

Primera Tarea

Ejercicio 1

Crear una variable llamada 'historial_J' que almacene en cada iteración del algoritmo Gradient Descent el valor de la Función Costo.

Solución

```
27 - sum_0 = 0;
28 - sum_1 = 0;
29 - sum_j = 0;
30
31 - for i = 1:m
32 -     h_theta = theta_0 + theta_1 * x(i);
33 -     sum_0 = sum_0 + ( h_theta - y(i) );
34 -     sum_1 = sum_1 + ( ( h_theta - y(i) ) * x(i) );
35 -     sum_j = sum_j + (h_theta - y(i))^2;
36 - end
37
38 - historial_J(iter) =(1/(2*m))*sum_j;
```

Ejercicio 2

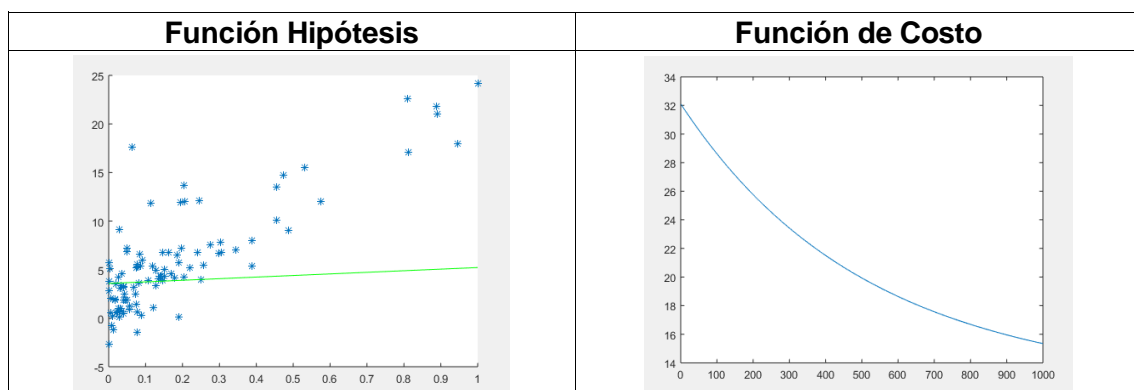
Para valores de $\alpha = 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 1.3, 2, 3$. Hacer un gráfico (iter, J) donde se plotee el valor de la Función Costo (J) en la i-ésima iteración (Iter). El máximo número de iteraciones será de 1000.

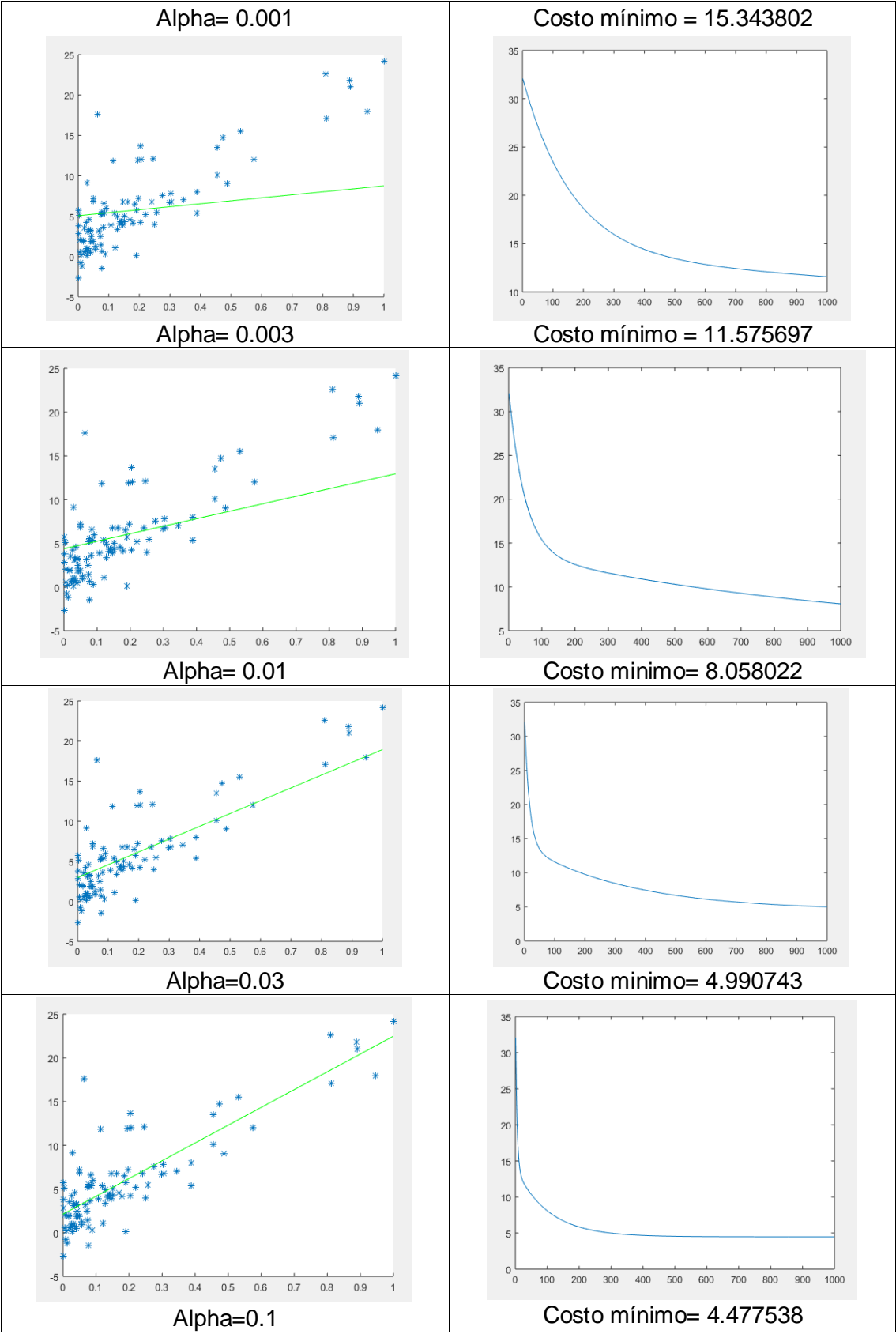
Pregunta: Cuál valor de alpha cree usted que consigue converger más rápido?

