最优化第三次上机作业

王恒亮　学号：1601210102

本次实现了解决 Minimax Problems 的三种方法，分别为 Least pth Method[?], Smoothing Method[?] 和 Adaptive Smoothing Method[?]。以下分别介绍算法及程序实现，实验结果及分析。

1 算法及程序实现

1.1 Least pth Method

Least pth Method 主要通过优化 U 函数来解决 Minimax 问题，U 函数的定义如下：

$$U(x,u,p,\xi)=\begin{cases} M(x,\xi)(\sum\limits_{i\in S(x,\xi)} u_i(\frac{f_i(x)-\xi}{M(x,\xi)})^q)^{1/q}, & \text{for} M(x,\xi)\neq 0, \\ 0, & \text{for} M(x,\xi)=0 \end{cases} \tag{1}$$

其中:

$$u_i \geq 0 \quad \text{and} \quad \sum_{i\in I} u_i = 1 \tag{2}$$

$$M(x,\xi)=\max_{i\in I}(f_i(x)-\xi) \tag{3}$$

算法的框架如下：

步 1 取 $p>1,\xi^{(1)}=0,u_i^{(1)}=1,i\in I,x=x^0,r=1$

步 2 固定 u,p,ξ，在公式 1中对 x 取极小，最优解设为 x^r

步 3 更新 u

$$u_i^{r+1}\frac{v_i^{r+1}}{\sum_{j\in I} v_j^{r+1}}, i\in I$$

其中：

$$v_i^{r+1} = \begin{cases} u_i^r(\dfrac{f_i(x^r)-\xi}{M(x^r,\xi)})^{q-1}, for i \in S(x^r,\xi) \\ 0 for i \in I - S(x^r,\xi) \end{cases}$$

$$S(x^r,\xi) = \begin{cases} \{i|f_i(x)-\xi>0, i\in I\}, if M(x^r,\xi) \geq 0 \\ I, if M(x^r,\xi) < 0 \end{cases}$$

步 4 更新 ξ

$$\xi^{r+1} = \begin{cases} \xi^r + M(x^r,\xi^r), if M(x^r,\xi) < 0 \\ \xi^r + \lambda M(x^r,\xi^r) \end{cases}$$

步 5 更新 $p, p = cp, c \geq 1$，转至步 2

在实现中，除了计算出了最优点和最优点的函数值之外，还记录了迭代次数，函数调用次数，具体的参数设置将在结果中说明。

1.2 Smoothing Method

Smoothing Method 是将 Minimax 问题转化为优化近似函数 $f(x,\mu)$，其定义如下：

$$f(x,\mu) = \mu \ln \sum_{i=1}^{m} \exp(\frac{f_i(x)}{\mu})$$

对于 $f(x,\mu)$ 的优化使用牛顿法求出下降方向，使用 Armijo 准则找到步长，在每步迭代后会更新 $\mu, \mu = \beta\mu$。对于 $f(x,\mu)$ 的函数，梯度和 Hessian 矩阵的计算如下：

$$\begin{aligned} f(x^k,\mu_k) =& f(x^k) + \mu_k \ln \sum_{i=1}^{m} exp(\frac{f_i(x^k)-f(x^k)}{\mu_k}) \\ \bigtriangledown_x f(x,\mu) =& \sum_{i=1}^{m} \lambda_i(x,\mu) \bigtriangledown f_i(x) \\ \bigtriangledown_x^2 f(x,\mu) =& \sum_{i=1}^{m}(\lambda_i(x,\mu) \bigtriangledown^2 f_i(x) + \frac{1}{\mu}\lambda_i(x,\mu) \bigtriangledown f_i(x) \bigtriangledown f_i(x)^T) \\ & - \frac{1}{mu}(\sum_{i=1}^{m} \lambda_i(x,\mu) \bigtriangledown f_i(x))(\sum_{i=1}^{m} \lambda_i(x,\mu) \bigtriangledown f_i(x))^T \\ \lambda_i(x^k,\mu_k) =& \frac{exp(\frac{f_i(x^k)-f(x^k)}{\mu_k})}{\sum_{j=1}^{m} exp(\frac{f_j(x^k)-f(x^k)}{\mu_k})} \end{aligned}$$

1.3 Adaptive Smoothing Method

Adaptive Smoothing Method 是在 Smoothing Method 的基础对算法做了改进，主要解决了在 $\mu \to 0$ 时产生的 ill-condition 问题。Adaptive Smoothing Method 将 Minimax 问题转化为优化公式 4。

$$\Phi_p(x) = \log(\sum_{j\in Q}(\exp(pf^j(x)))/p \tag{4}$$

具体算法如下：

步 1 取 $i=0, k=0, \gamma=1$

步 2 设 $\Phi_{p_i,xx}(x_i)$ 是 Hessian 矩阵 R 的 Cholesky 修正，$c(R)$ 为 R 的条件数的倒数，则若 $\Phi_{p_i,xx}(x_i)$ 正定且 $c(R)\geq k_1, p_i \leq k_3$，转步 3，若 $\Phi_{p_i,xx}(x_i)$ 正定且 $c(R)\geq k_1$ 且 $\Phi_{p_i,xx}(x_i)$ 的最大特征值 $\sigma_{p_i,max}(x_i)$ 满足 $\sigma_{p_i,max} \leq k_2$，转步 3，否则转步 4

