معادلات دیفرانسیل

DSolve[equation, y[x], x] : معادلهي **equation** را حل و **y[x]** را میدهد . متغیر **x** فرض میشود

DSolve[y'[x] == y[x], y[x], x]

$\{\{y[x] \to e^{x} C[1]\}\}$

میتوان شرایط اولیه داد

DSolve[{y'[x] == y[x], y[0] == 2}, y[x], x]

$\{\{y[x] \to 2 e^{x}\}\}$

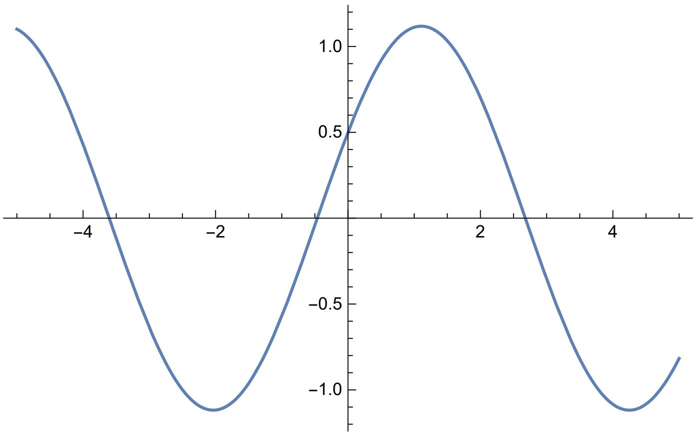
DSolve[y''[x] + y[x] == 0, y[x], x]

$\{\{y[x] \to C[1] \mathrm{Cos}[x] + C[2] \mathrm{Sin}[x]\}\}$

sol = DSolve[{y''[x] + y[x] == 0, y[0] == 0.5, y'[0] == 1}, y[x], x]

$\{\{y[x] \to 0.5 \mathrm{Cos}[x] + 1. \mathrm{Sin}[x]\}\}$

Plot[y[x] /. sol, {x, -5, 5}]



DSolve[x^2 y''[x] + x y'[x] + (x^2 - 4) y[x] == 0, y[x], x]

{{y[x] → BesselJ[2, x] C[1] + BesselY[2, x] C[2]}}

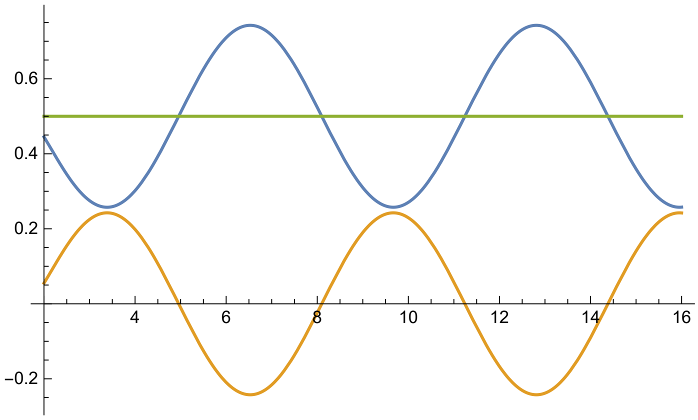
دستگاه معادلات

sol1 = DSolve[{y'[x] - 4 z[x] == Cos[x], y[x] + z[x] == 1/2, y[Pi/2] == 1/2}, {y[x], z[x]}, x]

$\{\{y[x] \to \frac{1}{34} e^{-4x} \left(-2 e^{2\pi} + 17 e^{4x} + 8 e^{4x} \mathrm{Cos}[x] + 2 e^{4x} \mathrm{Sin}[x]\right),$

$z[x] \to -\frac{1}{17} e^{-4x} \left(-e^{2\pi} + 4 e^{4x} \mathrm{Cos}[x] + e^{4x} \mathrm{Sin}[x]\right)\}\}$

```
Plot[{y[x] /. sol1, z[x] /. sol1, (y[x] + z[x]) /. sol1}, {x, 2, 16}]
```



