

**\* results**

**\***

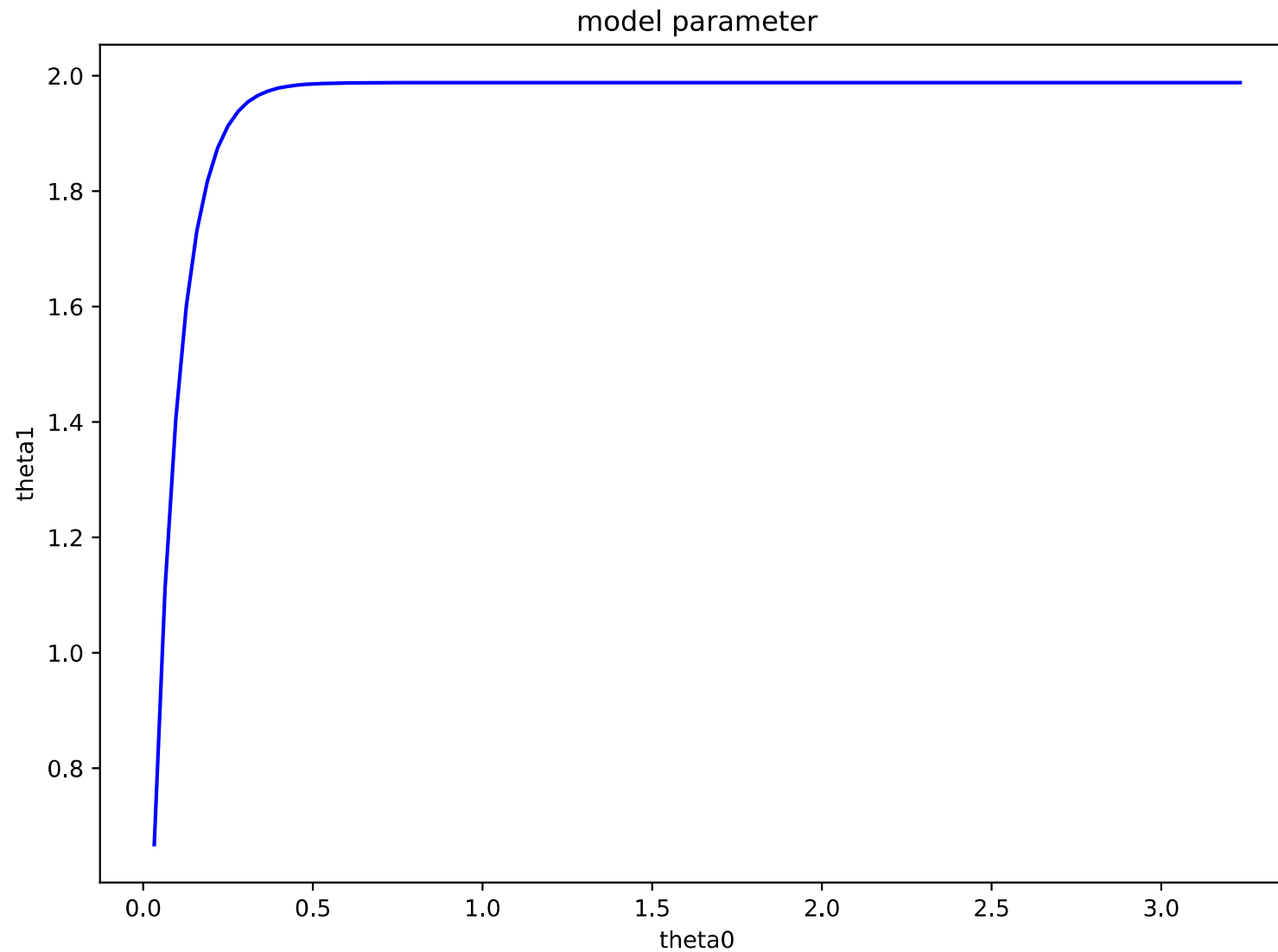
**# 01. plot the input data in blue point and the regression result in red curve**

```
In [ ]: plot_data_regression(x, y, f)
```



