laSalle
Ramon Llull University

Optimization of functions of several variables with Matlab

Statistics and Mathematical Analysis

Name: Eugènia Pacheco Ferrando

Name: Laia Casas Irure

Name: Valèria Ezquerra Rodriguez

laSalle
Ramon Llull University

Question 1 *(3p)*

Explain what criterion you apply to decide that the gradient descent algorithm has converged.

We apply two criterias at the same time. Whenever one is reached, it exits the program. The first one defines a maximum number of iterations and the second one determines when the step size related to the learning rate (alpha) approaches zero. Since the tolerance is also near zero, we break the for loop every time that the real tolerance is less than the defined one.

Question 2 *(2p)*

Complete the following table, indicating the coordinates of the real absolute minimum of each function.

Function	***x_real***	***y_real***
Circular Paraboloide	0	0
Elliptic Paraboloide	-20	-30
Inverted Gaussian	1	1
Arbitrary function	41	-39

Briefly explain how you find the position of the real absolute minimum of the function.

*We define the real minimum with the matlab function min twice. Once for each column of z, a matrix of 201*201, and another to find the final minimum of this previous collection. Finally, we need its coordinates, so we use matlab find functon which basically goes through every value of matrix z and check if it matches with the one stored (M). If it matches, it saves the row as well as the column, our real_x and real_y.*

laSalle
Ramon Llull University

Question 3 *(5p)*

For each of the four functions to optimize, complete the following table:

Function	x_initial	y_initial	alfa	x_final	y_final	Number of iterations	|minim_real-minim_final|
Circular Paraboloid	5	5	0.1	0.4	0.4	10	0
	5	5	0.01	0.48	0.48	109	0
Elliptic Paraboloid	1	100	0.1	-19.64	-29.6	22	0
	100	1	0.1	-19.52	-29.6	31	0
Inverted Gaussian	50	50	0.1	E	E	E	E
	50	50	10^4	1.4681	1.4681	87	0
Arbitrary function	150	150	10^4	40.69	-39.42	24	0
	50	190	10^4	41.43	-38.71	17	0

Briefly justify the differences you observe in the results obtained for each function, explaining how accurate the algorithm is in finding the absolute minimum:

Circular paraboloid

Since x_initial and y_initial have the same values, the difference is in the alpha (learning rate). The number of iterations increases since the alpha has decreased with respect to 0.1. Taking into account the coordinates of the real minimum, we conclude that it's 100% accurate.

Elliptic paraboloid

We observe that the more x_initial we have, the more iterations it took to converge to the minimum. Similarly to circular paraboloid, we can say that it's very accurate.

Inverted Gaussian

The inverted gaussian depends on the alpha value. If it's decimal, it gives us "Error in Gradient_Descendent x_final = positions(1, iterator);" , so we assume that the algorithm moves so slow that the iterator exceeds array out bounds. However, if alpha is big enough (10^4 for example), the accuracy is 100% again.

Arbitrary function
The number of iterations increases having both x_initial and y_initial the same values, and same alpha. The difference between those coordinates in the first case is smaller (1’27) than the one in the second case (2’72).