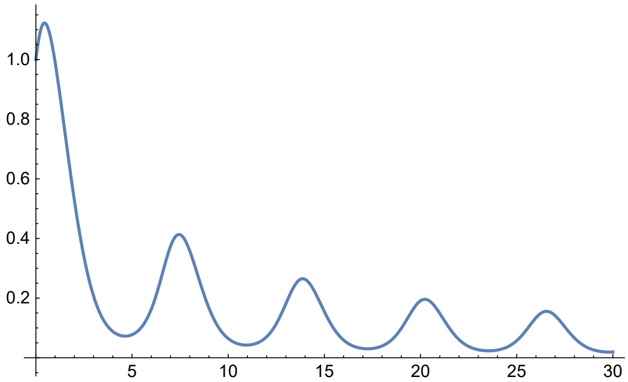
حل عددي معادلات ديفرانسيل

`NDSolve[eqn, y, {x, xmin, xmax}]` : حل eqn را در فاصله‌ي داده‌شده به طور عددي ميسازد

```
s = NDSolve[{y'[x] == y[x] Cos[x + y[x]], y[0] == 1}, y, {x, 0, 30}]
```

{{y → InterpolatingFunction[Domain: {{0., 30.}} Output: scalar]}}

```
Plot[y[x] /. s, {x, 0, 30}, PlotRange → All]
```



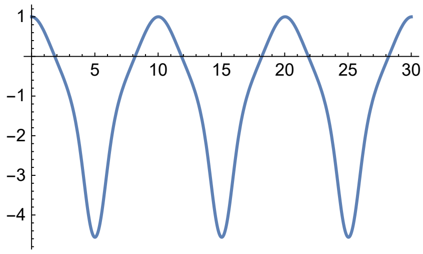

```
y[13] /. s[[1]]
```

0.181682

```
s = NDSolve[{y''[x] + Sin[y[x]] y[x] == 0, y[0] == 1, y'[0] == 0}, y, {x, 0, 30}]
```

{{y → InterpolatingFunction[Domain: {{0., 30.}} Output: scalar]}}

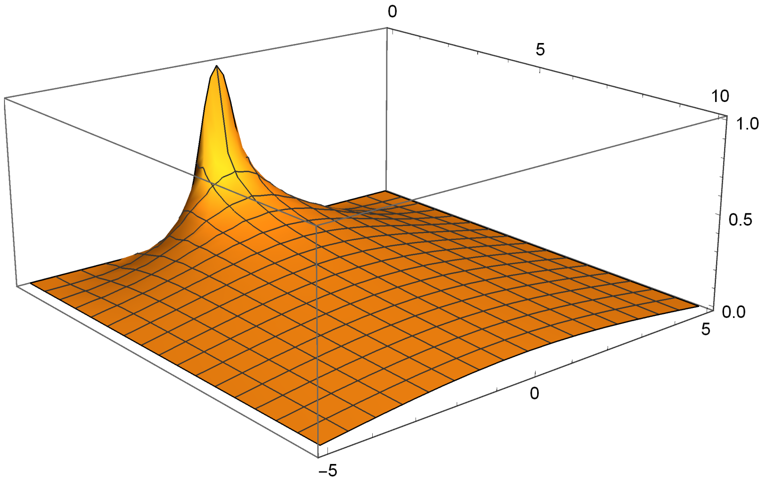
```
Plot[y[x] /. s, {x, 0, 30}, PlotRange → All]
```



