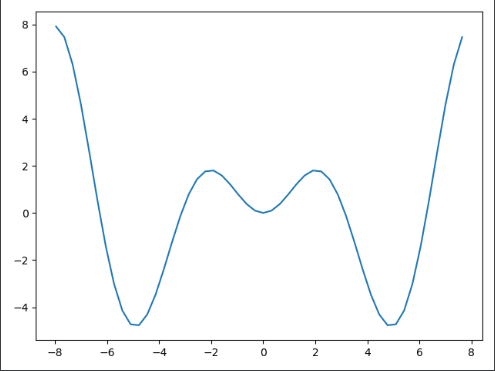
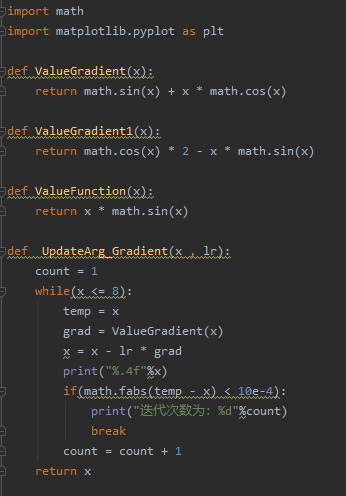
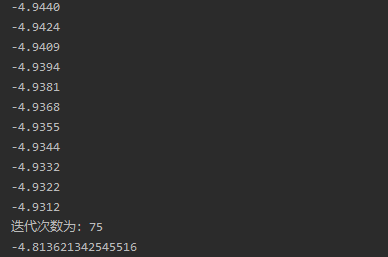
画出函数y = x \* sinx图像



* **定义的梯度下降求最小值的函数：**

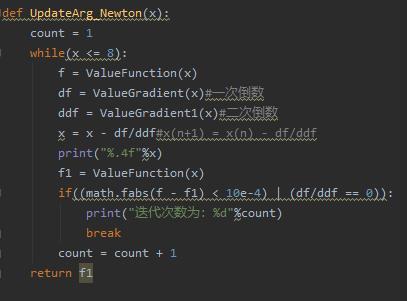


学习率lr = 0.01时，运行结果如下

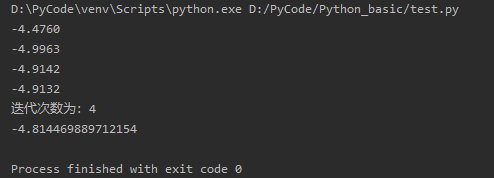


迭代次数为75次，最小值的点x=-4.9312，此时y = -4.8136

* **定义的牛顿法求最小值的函数：**



运行结果如下：

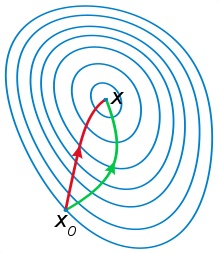


迭代次数为4次，最小值点为x=-4.9132,最小值为y=-4.8144

**分析：**

梯度下降法与牛顿法所求的最小值基本一致，但梯度下降法的迭代次数为75次，而牛顿法的迭代次数为4次。

牛顿法之所以比梯度下降法迭代次数少的原因是因为牛顿法引入了二阶导数，而梯度下降引入的是一阶导数，也就是说每次在选择梯度时，牛顿法不仅考虑下降坡度最大的点，它还会将选择该点下降后坡度会不会变得更大（个人理解就是把斜率最大的点和斜率变化最快的方向都考虑了进去）牛顿法考虑的是全局的问题，而梯度下降考虑的是局部问题，通过局部坡度最大点，慢慢向极值点逼近。



如图所示，红色路径为牛顿法的迭代路径，绿色是梯度下降法的迭代路径

参考资料：https://blog.csdn.net/qq\_17754181/article/details/50095075