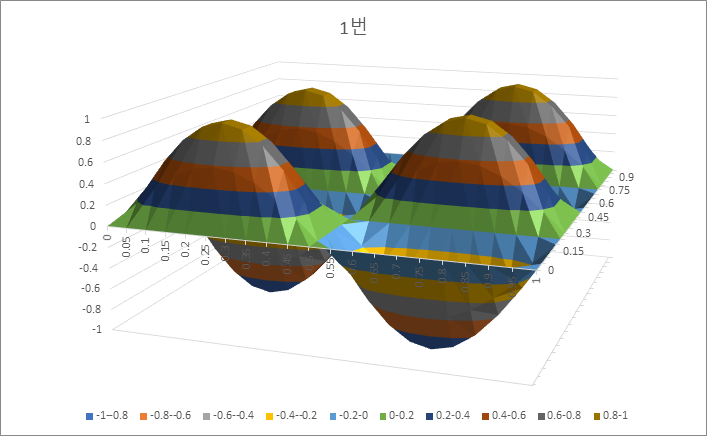
Gradient Descend

**1번)**



우선, 주어진 식 sin(2 \* 3.141592\*x)\*sin(4 \* 3.141592\*y)의 극점을 미분을 통해 구해보았다.

df(x,y)/dx = 0, df(x,y)/dy = 0을 동시에 만족하는 극점을 계산해보니

x가 0.25, 0.75 이고 y가 0.125, 0.375, 0.625, 0.875 일 때, 즉 총 8가지 점에서 극점을 가지는 것을 알 수 있었다.

Excel을 이용해 주어진 식에 x,y를 각각 0에서 1까지 0.05씩 증가시켜가며 대입해본 결과, 위의 그래프를 도출할 수 있었고, 위에서 계산을 통해 얻은 점들에서 최대, 최소 값을 가지는 것을 확인할 수 있었다. 확인된 최댓값은 0.95106, 최솟값은 –0.95106 이었다.

이를 코딩을 통해 확인해보고자 Gradient descending을 이용하였다.

psi = 0.03, eta = 0.001로 놓고 시작점을 적절히 바꾸어가며 실행해본 결과

49-th x=0.244052, y=0.119944 -> f(x,y)=0.997285

10-th x=0.2442 , y=0.370189 -> f(x,y)=-0.99751

10-th x=0.246258. y=0.620004 -> f(x,y)=0.997754

10-th x=0.246256, y=0.869993 -> f(x,y)=-0.997745

57-th x=0.743512, y=0.120143 -> f(x,y)=-0.997309

15-th x=0.74393, y=0.369998 -> f(x,y)=0.997299

57-th x=0.743522, y=0.619999 -> f(x,y)=-0.997199

61-th x=0.743634, y=0.869979 -> f(x,y)=0.997212

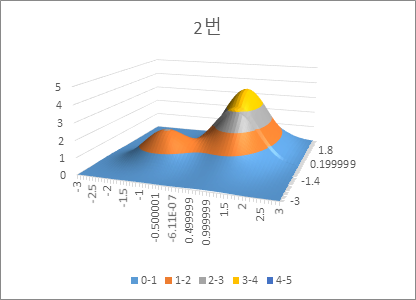
다음과 같이 x y가 8개의 극점과 비슷한 값으로 나타났고, 이를 주어진 식에 대입하니 앞서 확인했던 최대, 최소 값과 비슷한 값을 얻을 수 있었다.

하지만, 위의 극점을 도출하는 과정에서 시작점의 설정에 따라, 극점과 다른 점이 결과로 나타나거나, 0<x,y<1의 범위를 벗어나는 점이 결과도 나타나기도 했다.

위에서 그린 그래프에서 x,y의 범위를 확장 시키면 비슷한 형태의 그래프가 주위에도 나타날 것이고, 문제에서 x와 y의 범위를 설정하기는 했으나 코딩에서 직접적으로 범위에 제한을 두지 않았기 때문에 시작점에 따라 0<x,y<1 범위의 극점이 결과로 도출되지 않았을 것이다.