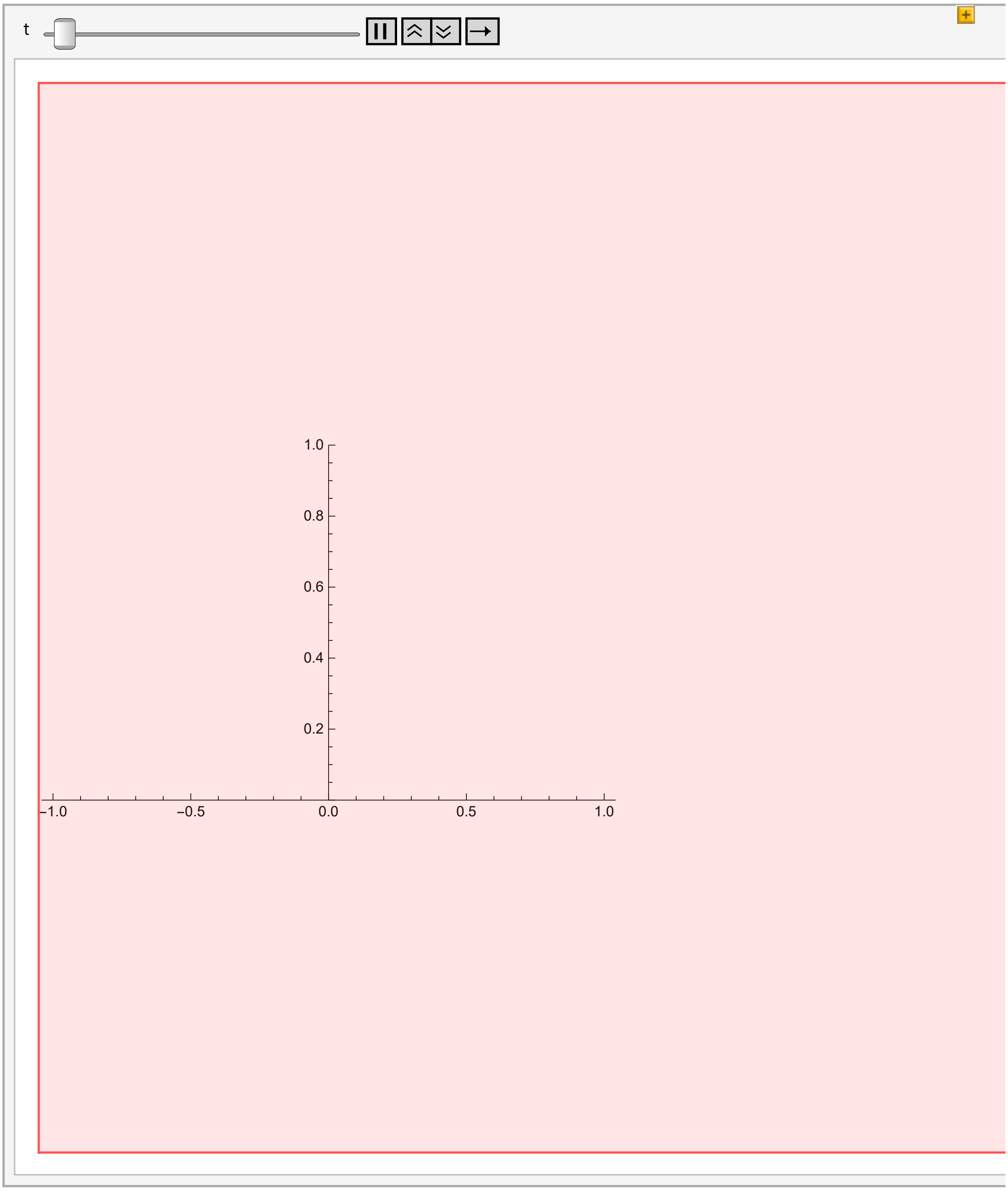
```
diff = NDSolve[{D[u[t, x], t] == D[u[t, x], x, x],
   u[0, x] == Exp[-2 x^2], u[t, -5] == 0, u[t, 5] == 0}, u, {t, 0, 10}, {x, -5, 5}]
```

```
{{u → InterpolatingFunction[ Domain: {{0., 10.}, {-5., 5.}}
                             Output: scalar ]}}
```

```
Plot3D[u[t, x] /. diff, {t, 0, 10}, {x, -5, 5}, PlotRange → All]
```



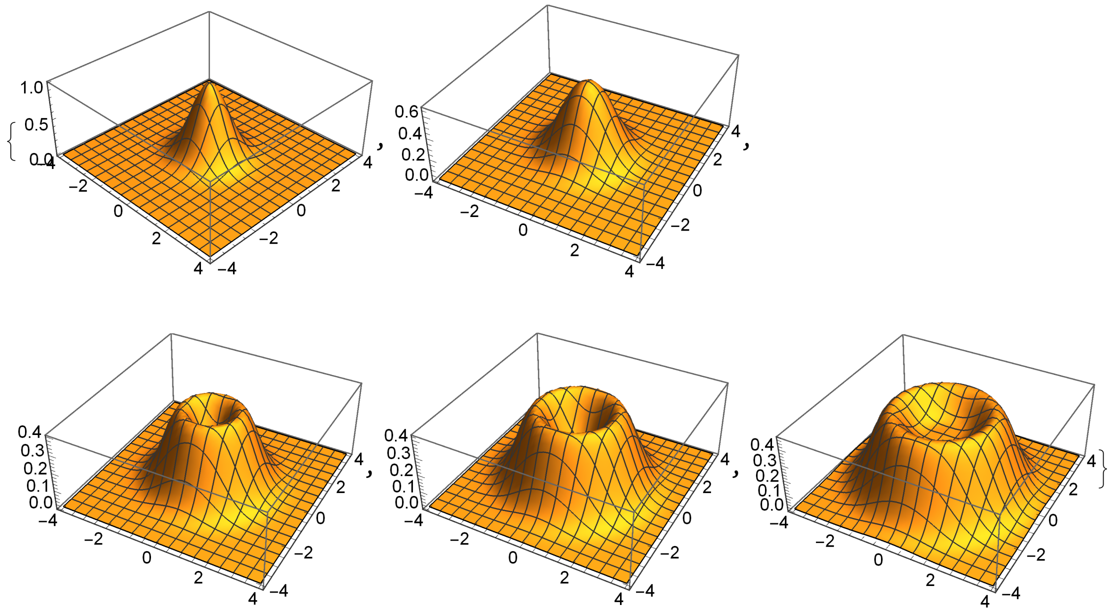
```
L = 5;
Animate[
 Plot[u[t, x] /. diff, {x, -L, L}, PlotRange → {0, 1}, PlotPoints → 25], {t, 0, 5, .02}]
```