步 3 $h_{p_i}(x_i) = -\Phi_{p_i,xx}(x_i)^{-1} \bigtriangledown \Phi_{p_i}(x_i)$，转步 5

步 4 $h_{p_i}(x_i) = -\bigtriangledown_{p_i}(x_i)$

步 5 计算步长

$$\lambda_{p_i}(x_i) = \max_{l\in N}\{\beta^l|\Phi_{p_i}(x_i+\beta^l h_{p_i}(x_i)) - \Phi_{p_i}(x_i) \leq \alpha\beta^l < \bigtriangledown\Phi_{p_i}(x_i), h_{p_i}(x_i) >\}$$

步 6 $x_{i+1} = x_i + \lambda_{p_i}(x_i)h_{p_i}(x_i)$

步 7 若 $|\bigtriangledown\Phi_{p_i}(x_i)|^2 \leq \tau(p_i)$，转步 8，否则设 $p_{i+1}=2p_i, i=i+1$，转步 2

步 8 若 $p^* \leq \hat{p}, \gamma=1$, 其中 p^* 满足 $\epsilon_a(p_i) \leq |\bigtriangledown\Phi_{p^*}(x_i)|^2 \leq \epsilon_b(p_i)$，则设 $\gamma = \max\{2, (\hat{p}+2)/(k+1)\}, p_{i+1}=\gamma(k+2), k=k+1, i=i+1$，否则设 $p_{i+1}=\gamma(k+2), k=k+1, i=i+1$，转步 2

2 实验结果及分析

本次实验分别在 Xu[?] 中实验中第 2,3 题中的函数和 [?] 中的 Digital filter example 滤波器函数进行结果测试。实验结果如下：

2.1 Xu[?] 中例子 2

问题描述如下：

$$minimize_{i=1,2,3} f_i(x)$$

其中：

$$f_1(x) = x_1^4 + x_2^2 \tag{5}$$

$$f_2(x) = (2 - x_1)^2 + (2 - x_2)^2 \tag{6}$$

$$f_3(x) = 2\exp(-x_1 + x_2) \tag{7}$$

初始点取值为 $(1, -0.1)$，最优点取值为 $(1, 1)$。各个方法的数值结果如表 1。

表 1: Xu 中例子 2

方法	$\|f^* - f\|$	迭代次数	函数调用次数
Least pth	7.2482×10^{-5}	24	72
Smooth Method	4.3247×10^{-7}	28	57
Adaptive Smooth	4.7311×10^{-5}	43	167

2.2 Rosen-Suzuki Problem

问题描述为：

$$F(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4 \tag{8}$$

$$g_2(x) = -x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8 \tag{9}$$

$$g_3(x) = -x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10 \tag{10}$$

$$g_4(x) = -x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5 \tag{11}$$

$$f_2 = F(x) - \alpha_2 g_2(x) \tag{12}$$

$$f_3 = F(x) - \alpha_3 g_3(x) \tag{13}$$

$$f_4 = F(x) - \alpha_4 g_4(x) \tag{14}$$

$$\alpha_2 = \alpha_3 = \alpha_4 = 10 \tag{15}$$

初始值取值为 $(0, 0, 0, 0)$，最优解 $x^* = (0, 1, 2, -1)$。各个方法的数值结果如表 3。

表 2: Rosen-Suzuki Problem 数值结果

方法	$\lvert f^* - f\rvert$	迭代次数	函数调用次数
Least pth	3.6745×10^{-5}	13	39
Smooth Method	1.4665×10^{-10}	47	147
Adaptive Smooth	7.5858×10^{-11}	41	113

2.3 Digital filter example

滤波器函数主要参考了 [?] 中的函数。滤波器函数定义如下：

$$|H(x,\theta)| = A\prod_{k=1}^{K}\frac{N_k}{D_k} \tag{16}$$

$$A\prod_{k=1}^{K}(\frac{1+a_k^2+b_k^2+2b_k(2\cos^2\theta-1)+2a_k(1+b_k)\cos\theta}{1+c_k^2+d_k^2+2d_k(2\cos^2\theta-1)+2c_k(1+d_k)\cos\theta})^{1/2} \tag{17}$$