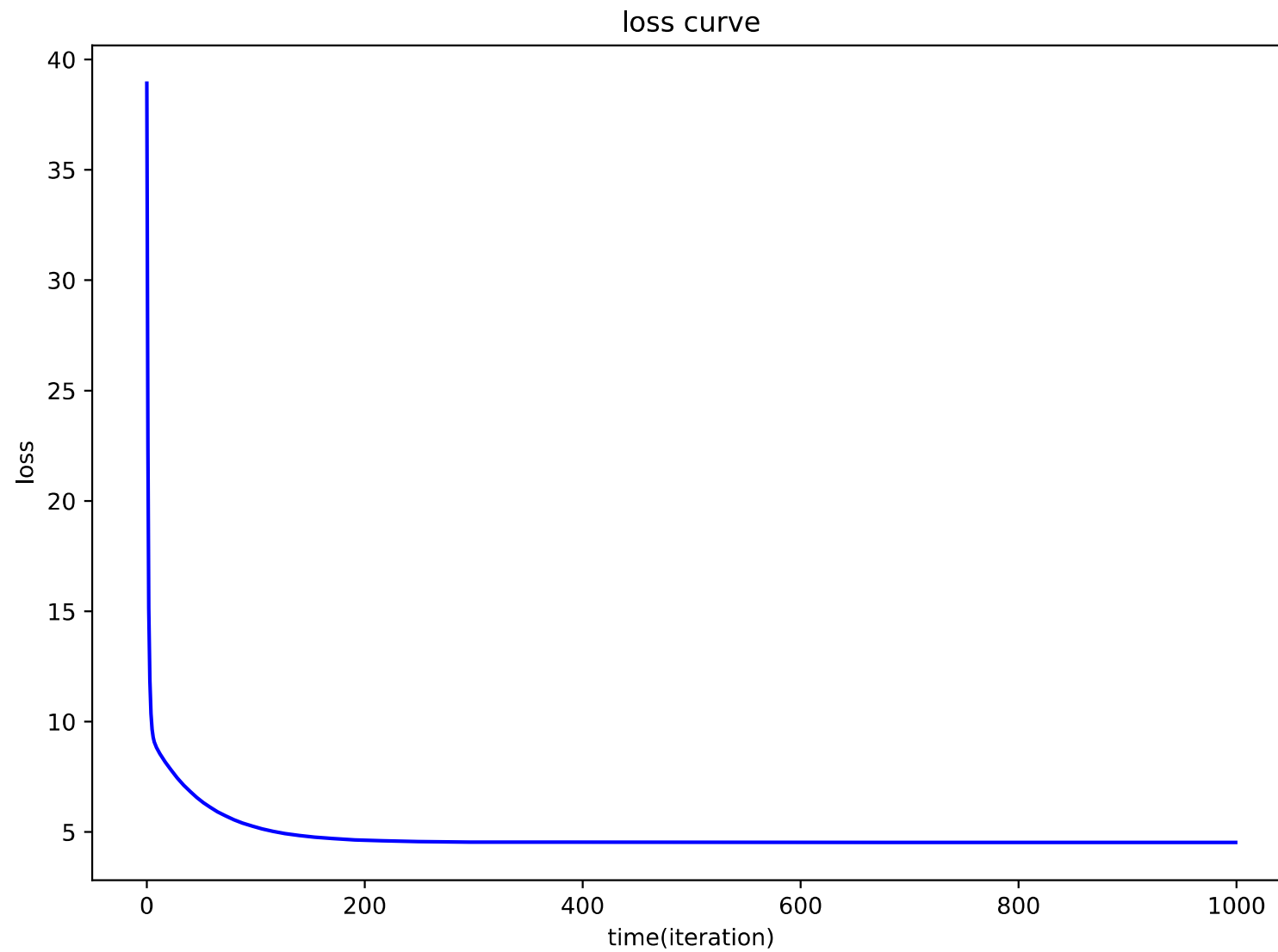
**# 02. plot the values of the model parameters  $\theta_0$  in blue curve and  $\theta_1$  in green curve over the gradient descent iterations**

```
In [ ]: plot_model_parameter(theta0_iteration, theta1_iteration)
```