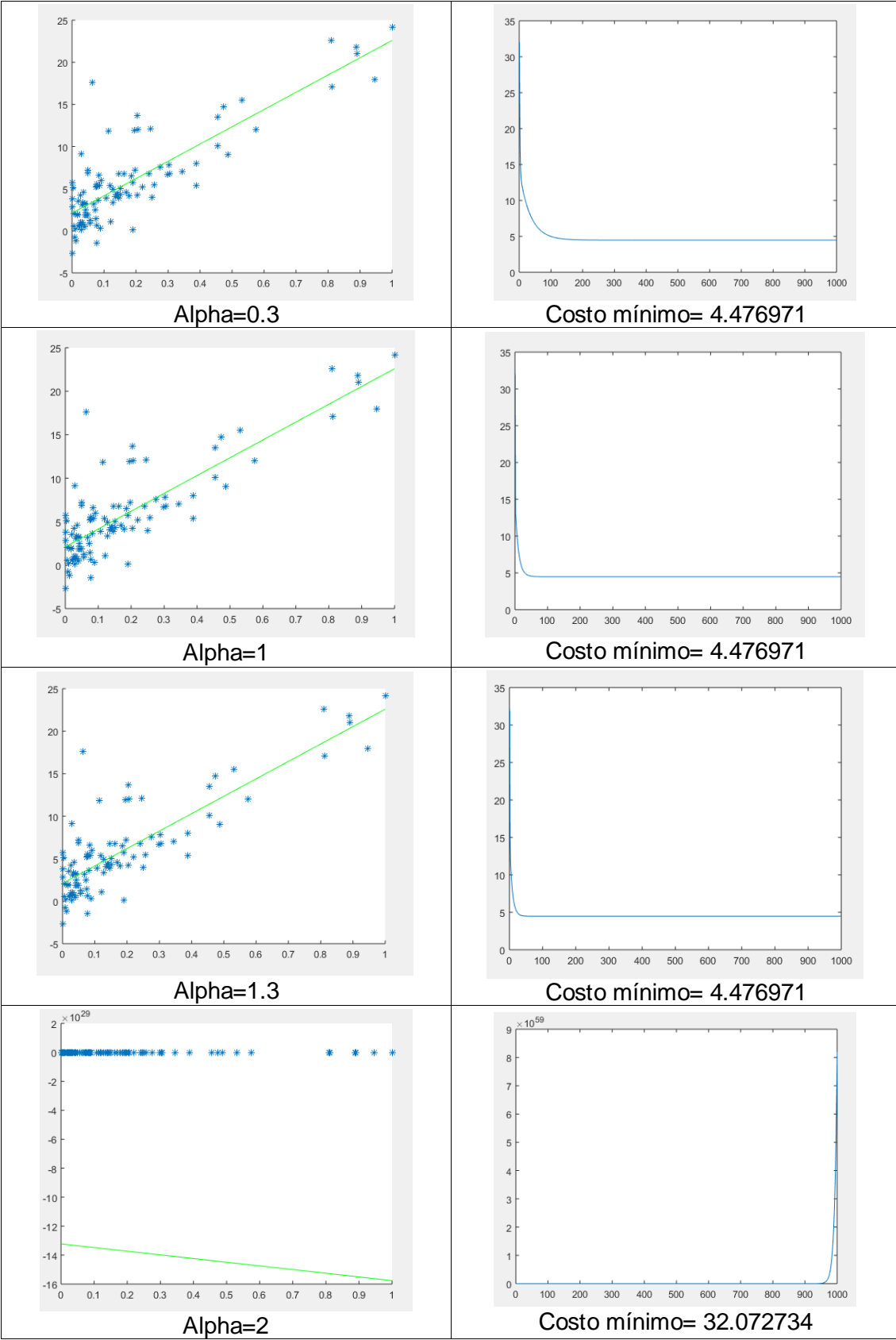
Solución

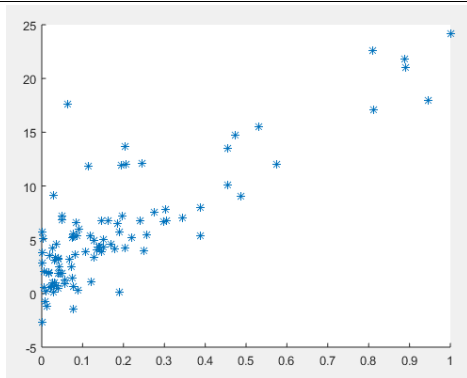
$\alpha = 0.3, 1, 1.3$

Sustentación:

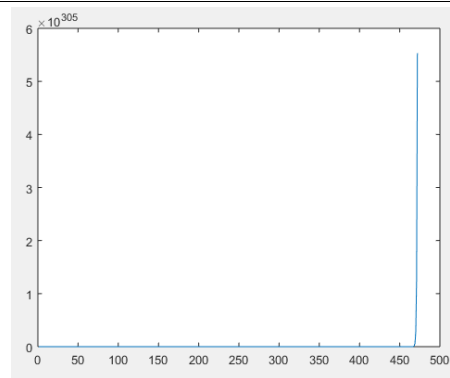








Alpha=3



Costo mínimo= 32.072734