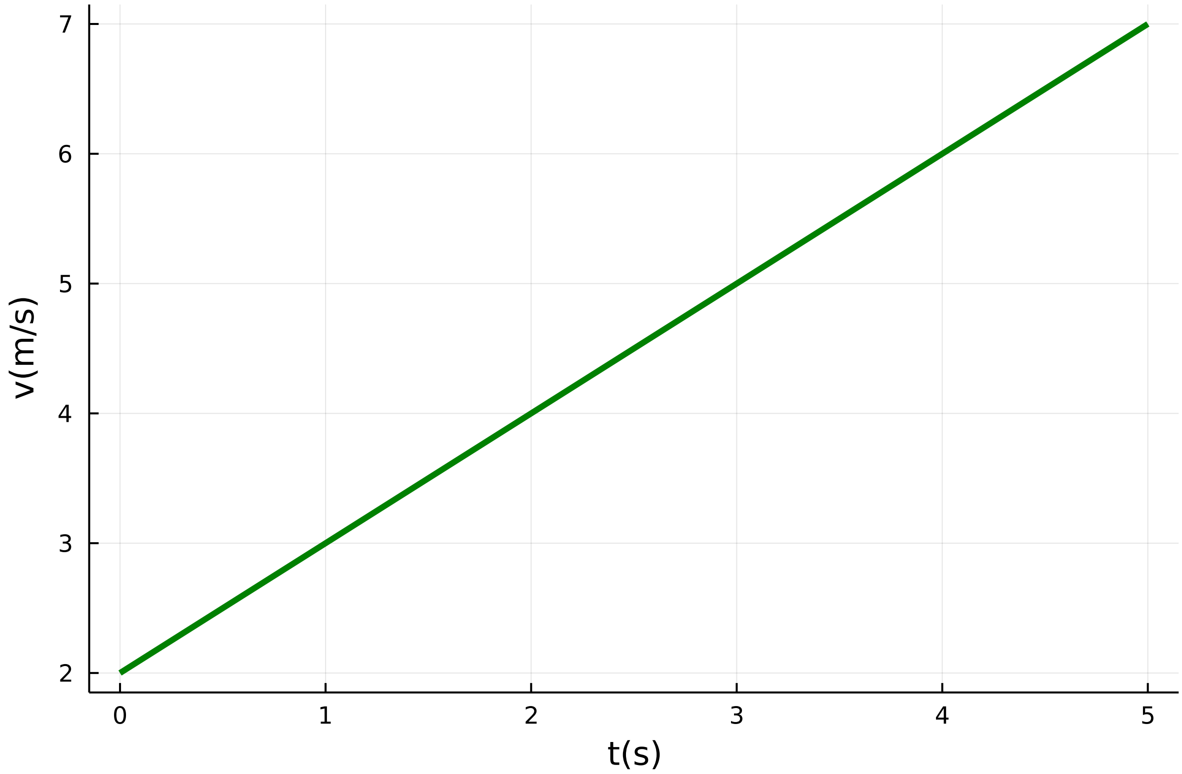

```julia
plot(0:0.2:5, s.(0:0.2:5), label=false, xlabel="t(s)", ylabel="s(m)", linewidth=3)
```

Derivando, encontramos a expressão para a velocidade da partícula.

$$\frac{ds}{dt} = v(t) = t + 2$$

```
v (generic function with 1 method)
```

```julia
v(t) = t + 2
```

