2.1 作业 1

Perform three iterations to find minimum of $f(x_1, x_2) = (x_1 - 3)^4 + (x_1 - 3x_2)^2$, initial point $x_0 = [0, 0]^T$. 要求用以下四种方法求解 (**用 Matlab 实现，把迭代过程自己实现一下**):

- ★ **Newton's method**
- ★ **Levenberg-Marquardt's method**
- ★ **Broyden-Fletcher-Goldfarb-Shanno's method**
- ★ **Davidon-Fletcher-Powell's method**

Newton’s method
牛顿法：具体如下表所示，代码部分详见 Newton.m（就是助教上传的代码）

迭代次数	x_k^T	$f(x_k)$	$\nabla f(x_k)$	$H(x_k)$
0	$(0,0)$	81.0	$(-108,0)$	$\begin{bmatrix} 110 & -6 \\ -6 & 18 \end{bmatrix}$
1	$(1.00,0.33)$	16.0	$(-32,0)$	$\begin{bmatrix} 50 & -6 \\ -6 & 18 \end{bmatrix}$
2	$(1.67,0.56)$	3.16	$(-9.48,0)$	$\begin{bmatrix} 23.33 & -6 \\ -6 & 18 \end{bmatrix}$
3	$(2.11,0.70)$	0.62	$(-2.80,0)$	$\begin{bmatrix} 11.48 & -6 \\ -6 & 18 \end{bmatrix}$

Levenberg-Marquardt’s method
LM 方法，修改代码的部分，主要见 LM.m，其中 $\lambda = 0.0001$，取值非常小，实际结果和原来牛顿法没有啥区别。

迭代次数	x_k^T	$f(x_k)$	$\nabla f(x_k)$	$H(x_k)$
0	$(0,0)$	81.0	$(-108,0)$	$\begin{bmatrix} 110 & -6 \\ -6 & 18 \end{bmatrix}$
1	$(1.00,0.33)$	16.0	$(-32,0)$	$\begin{bmatrix} 50 & -6 \\ -6 & 18 \end{bmatrix}$
2	$(1.67,0.56)$	3.16	$(-9.48,0)$	$\begin{bmatrix} 23.33 & -6 \\ -6 & 18 \end{bmatrix}$
3	$(2.11,0.70)$	0.62	$(-2.80,0)$	$\begin{bmatrix} 11.48 & -6 \\ -6 & 18 \end{bmatrix}$

Broyden-Fletcher-Goldfarb-Shanno' s method
BFGS 方法，修改代码部分，具体见 BFGS.m

迭代次数	x_k^T	$f(x_k)$	$\nabla f(x_k)$	$H(x_k)$
0	$(0,0)$	81.0	$(-108,0)$	$\begin{bmatrix}110 & -6\\ -6 & 18\end{bmatrix}$
1	$(1.00,0.33)$	16.0	$(-32,0)$	$\begin{bmatrix}78 & -6\\ -6 & 18\end{bmatrix}$
2	$(1.42,0.47)$	6.22	$(-15.75,0)$	$\begin{bmatrix}40.60 & -6\\ -6 & 18\end{bmatrix}$
3	$(1.82,0.61)$	1.88	$(-6.42,0)$	$\begin{bmatrix}24.85 & -6\\ -6 & 18\end{bmatrix}$

Davidon-Fletcher-Powell' s method
DFP 方法，修改代码部分，具体见 DFP.m

迭代次数	x_k^T	$f(x_k)$	$\nabla f(x_k)$	$D(x_k)$
0	$(0,0)$	81.0	$(-108,0)$	$\begin{bmatrix}0.0093 & 0.0031\\ 0.0031 & 0.0566\end{bmatrix}$
1	$(1.00,0.33)$	16.0	$(-32,0)$	$\begin{bmatrix}0.0224 & 0.0075\\ 0.0075 & 0.0580\end{bmatrix}$
2	$(1.71,0.57)$	2.70	$(-8.44,-0.0021)$	$\begin{bmatrix}0.0528 & 0.0176\\ 0.0176 & 0.0614\end{bmatrix}$
3	$(2.16,0.72)$	0.49	$(-2.34,-0.0042)$	$\begin{bmatrix}0.1256 & 0.0417\\ 0.0417 & 0.0694\end{bmatrix}$