其中：

$$x = [a_1, b_1, \cdots, c_K, d_k, A]^T \tag{18}$$

$$n = 4K+1 \tag{19}$$

则对其求导可得 $|H(x,\theta)|$ 对 x 中各个变量的导数。

$$\frac{\partial |H(x,\theta)|}{\partial a_j} = \frac{|H(x,\theta)|}{N_j^2}[a_j + (1+b_j)\cos\theta] \tag{20}$$

$$\frac{\partial |H(x,\theta)|}{\partial b_j} = \frac{|H(x,\theta)|}{N_j^2}[-1 + 2\cos^2\theta + a_j\cos\theta + b_j] \tag{21}$$

$$\frac{\partial |H(x,\theta)|}{\partial c_j} = -\frac{|H(x,\theta)|}{D_j^2}[c_j + (1+d_j)\cos\theta] \tag{22}$$

$$\frac{\partial |H(x,\theta)|}{\partial d_j} = \frac{|H(x,\theta)|}{D_j^2}[-1 + 2\cos^2\theta + c_j\cos\theta + d_j] \tag{23}$$

$$\frac{\partial |H(x,\theta)|}{\partial A} = \frac{|H(x,\theta)|}{A} \tag{24}$$

再对导数求导可以得到其 Hessian 矩阵如下：

(25)

$$\frac{\partial^2|H(x,\theta)|}{\partial a_j \partial a_k} = \frac{|H(x,\theta)|}{N_j^2 N_k^2}[a_j + (\ldots$$

(26)

$$\frac{\partial^2|H(x,\theta)|}{\partial a_j \partial b_j} = \frac{|H(x,\theta)|}{N_j^4}(a_j(\cos\theta \ldots$$

(27)

$$\frac{\partial^2|H(x,\theta)|}{\partial a_j \partial b_k} = \frac{|H(x,\theta)|}{N_j^2 N_k^2}[a_j + (1+b_j)\cos \ldots$$

(28)

$$\frac{\partial^2|H(x,\theta)|}{\partial a_j \partial c_k} = -\frac{|H(x,\theta)|}{N_j^2 D_k^2} \ldots$$

(29)

$$\frac{\partial^2|H(x,\theta)|}{\partial a_j \partial d_k} = -\frac{|H(x,\theta)|}{N_j^2 D_k^2}[a_j + (1 \ldots$$

(30)

$$\frac{\partial^2| \ldots}{\partial \ldots}$$

(31)

$$\frac{\partial^2| \ldots}{\ldots}$$

(32)

$$\frac{\partial^2|H(x,\theta)|}{\partial b_j \partial b_k} = \frac{|H(x,\theta)|}{N_j^2 N_k^2}(-1 + 2\cos^2\theta + a_j\cos\theta + b \ldots$$

(33)

$$\frac{\partial^2|H(x,\theta)|}{\partial b_j \partial c_k} = -\frac{|H(x,\theta)|}{N_j^2 D_k^2}[-1 + 2 \ldots$$

(34)

$$\frac{\partial^2|H(x,\theta)|}{\partial b_j \partial d_k} = -\frac{|H(x,\theta)|}{N_j^2 D_k^2}[-1 + 2\cos^2\theta + a_j \ldots$$

(35)

$$\frac{\partial^2|H(x,\theta)|}{\partial b_j \partial A} = \ldots$$

(36)

$$\frac{\partial^2|H(x,\theta)|}{\partial c_j^2} = \frac{|H(x,\theta)|}{D_j^4}(\ldots$$

(37)

$$\frac{\partial^2|H(x,\theta)|}{\partial c_j \partial d_j} = \frac{|H(x,\theta)|}{D_j^4}(4\cos\theta - 5c_j^2\cos\theta + 3c_j - 10c_j\cos^2\theta - 2d_j\cos\theta - 2d_j^2\cos\theta - 2d_j\cos^3\theta - 6\,\mathrm{co} \ldots$$

(38)

$$\frac{\partial^2|H(x,\theta)|}{\partial c_j \partial d_k} = \frac{|H(x,\theta)|}{D_j^2 D_k^2}[c_j + (1+d_j)\cos \ldots$$

在实验中逼近函数为 $S(\phi) = |1-2\phi|, 0 \le \phi \le 1$，误差函数为 $e(x,\phi) = |H(x,\theta) - S(\phi)|$，问题转化为 $\min_x \max_{0\le\Phi\le1} |e(x,\Phi)|$。$\Phi$ 的取值如下：

$$\Phi = 0.0, 0.05(0.01); \quad i = 1, \cdots, 6, \tag{44}$$

$$\Phi = 0.07, 0.46(0.03); \quad i = 7, \cdots, 20, \tag{45}$$

$$\Phi = 0.5, 0.54(0.04); \quad i = 21, 22 \tag{46}$$

$$\Phi = 0.57, 0.67(0.1); \quad i = 23, 24 \tag{47}$$

$$\Phi = 0.63, 0.93(0.03); \quad i = 25, \cdots, 35, \tag{48}$$

$$\Phi = 0.95, 1.0(0.01); \quad i = 36, \cdots, 41 \tag{49}$$

初始点设为 $(0,1,0,-0.15,0,-0.68,0,-0.72,0.37)$,最优解为 $(0., 0.980039, 0., -0.165771, 0., -0.767228$
各个方法的数值结果如表**??** 三种方法的误差曲线图如图 1,2,3。

