1、 可视化函数

首先，我们可以利用 MATLAB 中的平面等高线函数 contour 对我们的函数进行可视化处理，我们可以粗略的判断，函数的极小值肯定分布在原点附近（针对整个平面而言），因此我们就只须画出定义域在 $x \in [-2,2], y \in [-2,2]$ 上的图像，代码参考 visualization.m，其运行结果等高线图如下所示。

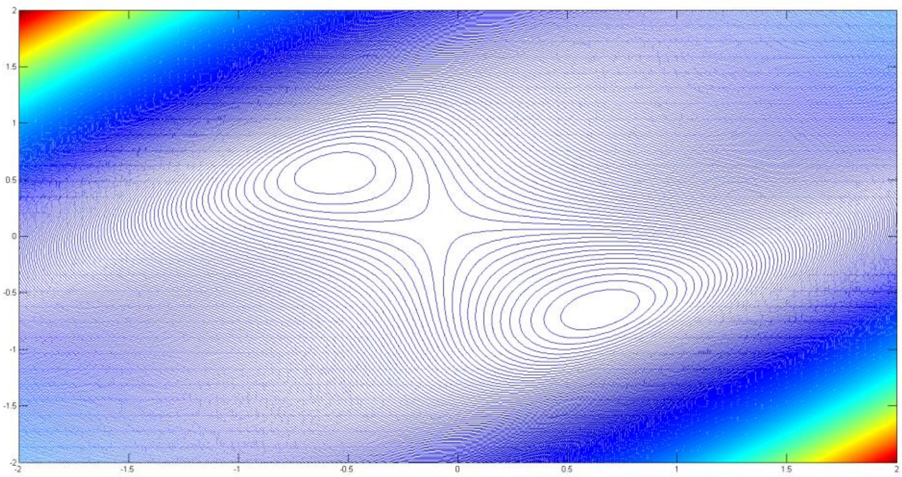

图 1 在特定定义域上的等高线图

因此我们就可以回答第一个问题，该函数有两个局部最小点，具体是多少以及哪个是全局最优点，我们可以利用 Nelder-Mead Simplex 算法进行求解。

2、 算法求解

根据 Nelder-Mead Simplex 算法的原理以及参考后文给出的参考资料后，自行编写代码，可自行参考源文件 NMS.m。NMS 函数调用格式如下：

[x,fval,flag,time,object_value]=NMS(fun,x0,max_time,eps)

Input:

% fun: object function

% x0: starting point

% max_time: the max number of iteration ,the default value is 10000

% eps: accuracy, the default value is 1e-5

Output:

% x: target point

% fval: minimal value

% flag: indicator of the algorithm is successful or not

% time: the max number of iteration, the default value is 10000

% object_value: matrix which include minimal value in the each iteration

最后我们给定三个初始点 $x_1(0.6, 0.6)$ 、$x_2(-0.8,1)$ 、$x_3(0.5,1.7)$，其中前两个是作业文件中给定的，而第三个点是我自行添加的。参考源程序 main.m，运行程序，我们可以得到以下的结果。

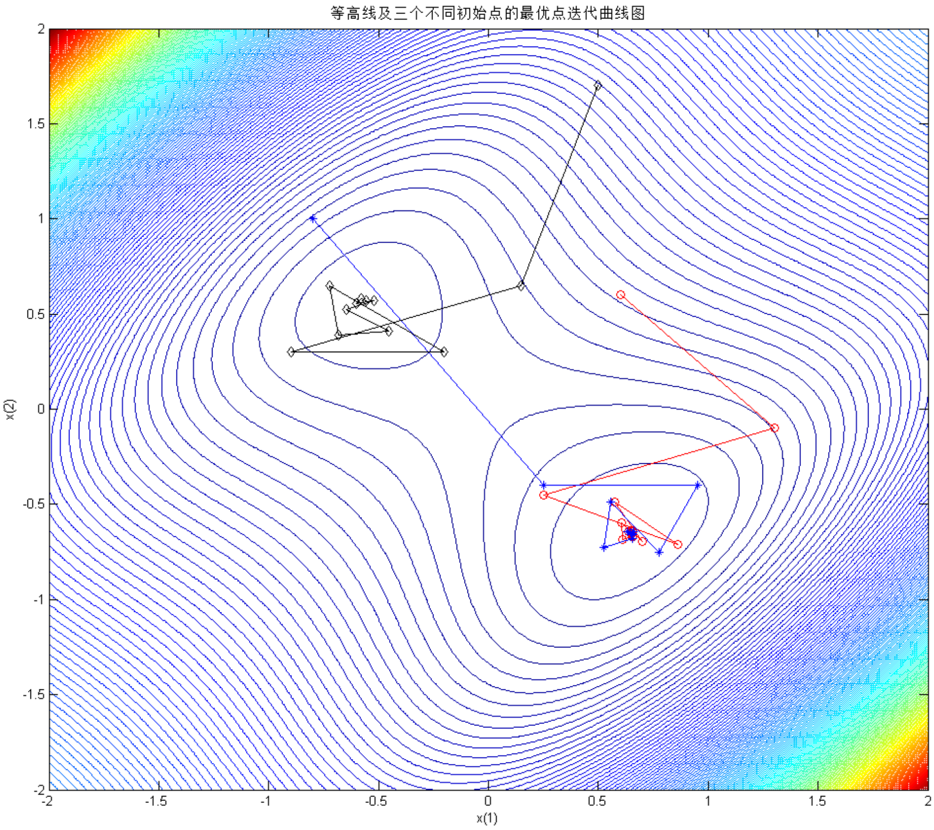

图 2 不同初始点的迭代曲线

说明：红色和蓝色曲线分别对应第一个初始点和第二个初始点，发现它们都收敛到同一局部最优点 $[0.6504, -0.6504]$ ，最小值为-6.5139；而黑色曲线则是对应第三个初始点，它单独收敛到另一个局部最优点 $[-0.5655, 0.5655]$，最小值为-4.0702。因此第一个初始点和第二个初始点所收敛的最优点是此函数的全局最优点，其最小值为-6.5139。

3、 结果验证

与 MATLAB 中内置的函数 fminsearch 对比发现，初始点 $x_2(-0.8,1)$ 并不是收敛到全局最优点，而是收敛到了另一个局部最优点。原因应该在于我的代码和内置代码在构造初始单纯形的方法有一定差异导致。但这并不会在根本上影响到对问题的求解。

4、 参考资料

[1] http://www.360doc.com/content/14/0405/10/13256259_366532138.shtml

[2] MATLAB 中 fminsearch 函数源代码