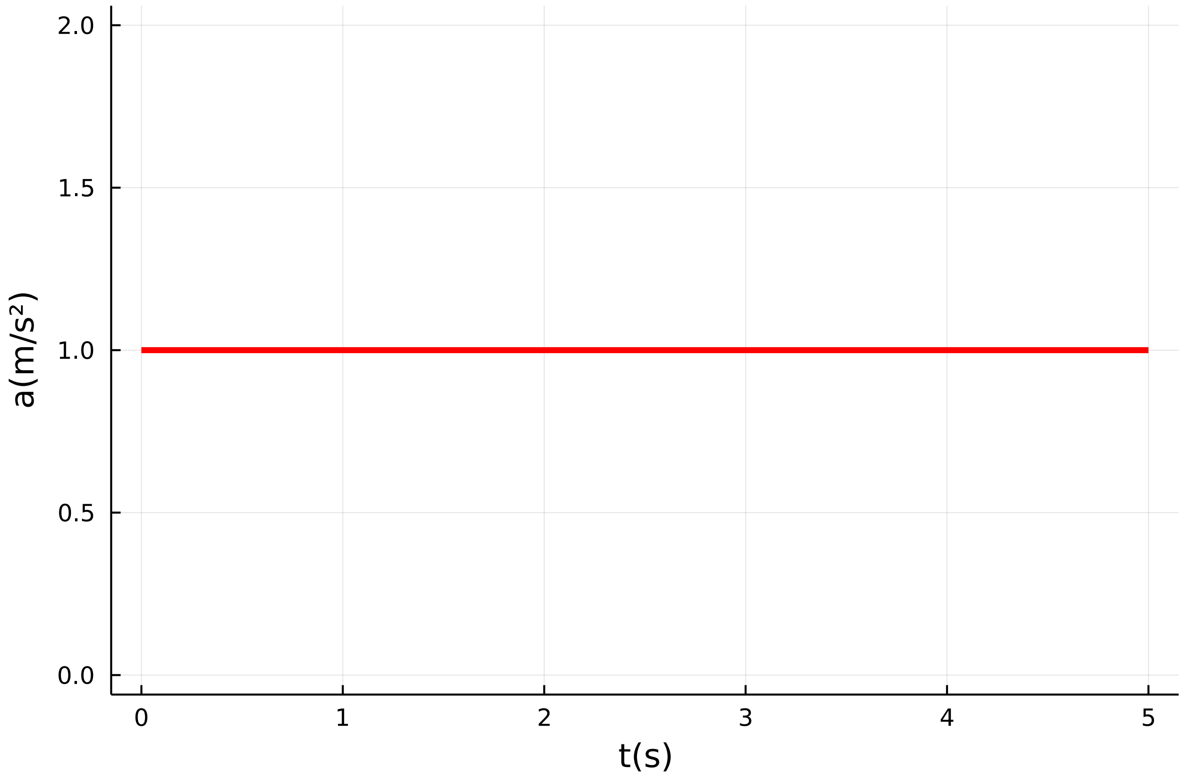



```
plot(0:0.2:5, v.(0:0.2:5), label=false, xlabel="t(s)", ylabel="v(m/s)",
linecolor="green", linewidth=3)
```

Derivando novamente, descobrimos que a aceleração é constante e igual a 1 m/s²

```
a (generic function with 1 method)
```

```
a(t) = 1
```




```
plot(0:0.2:5, a.(0:0.2:5), label=false, xlabel="t(s)", ylabel="a(m/s²)",
linecolor="red", linewidth=3, ylim=[0,2])
```

Podemos percorrer o caminho inverso e integrar a aceleração para recuperar a velocidade, em seguida integrar a velocidade para encontrar a posição (teorema fundamental do cálculo)

Vamos tentar integrar a aceleração para ver o que acontece.

$$v_2(t) = \int a(t)dt = \int dt = t$$

Vemos que a expressão final obtida, $v_2(t) = t$, é diferente da que começamos, $v(t) = t + 2$

Qual a explicação para isso?

Ao integrarmos não conseguimos recuperar algumas informações (informaçõa contida no espaço nulo, se traçarmos um paralelo com álgebra linear). Na realidade, o que encontramos é uma família de funções. A função específica apenas será determinada a partir das condições de contorno.