**2번)**

**2-1)**



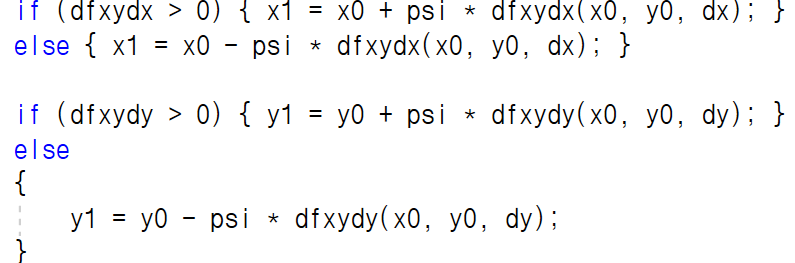
2번의 경우 2개의 peak를 가지며 그 점이 (1,1), (-1,-1)이라고 주어졌다.

첨부된 excel 파일을 통해 극점에서의 peak값을 확인해보니 약 4와 2임을 알 수 있었다.

그 후에, ppt에 주어진 gaussian 함수와 fxy를 1번에서 이용한 코드에 적용시키고 시작점을 바꿔가며 peak 점을 찾아보았다,

그러나 매우 다양한 시작점을 설정해보았음에도 주어진 peak 점을 결과로 얻을 수가 없었고 해당 점을 대입한 함수 값이 음수가 나오기도 했다.(그래프 상에선 음수 값을 갖지 않는다.)

고민해본 결과, 시작점이 조금만 어긋나도 peak가 아닌 다른 점으로 향한다고 판단하여



다음과 같이 코드를 조금 수정하고 다시 진행하였다.

starting point (-1,-1)

175-th 0.999062 0.948902

f(x,y) = 3.99821

starting point (0,0)

185-th -0.946976 -0.980633

f(x,y) = 2.02751

starting point (-2,-2)

152-th -0.991063 -1.00835

f(x,y) = 2.0274

starting point (2,2)

167-th 0.999243 1.04726

f(x,y) = 3.99832

그 결과, 극점 (-1,-1),(1,1)일 때 peak값 약 4와 2가 도출되는 것을 확인할 수 있었다.

**2-2)**

다음은 eta값에 따른 x y의 변화, 함수값, computation time의 변화를 알아보기 위해 시작점을 고정하고 psi = 0.01로 두고, eta 값을 바꾸어보았다.

psi=0.01 eta=0.01

59-th 0.954359 0.508984

f(x,y) = 3.76269

computation time = 1334614 ns

psi=0.01 eta=0.008

69-th 0.977955 0.593451

f(x,y) = 3.83938

computation time = 1524054 ns

psi=0.01 eta=0.005

94-th 0.995717 0.749575

f(x,y) = 3.93945

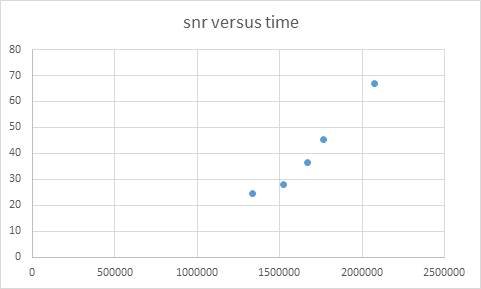
computation time = 1666276 ns

psi = 0.01, eta = 0.003

120-th 0.998452 0.849873

f(x,y) = 3.97872

computation time = 1766400 ns



eta값이 줄어들수록 computation time은 줄어들고 f(x,y)값은 점점 더 4에 가까워지는 것을 확인 할 수 있었다. 위 결과값에 대하여 x축을 computation time으로 두고 y축을 SNR로 두었을 때, 다음과 같은 그래프를 얻을 수 있었다. 즉, eta 값이 작아질수록 계산하는데 걸리는 시간은 길어지지만, 정확도는 더욱 더 커지게 된다. 따라서 노이즈는 작아지고 계산식에 따라 SNR값은 커진다.