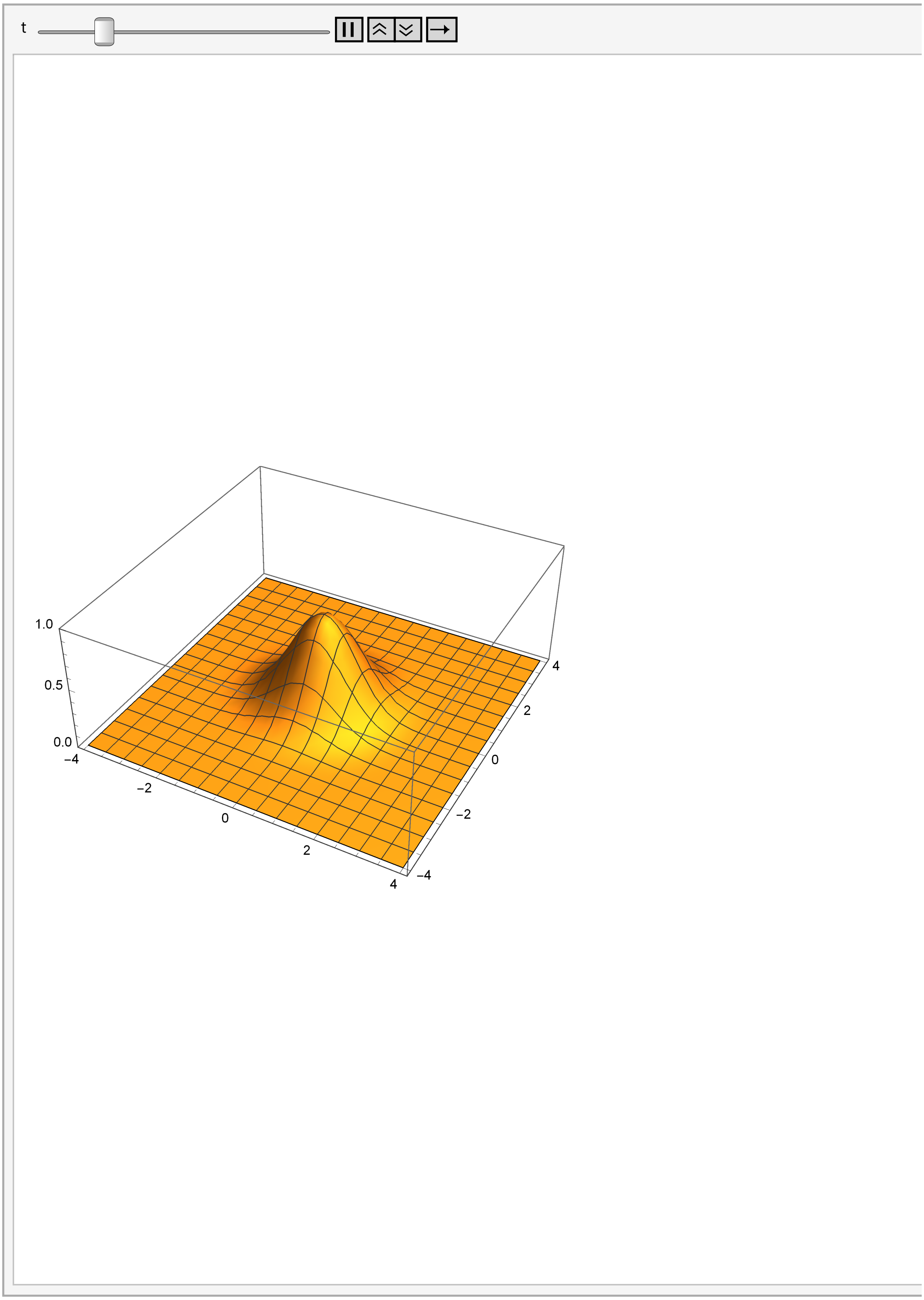
In[1]:=
```
L = 4;
sol = NDSolve[{D[u[t, x, y], t, t] == D[u[t, x, y], x, x] + D[u[t, x, y], y, y] ,
   u[t, -L, y] == u[t, L, y], u[t, x, -L] == u[t, x, L], u[0, x, y] == Exp[-(x^2 + y^2)],
   Derivative[1, 0, 0][u][0, x, y] == 0}, u, {t, 0, L/2}, {x, -L, L}, {y, -L, L}]
```

