



# 最优化及其应用

## 期末大作业

姓名: ze

学号: 12345678

专业: 数学与应用数学

分配: 文稿、代码、作图

学院: 统计与数学学院

老师: 崔峰

2021 年 6 月 18 日

## 摘要

请在这里输入摘要内容.

# 目录

<b>1</b>	<b>常用命令模板</b>	<b>1</b>
1.1	公式 . . . . .	1
1.2	表格 . . . . .	2
1.3	图片 . . . . .	2
1.4	其他 . . . . .	3
<b>A</b>	<b>附录一：matlab 代码</b>	<b>5</b>
<b>B</b>	<b>附录二：python 代码</b>	<b>5</b>

# 1 常用命令模板

## 1.1 公式

$$\frac{\partial r_i(x)}{\partial x_j} = \begin{cases} -1 & j = i - 1 \\ 2 + 3h^2(x_i + t_i + 1)^2/2 & j = i \\ -1 & j = i + 1 \\ 0 & else \end{cases}, \quad 1 \leq i \leq m$$

$$\frac{\partial(r_1(x), \dots, r_n(x))}{\partial(x_1(x), \dots, x_n(x))} = \begin{bmatrix} \frac{\partial r_1(x)}{\partial x_1} & -1 & 0 & 0 & \dots & 0 & 0 \\ -1 & \frac{\partial r_2(x)}{\partial x_2} & -1 & 0 & \dots & 0 & 0 \\ 0 & -1 & \frac{\partial r_3(x)}{\partial x_3} & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \frac{\partial r_{m-1}(x)}{\partial x_{n-1}} & -1 \\ 0 & 0 & 0 & 0 & \dots & -1 & \frac{\partial r_m(x)}{\partial x_n} \end{bmatrix}$$

$$\nabla f(x)_j = 2((\frac{\partial r_j(x)}{\partial x_j})r_j(x) - r_{j-1}(x) - r_{j+1}(x)), \quad 1 \leq j \leq n, r_0(x) = r_{n+1}(x) = 0$$

$$\begin{aligned} \nabla f(x)_1 &= 2 \sum_{i=1}^m (\frac{\partial r_i(x)}{\partial x_j}) r_i(x) \\ &= 2 \sum_{i=1}^{29} (\frac{\partial r_i(x)}{\partial x_1}) r_i(x) + 2(\frac{\partial r_{30}(x)}{\partial x_1}) r_{30}(x) + 2(\frac{\partial r_{31}(x)}{\partial x_1}) r_{31}(x) \\ &= 2 \sum_{i=1}^{29} (\frac{\partial r_i(x)}{\partial x_1}) r_i(x) + 2x_1 - 4x_1(x_2 - x_1^2 - 1) \end{aligned}$$

1.2 表格

表 1: 使用 FR 法解决二次型问题

1	$k \leftarrow 0, g_1 \leftarrow \nabla f(x^{(1)}), d^{(1)} \leftarrow -g_1$
2	if $\ \nabla f(x^{(1)})\  \leq gtol$ :
3	return $x^{(1)}$
4	while $k \leq \text{MaxIter}$ :
5	$\lambda_k \leftarrow \frac{-g_k^T d^{(k)}}{d^{(k)T} A d^{(k)}}$
6	$x^{(k+1)} \leftarrow x^{(k)} + \lambda_k d^{(k)}$
7	if $\ \nabla f(x^{(k)})\  \leq gtol$ :
8	return $x^{(k)}$
9	$\beta_k \leftarrow \frac{\ g_{k+1}\ ^2}{\ g_k\ ^2}$
10	$d^{(k+1)} \leftarrow -g_{k+1} + \beta_k d^{(k)}$
11	$k \leftarrow k + 1$
12	return $x^{(k)}$

表格引用：表1.

1.3 图片

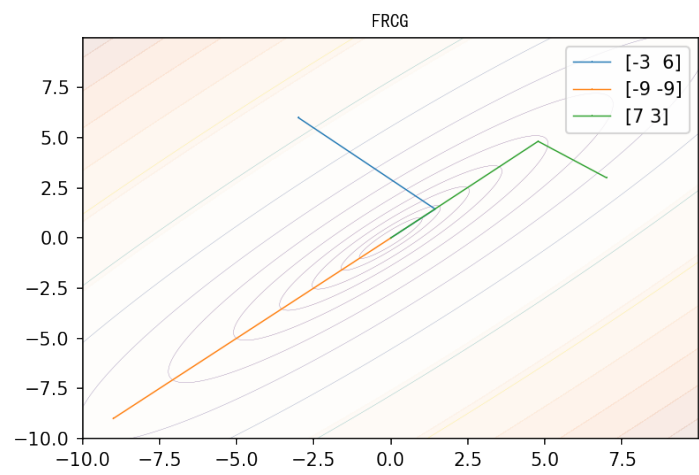


图 1: 用于正定二次函数的共轭梯度法

图片引用: 图1.

## 1.4 其他

章节引用：节1.4。

文献引用：[1] 或<sup>[1]</sup>。

使用 FR 法解决二次型问题的 Python 代码参见附录B.

## 参考文献

- [1] Jorge J. Moré, Burton S. Garbow, and Kenneth E. Hillstom. 1981. Testing Unconstrained Optimization Software. *ACM Trans. Math. Softw.* 7, 1 (March 1981), 17–41.

## A 附录一：matlab 代码

---

```
[X, Y] = meshgrid(0.01:0.01:1, 0.01:0.01:1);
Zfun = @(x,y)12.5*x.*log10(x).*y.*(y-1)+exp(-(25 ...
*x - 25/exp(1)).^2+(25*y-25/2).^2).^3)./25;
Z = Zfun(X,Y);
figure;
surf(Y,Z,X,'FaceColor',[1 0.75 0.65],'linestyle','none');
hold on
surf(Y+0.98,Z,X,'FaceColor',[1 0.75 0.65],'linestyle','none');
axis equal;
view([116 30]);
camlight;
lighting phong; % 设置光照和光照模式
```

---

## B 附录二：python 代码

使用 FR 法解决二次型问题的 Python 代码如下：

---

```
def FRCG(x,grad,A,gtol=1e-6,maxit=200):
    out = dict()
    out['x']=x
    g = grad(x)
    nrmg = np.linalg.norm(g, ord=2)
    if nrmg<=gtol:
        out['iter']=0
        return out
    d=-g
    for iter in range(1, maxit + 1):
        out['iter']=iter
        lam=-np.dot(g.T,d)/np.dot(d.T,np.dot(A,d))
        x=x+lam*d
        out['x']+=[x]
        gp=g
        g = grad(x)
        nrmg = np.linalg.norm(g, ord=2)
        if nrmg<=gtol:
            break
```



```

        beta=np.dot(g.T,g)/np.dot(gp.T,gp)
        d=-g+beta*d
    return out
def grad(x):
    return np.array([0.52*x[0]-0.48*x[1], -0.48*x[0]+0.52*x[1]])
A=np.array([[0.52,-0.48],[-0.48,0.52]])
out=FRCG(np.array([3,4]),grad,A)

```

---