Modelos matemáticos

Como exemplo, vamos interpretar o modelo epidemiológico SIR.

$$\begin{cases} \frac{dS}{dt} = -rSI \\ \frac{dI}{dt} = rSI - aI \\ \frac{dR}{dt} = aI \end{cases}$$

```
using DifferentialEquations
```

```
sir! (generic function with 1 method)
```

```
function sir!(du, u, p, t)
    du[1] = -p[1]*u[1]*u[2]
    du[2] = p[1]*u[1]*u[2] - p[2]*u[2]
    du[3] = p[2]*u[2]
end
```

```
ODEProblem with uType Vector{Float64} and tType Float64. In-place: true
timespan: (0.0, 360.0)
u0: 3-element Vector{Float64}:
 499.0
   1.0
   0.0
```

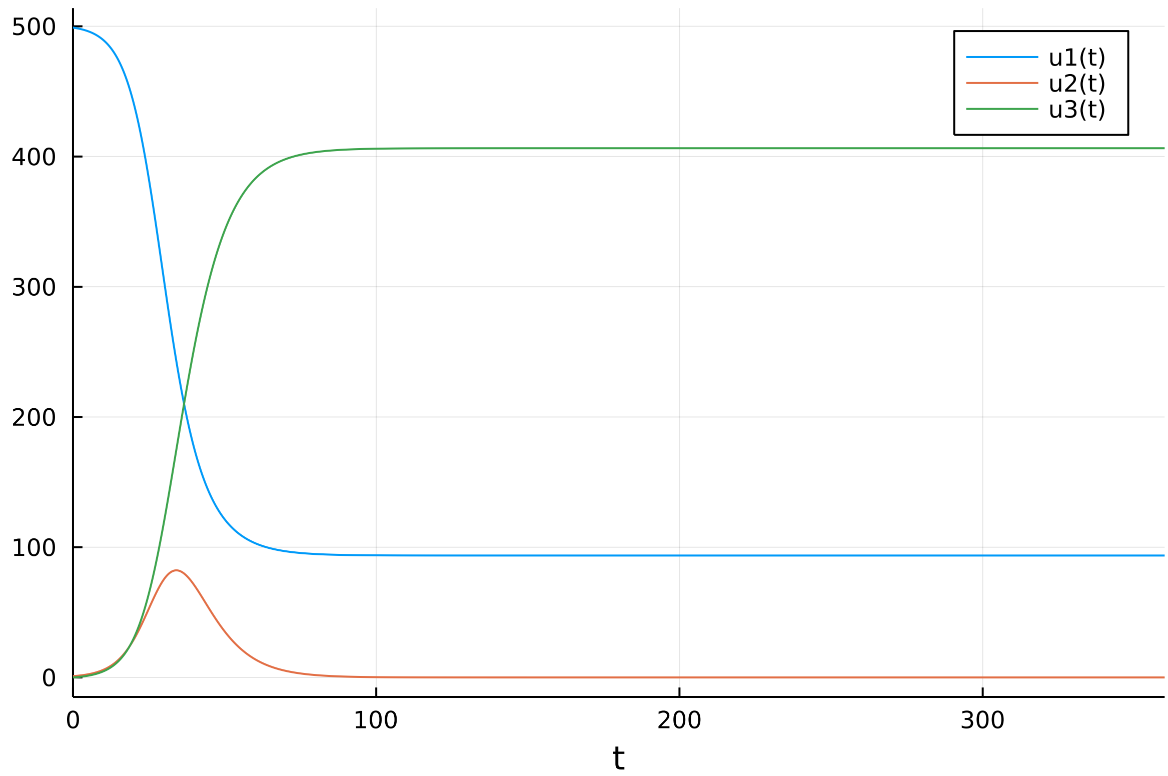
```
begin
    # Condições iniciais:
    u₀ = [499.0, 1.0, 0.0]
    tspan = [0.0,360.0]
    p = [0.35/500, 0.17]

    # Definindo o problema
    prob = ODEProblem(sir!, u₀, tspan, p)
end
```

sol =

	timestamp	value1	value2	value3
1	0.0	499.0	1.0	0.0
2	0.00831474	498.997	1.00149	0.00141456
3	0.0914621	498.968	1.01653	0.0156767
4	0.564398	498.793	1.10645	0.100969
5	1.5923	498.357	1.32997	0.313251
6	3.09756	497.556	1.74015	0.703743
7	5.00856	496.178	2.44443	1.3771
8	7.41229	493.642	3.73619	2.62174
9	10.2847	488.851	6.15888	4.99051
10	13.6583	479.34	10.8982	9.76195
	more			

```
# Resolvendo problema
sol = solve(prob)
```




```
plot(sol)
```