2.2 作业 2

Using golden search method to find the value of x that minimize $f(x) = -min\{\frac{x}{2}, 2-(x-3)^2, 2-\frac{x}{2}\}$. **(手写编程都可以，要求给出每次迭代的 x 值和 f 值)**

- Function is unimodal on [0,8]
- Perform five iterations
- Compare results with that of Fibonacci method and a fixed step method (take a step length $\Delta s = 2$)

具体部分见 python 文件 GoldSecFibFixSetp.py，需要 numpy 和 matplotlib 包，实现了三种算法，下面依次展示。

Golden Searchmethod 黄金分割法：

迭代次数	(a, b, c, d)	$(f(a), f(b), f(c), f(d))$
0	$(0, 3.06, 4.94, 8.0)$	$(7, -0.47, 1.78, 23)$
1	$(0, 1.89, 3.06, 4.94)$	$(7, -0.76, -0.47, 1.78)$
2	$(0, 1.17, 1.89, 3.06)$	$(7, 1.36, -0.76, -0.47)$
3	$(1.17, 1.89, 2.33, 3.06)$	$(1.36, -0.76, -0.83, -0.47)$
4	$(1.89, 2.16, 2.33, 2.61)$	$(-0.76, -0.92, -0.83, -0.70)$
5	$(1.89, 2.06, 2.16, 2.33)$	$(-0.76, -0.97, -0.92, -0.83)$