Out[2]=
```
{{u → InterpolatingFunction[ Domain: {{0., 2.}, {-4., 4.}, {-4., 4.}}
                             Output: scalar ]}}
```

```
Table[Plot3D[u[t, x, y] /. sol, {x, -L, L},
  {y, -L, L}, PlotPoints → 50, PlotRange → All], {t, 0, 2, .5}]
```



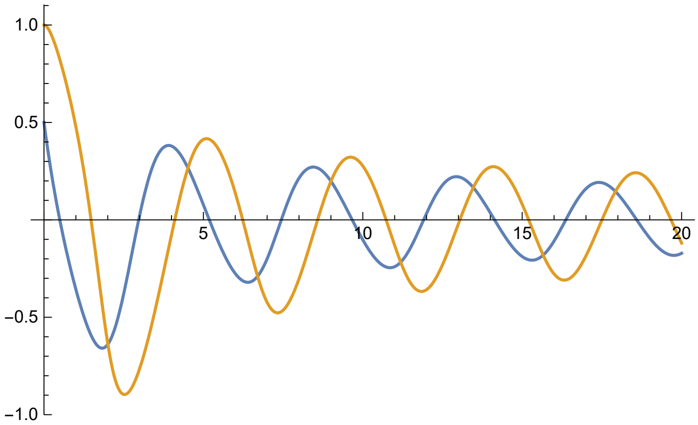
```
Animate[Plot3D[u[t, x, y] /. sol, {x, -L, L},
  {y, -L, L}, PlotRange → {0, 1}, PlotPoints → 25], {t, 0, 2, .02}]
```



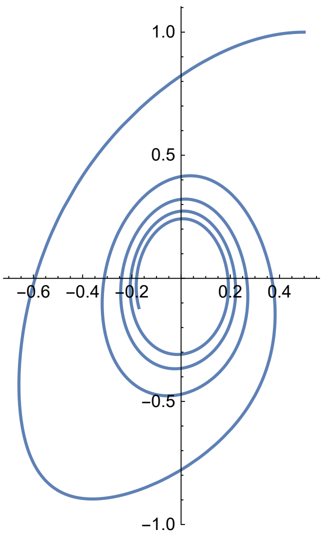
```
s = NDSolve[
  {x'[t] == -y[t] - x[t]^2, y'[t] == 2 x[t] - y[t]^3, x[0] == .5, y[0] == 1}, {x, y}, {t, 0, 20}]
```

```
{{x → InterpolatingFunction[ Domain: {{0., 20.}} Output: scalar ],
  y → InterpolatingFunction[ Domain: {{0., 20.}} Output: scalar ]}}
```

```
Plot[{x[t] /. s, y[t] /. s}, {t, 0, 20}]
```




```
ParametricPlot[{x[t], y[t]} /. s, {t, 0, 20}]
```



```
m = 2.3; g = 9.8; z0 = 0; vz0 = 15.7;
s = NDSolve[{m z''[t] == -m g, z[0] == z0, z'[0] == vz0}, z[t], {t, 0, 4}]
```

```
{{z[t] → InterpolatingFunction[ Domain: {{0., 4.}}
                                Output: scalar ][t]}}
```

```
Plot[z[t] /. s, {t, 0, 4}, AxesLabel → {t, z}]
```