Experimental Mathematics, Homework 13

Jingze Li, RUID 195000313

OK to post homework Jingze Li(RUID:195000313),03/10/19
The following is my idea on the implementation of gradient method:

1. I think we should adopt gradient without normalisation. The reason is that: when our points approaches to the desires minimum, the gradient is usually small in norm and thus the points get closer and closer, finally converge. However, if we normalise it, every step we get the same length, this will result in oscillation around the minimum.

   Thus I think that the procedure G2(f,var,pt,sz,MaxI,tol) actually works nicely. The anomaly in class is caused by inadequate choice of starting point and the function to minimize.

   I explain my idea in the following parts.

2. In class we use polynomials in the form

   $f := x^2*y^2*z^2+x^2*y^2*z+x^2*y*z^2+x*y^2*z^2+x^2*y*z+x*y^2*z+x*y*z^2+x*y*z$

   Now we see if there IS a minimum for this kind of functions.

   To visualize and to be simple, we study polynomial with 2 variables, namely x and y.

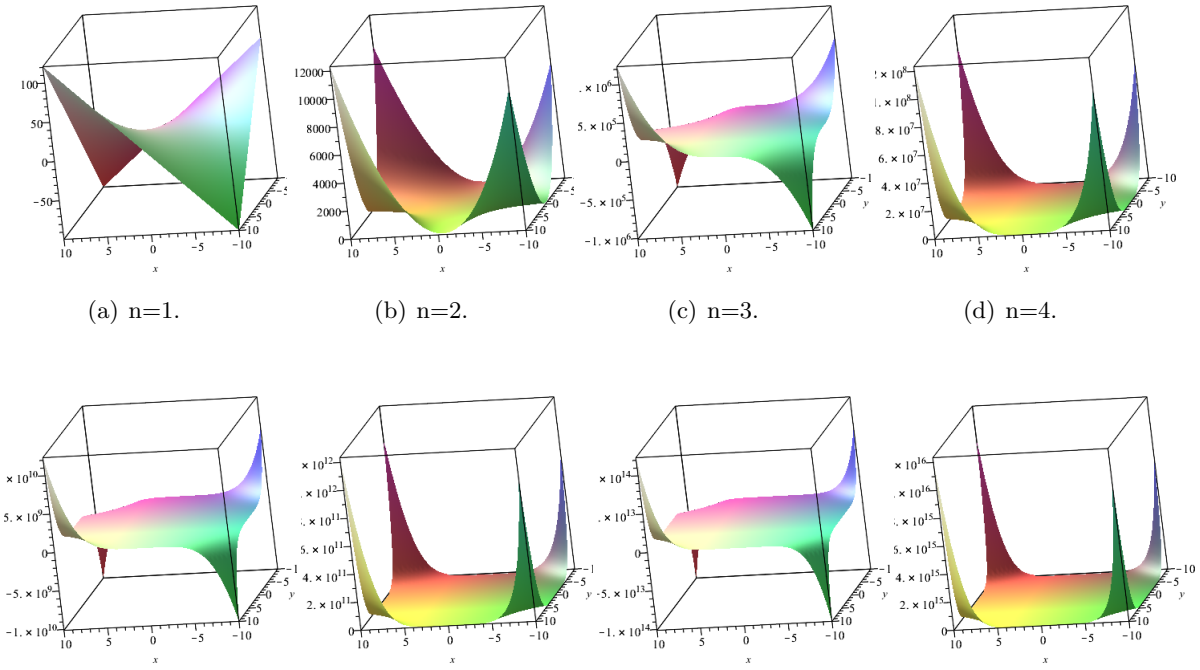
```
n :=2:
f:=add(add(x^i*y^j,i=0..n),j=0..n):
```

   then we plot them

```
plot3d(f)
```

   we give the following picture in order of the choice of n, which is the degree of f

   we notice that for even n we have a minimum near the origin, but for odd n, the point near the origin is very likely to be a saddle(yet to prove), so that could be a reason for the non convergence of our G2 procedure in some situations.



(a) n=1. (b) n=2. (c) n=3. (d) n=4.

(e) n=5. (f) n=6. (g) n=7. (h) n=8.

Figure 1: pics

3. As for the starting guess pt, it is very important that this point must be close enough to the actual minimum, or more precisely, the function must be convex in the area around these two points. I give the following example:

```
   n :=6:
f:=add(add(x^i*y^j,i=0..n),j=0..n):
G2(f,[x,y],[1,-1],0.1,1000,0.001)
            [-Float(infinity), Float(infinity)]
```

Very bad! The procedure does not work! However, try the following:

```
   n :=6:
f:=add(add(x^i*y^j,i=0..n),j=0..n):
G2(f,[x,y],[0.1,0.1],0.1,100000,0.00001)
               [-0.6703291817, -0.6703291815]
```

If we change the number of Maximum iteration, the output does not change; if we decrease tol, we get a point near $[-0.6703291817, -0.6703291815]$. This suggests convergence.

Why? Obviously $[0.1, 0.1]$ is closer to the minimum than $[1, -1]$. Try the following:

```
   with(LinearAlgebra):
   with(Student[VectorCalculus]):
  H:=Hessian(f,[x,y]):
H:=eval(H,[x=2,y=2])
                        [ 89154 103041]
                   H := [              ]
                        [103041  89154]
Eigenvalues(H)
                          [192195]
                          [     ]
                          [-13887]

H:=eval(H,[x=0.1,y=0.1])
                     [3.047777473 1.524138394]
               H := [                        ]
                     [1.524138394 3.047777473]
Eigenvalues(H)
                   [4.57191586700000 + 0. I]
                   [                       ]
                   [1.52363907900000 + 0. I]
```

This shows that

$$f := x^6*y^6+x^6*y^5+x^5*y^6+x^6*y^4+x^5*y^5+x^4*y^6+x^6*y^3+x^5*y^4+x^4*y^5+x^3*y^6+x^6*y^2+x^5*y^3+x^4*y^4+x^3$$

is not convex at the point $[1, -1]$. Surely we cannot get convergence!

4. thus I think that our G2 works, if we choose the function and starting point properly.

   By the way, to test $G2$, if the iteration number reaches $MaxC$, we can print a line " fail to convergence within the given tol" , which is a good way to warn that something went wrong.