选取每一次迭代的最小点和对应的 x 值，绘制图 1

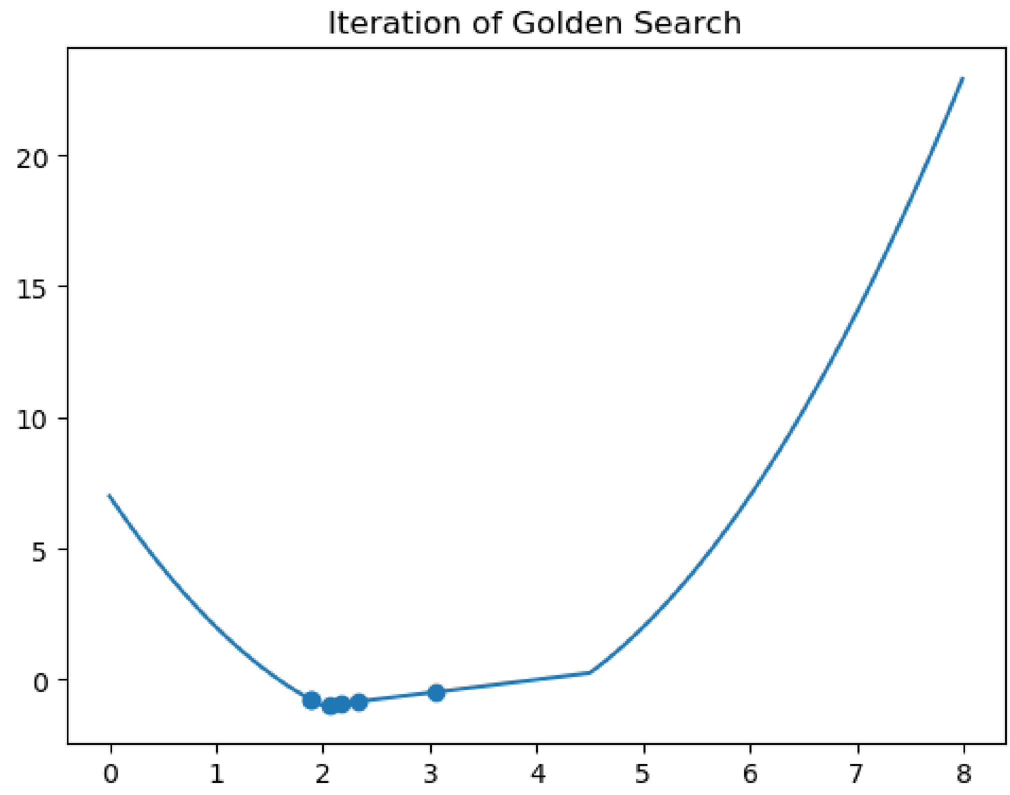

图 1: 黄金分割法的最小点

Fibonacci method
裴波那契法，由于计算了相关的迭代次数，这里设置的精度 $\epsilon = 0.1$，计算得到 $n = 11$，最终计算得出需要迭代 $(n - 2 = 9)$ 次，每次的更新值如下函数输出所示：

a,b,c,d 的值 (9 次迭代, 包括初始值共 10 组数据)

迭代次数	(a,b,c,d)	$(f(a),f(b),f(c),f(d))$
0	(0, 3.05, 4.94, 8)	(7, −0.47, 1.78, 23)
1	(0, 1.88, 3.05, 4.94)	(7, −0.76, −0.47, 1.78)
2	(0, 1.16, 1.886, 3.05)	(7, 1.36, −0.76, −0.47)
3	(1.16, 1.88, 2.33, 3.05)	(1.36, −0.76, −0.83, −0.47)
4	(1.88, 2.16, 2.33, 2.61)	(−0.76, −0.91, −0.83, −0.69)
5	(1.88, 2.06, 2.16, 2.33)	(−0.76, −0.96, −0.91, −0.83)
6	(1.88, 1.99, 2.06, 2.16)	(−0.76, −0.98, −0.96, −0.91)
7	(1.88, 1.957, 1.99, 2.06)	(−0.76, −0.91, −0.98, −0.96)
8	(1.95, 1.99, 2.02, 2.06)	(−0.91, −0.98, −0.98, −0.96)
9	(1.99, 2.00, 2.00, 2.02)	(−0.98, −0.99, −0.99, −0.98)

选取每一次迭代的最小点和对应的 x 值，绘制图 2

fixed step method (take a step length s = 2)
步长选的太好了，一次迭代就收敛了。具体如表所示：

迭代次数	x	$f(x)$
0	0	7
1	2	−1.0

取每一次的迭代值，绘制图 3。

结果最终对比：斐波那契方法可以通过精度限制迭代次数，这样比黄金分割的方法精度更高，也方便实际的操作，黄金分割算法没有办法确定实际的迭代次数，而且理论上讲可能会震荡，可变步长的裴波那契方法既是继承了黄金分割方法的点的迭代方式（每一次只迭代一个点），有保证了精度，而对于固定步长法，步长需要认为设定，这样会很繁琐，可能会导致震荡。

