# 1.1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a=0.5 | a=1 | a=2 | a=5 |
| .001 | >1000 | >1000 | >1000 | 990 |
| .01 | 760 | 414 | 223 | 97 |
| .03 | 252 | 137 | 73 | 31 |
| .1 | 65 | 40 | 21 | 8 |
| .3 | 24 | 12 | 5 | 8 |
| 1 | 6 | 1 | Div | Div |
| 3 | 6 | div | Div | div |
| Fastest | 2 | 1 | 0.5 | 0.25 |
| Divergence  threshold | 4 | 2 | 1 | 0.5 |

# 1.2)

Como se pode observar, quanto mais pequeno fôr o, mais interações o algoritmo fará, sendo que é preciso ter cuidado para não ter um pequeno demais. Por outro lado, um demasiado grande leva a que o algoritmo não converja. Além disso é importante referir, que o deve ser de acordo com o parâmetro a.

Tal como foi visto na aula de problemas, o valor de que converge mais rapidamente é ,pois:

E o valor que faz o algoritmo oscilar (estar no divergence threshold) será pois:

# 1.3)

Tal como foi visto na aula de problemas, o valor de que converge mais rapidamente é ,pois:

[TIRAR FOTO DE FORMULAS E POR AQUI OU ESCREVER E TIRAR FOTO]

Não. Basta o corresponder a um ponto estácionário (máximo, mínimo ou ponto de sela) e o algoritmo não vai ficar preso (não irá avançar). Isto deve-se a que nestes pontos, o declive (gradiente em ) corresponde a 0 e por isso:

[POR FORMULA INICIAL E SEU DESENVOLVIMENTO AQUI]

# 2.1)

|  |  |  |
| --- | --- | --- |
|  | a=2 | a=20 |
| .01 | 448 | 563 |
| .03 | 143 | 186 |
| .1 | 43 | Div |
| .3 | 13 | Div |
| 1 | Div | Div |
| 3 | Div | Div |
| Fastest | 0.6 | 0.095 |
| Divergence  threshold | 1 | 0.1 |

# 2.2)

Tal como acontece em com a parábola algo semelhante ocorre em para o paraboloíde.

Como se pode observar, com que não seja exagerado pequeno que hajam muitas iterações, ou grande demais que o algoritmo divirja, consegue-se chegar ao mínimo da função dada, sendo que este deverá ser maior para a maior. Outra coisa importante frisar, é que devido a no algoritmo de gradient descent, a cada iteração, o próximo x será sempre um ponto mais localizado num plano que tem a mesma direção que o gradiente, para o a=20, demorará mais iterações a convergir pois o gradiente não está tão direcionado, para o mínimo da função.

# 2.3)

Não [PORQUE NAO SEI BEM], porque ao aumentar a em não se está a dificultar a tarefa de convergir (tendo em conta que se escolheu um bom valor de ) enquante que neste caso, ao aumentar a largura do vale (a) tal tarefa é dificultada pois o algoritmo aproxima-se cada vez mais lentamente do centro devido à direção do gradiente. Assim quanto maior for maior será o número mínimo de iterações.

# 3.1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| .003 | >1000 | >1000 | >1000 | >1000 | >1000 |
| .01 | 563 | 558 | 552 | 516 | 448 |
| .03 | 186 | 181 | 178 | 115 | 172 |
| .1 | div | 48 | 35 | 91 | 122 |
| .3 | Div | Div | 29 | 83 | 92 |
| 1 | Div | Div | div | 92 | 146 |
| 3 | Div | div | Div | div | 147 |
| 10 | div | div | div | div | Div |
| Divergence  threshold | .1 | .3 | .567? | 1.9? | 3.9? |

# 3.2)