**2-3)**

다음은 psi값에 따른 x y의 변화, 함수값, computation time의 변화를 알아보기 위해 시작점을 고정하고 eta = 0.001로 두고, psi 값을 바꾸어보았다.

psi=0.01 eta=0.001

175-th 0.999062 0.948902

f(x,y) = 3.99821

computation time = 276694 ns

psi=0.03 eta=0.001

77-th 0.954359 0.508984

f(x,y) = 4.00043

computation time = 233700 ns

psi = 0.04, eta = 0.001

61-th 0.998956 0.911486

f(x,y) = 4.00054

computation time = 210800 ns

psi = 0.05, eta = 0.001

51-th 0.999129 0.960255

f(x,y) = 4.00059

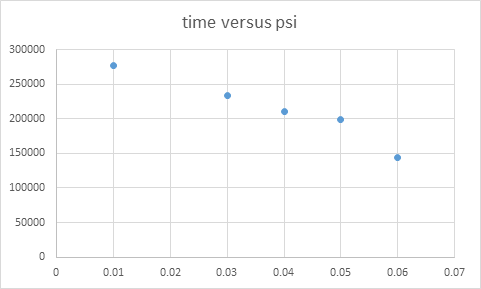
computation time = 198700 ns

psi=0.06, eta=0.001

44-th 0.999062 0.948902

f(x,y) = 4.00062

computation time = 143300 ns



psi 값이 커질수록 computation time 값은 줄어드는 것을 확인할 수 있었다.

Performance = (accuracy) / (computation time) 이다. 위의 도출된 결과를 보았을 때 psi값이 커질수록 performance 값도 증가함을 알 수 있다.(함수값은 약 4로 거의 동일하므로 accuracy는 거의 동일할 것이기 때문이다.)

여기서 psi 값이 더 커져서 1에 가까워진다면 어떤 상황이 발생하는지 알아보았다.

psi=0.1

29-th 0.998956 0.911486

f(x,y) = 4.00066

computation time = 133100 ns

psi=0.2

15-th 0.999062 0.948902

f(x,y) = 4.00067

computation time = 91600 ns

psi=0.3

238-th –2.06101, 1.73888

f(x,y) = 0.000655854

computation time = 819100 ns

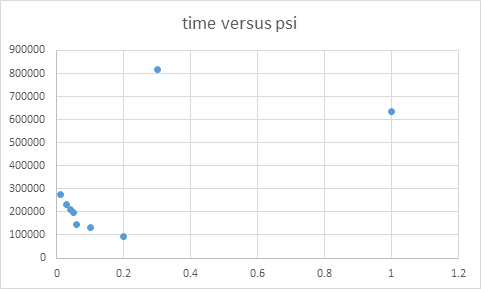
psi=1

181-th –2.89549, 1.41548

f(x,y) = 0.000161955

computation time = 633900 ns

앞에서 얻은 결론과 달리 psi값이 너무 커지게 되면 정확성이 매우 떨어지고, iteration 값이 매우 증가하며 동시에 computation time이 급격히 증가하는 것을 확인할 수 있었다.



위의 문제에서도 보았듯이 정확성을 얻기 위해선 계산이 복잡해져서 속도를 포기해야 하고, 이 속도를 위해서는 비용이 들기 마련이다. 그렇다고 어느 한 쪽을 포기하는 것은 적절한 performance라고 보기 어려울 것 이다. 결국 benefit과 cost를 적절히 타협하여 그에 맞는 eta와 psi를 설정해야만 한다.

지금까지 위에서 얻은 결과들중에서는 psi = 0.2, eta = 0.001일때가 accuracy아 cost를 고려했을 때 가장 좋은 performance를 갖는다고 생각한다.