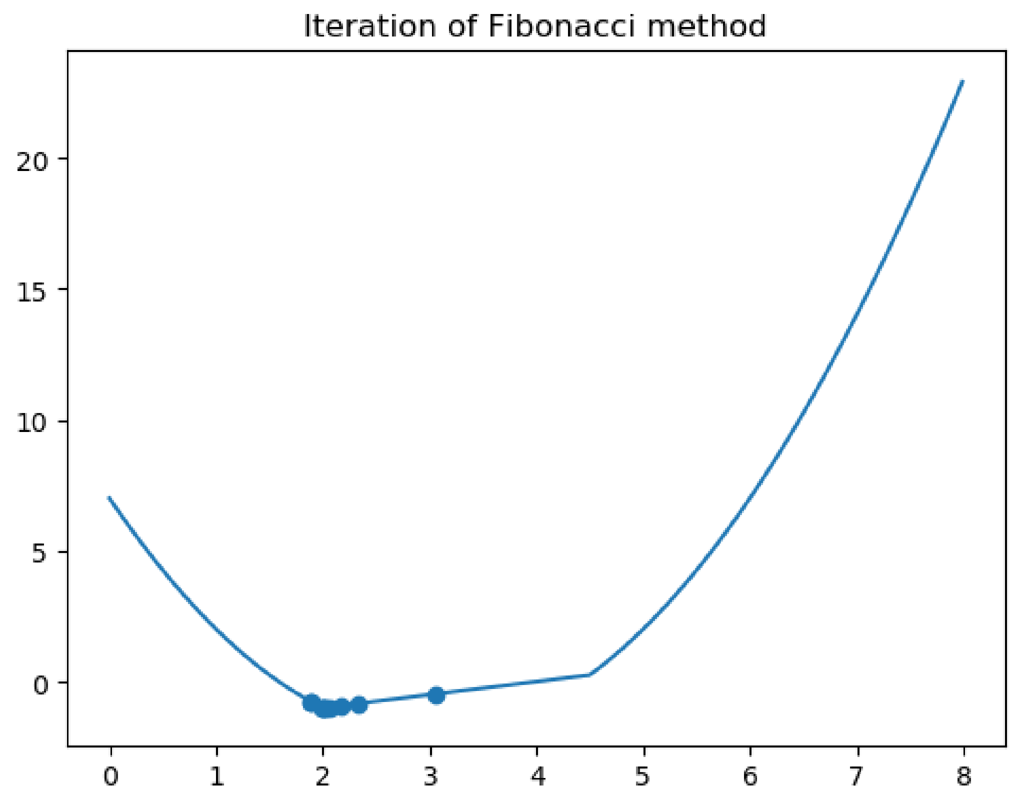

图 2: 裴波那契法的最小点

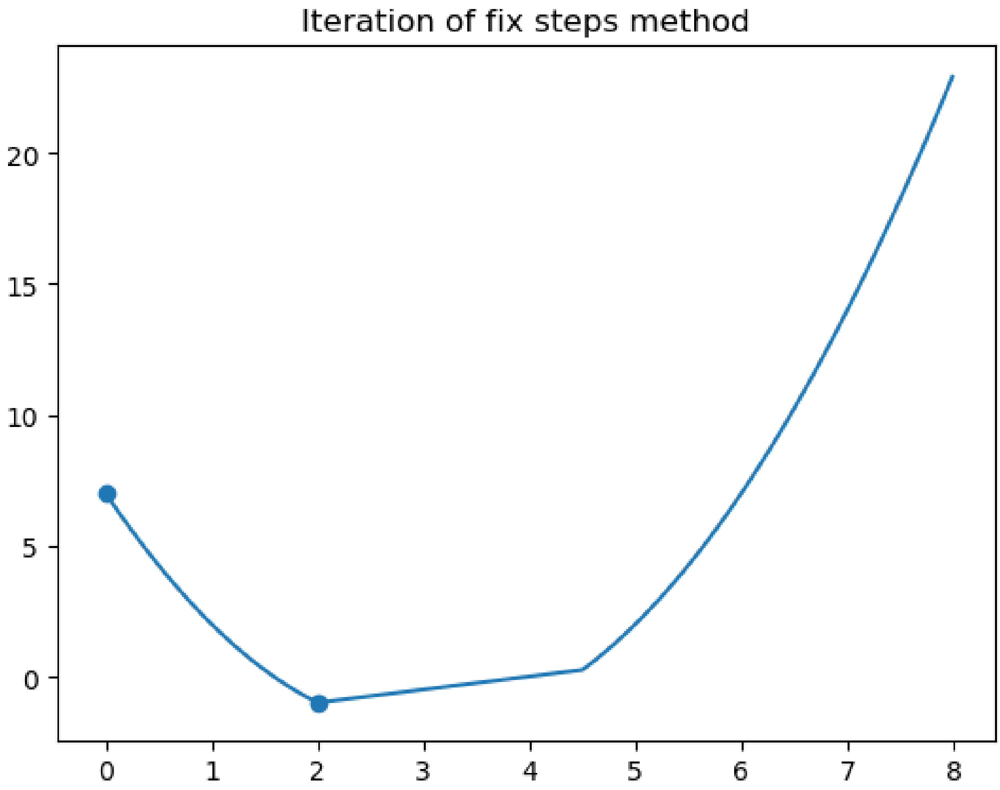

图 3: 固定步长法的最小点