表 3: 滤波器数值结果

方法	$\lvert f^* - f \rvert$	迭代次数	函数调用次数
Least pth	0.0064	7	21
Smooth Method	0.0086	26	105
Adaptive Smooth	0.0078	42	165

2.4 结果分析

在实验中，可以看出以下几点：

- Least pth 方法在几种情况是比较稳定的，而且在几组实验中的迭代次数和函数调用次数均是最少的，而且在滤波器的例子中达到了最优的效果。
- Smooth Method 方法实现比较简单，同时在三组实验中也达到了较不错的效果。
- Adaptive Smooth Method 方法虽然是在 Smooth Method 的基础上进行改进的，但是并不能一定比 Smooth Method 的效果好，但在多数情况下能够比 Smooth Method 在更少的迭代次数和函数调用次数下达到更好的效果。方法的稳定性不如 Least pth 方法。

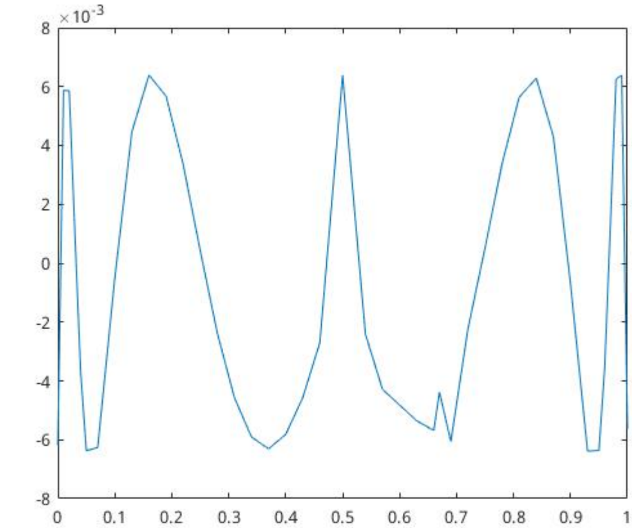

图 1: Leastpth 误差曲线

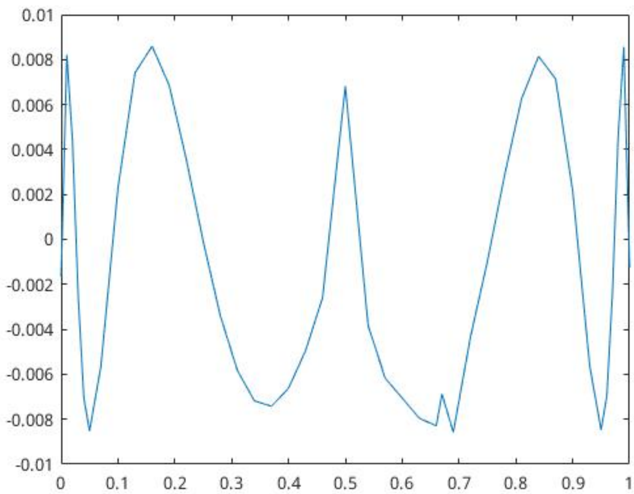

图 2: Smooth Method 误差曲线

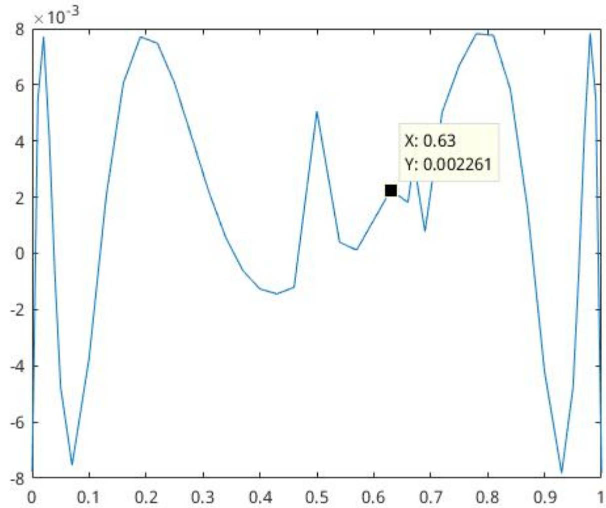

图 3: Adaptive Smooth 误差曲线