**# 03. plot the loss values  $\mathcal{L}(\theta)$  in red curve over the gradient descent iterations**

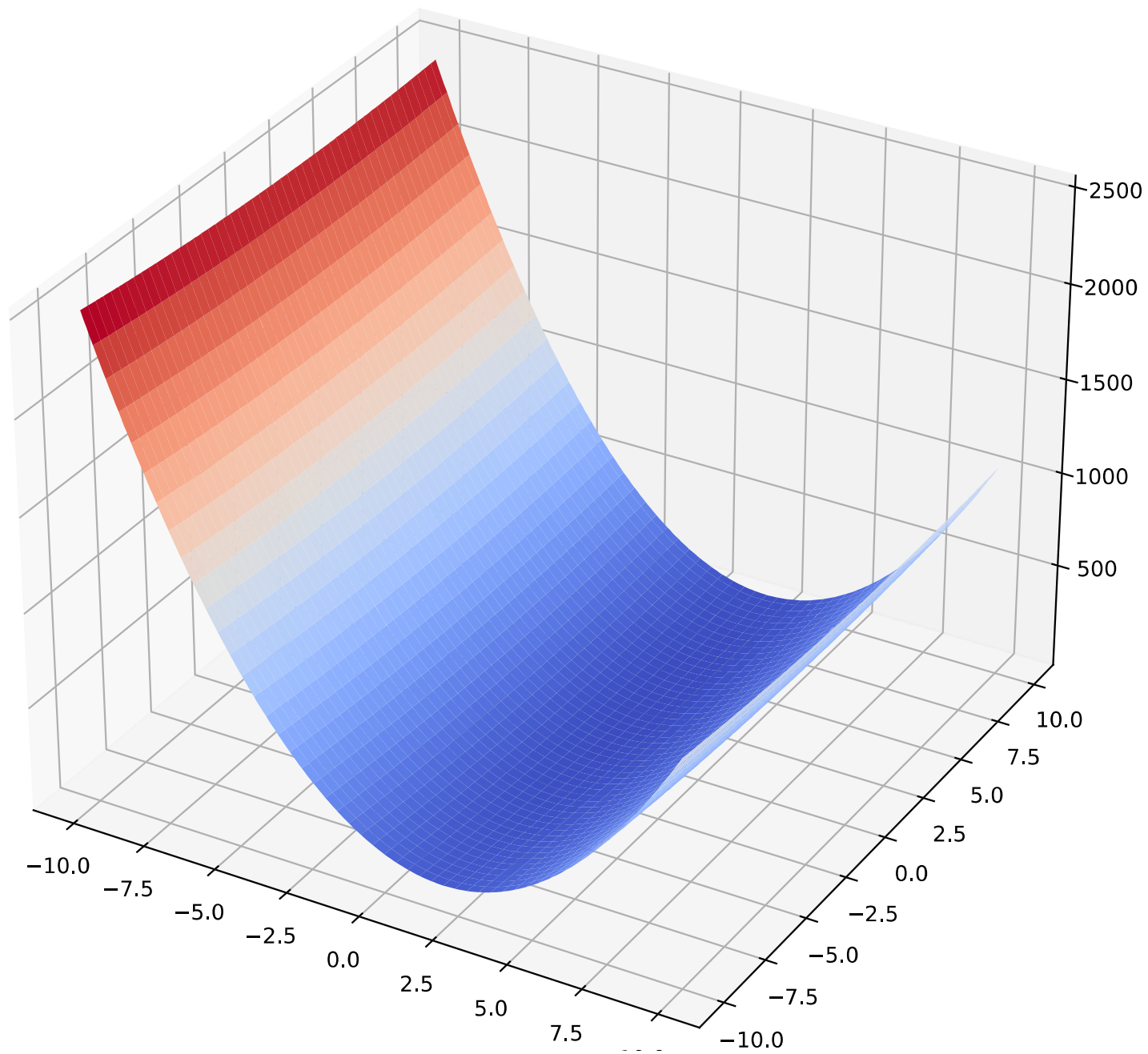
```
In [ ]: plot_loss_curve(loss_iteration)
```



**# 04.** plot the loss surface in 3-dimension surface where  $x$ -axis represents  $\theta_0$ ,  $y$ -axis represents  $\theta_1$  and  $z$ -axis represents  $\mathcal{L}$

```
In [ ]: plot_loss_surface(grid_theta0, grid_theta1, grid_loss)
```

```
<ipython-input-12-4681cda39795>:18: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by passing a unique label to each axes instance.  
... pls = plt.axes(projection='3d')
```



In [ ]: