

# 深蓝学院 VIO 第三次课程作业

温焕宇

2019.6.27

## 1 题一

### 1.1 请绘制样例代码中 LM 阻尼因子 $\mu$ 随着迭代变化的曲线图

答：使用工具：python，课程给代码'CurveFitting-LM'

步骤：

1. 修改并编译代码

在 problem.cc 文件里 while 循环下加入以下用于存储迭代次数和  $\lambda$ ：

```
1 ofstream save_lambda;  
2 save_lambda.open("../data/Lambda.txt", ios::app);  
3 save_lambda << iter << "\t" << currentLambda_ << endl;  
4 save_lambda.close();
```

编译并执行

```
1 mkdir build  
2 cd build/  
3 cmake ..  
4 make  
5 cd ..  
6 ./build/app/testCurveFitting
```

得到结果如下：

```
why@why-desktop:~/Desktop/深蓝VIO课程内容/作业3/CurveFitting_LM/build$ ./app/testCurveFitting  
Test CurveFitting start...  
iter: 0 , chi= 36048.3 , Lambda= 0.001  
iter: 1 , chi= 30015.5 , Lambda= 699.051  
iter: 2 , chi= 13421.2 , Lambda= 1864.14  
iter: 3 , chi= 7273.96 , Lambda= 1242.76  
iter: 4 , chi= 269.255 , Lambda= 414.252  
iter: 5 , chi= 105.473 , Lambda= 138.084  
iter: 6 , chi= 100.845 , Lambda= 46.028  
iter: 7 , chi= 95.9439 , Lambda= 15.3427  
iter: 8 , chi= 92.3017 , Lambda= 5.11423  
iter: 9 , chi= 91.442 , Lambda= 1.70474  
iter: 10 , chi= 91.3963 , Lambda= 0.568247  
iter: 11 , chi= 91.3959 , Lambda= 0.378832  
problem solve cost: 2.62049 ms  
makeHessian cost: 1.6515 ms  
-----After optimization, we got these parameters :  
0.941939 2.09453 0.965586  
-----ground truth:  
1.0, 2.0, 1.0
```

图 1: 没修改后的结果图

2. 写 python 画散点图:

```

1 import matplotlib.pyplot as plt
2 filename = 'Lamda.txt'
3 X, Y = [], []
4 for line in open(filename, 'r'):
5     value = [float(s) for s in line.split()]
6     X.append(value[0])
7     Y.append(value[1])
8
9 plt.plot(X, Y, 'ro')
10 plt.title('Lamda trends')
11 plt.xlabel('Iter')
12 plt.ylabel('Lamda')
13 plt.show()

```

结果图如下:

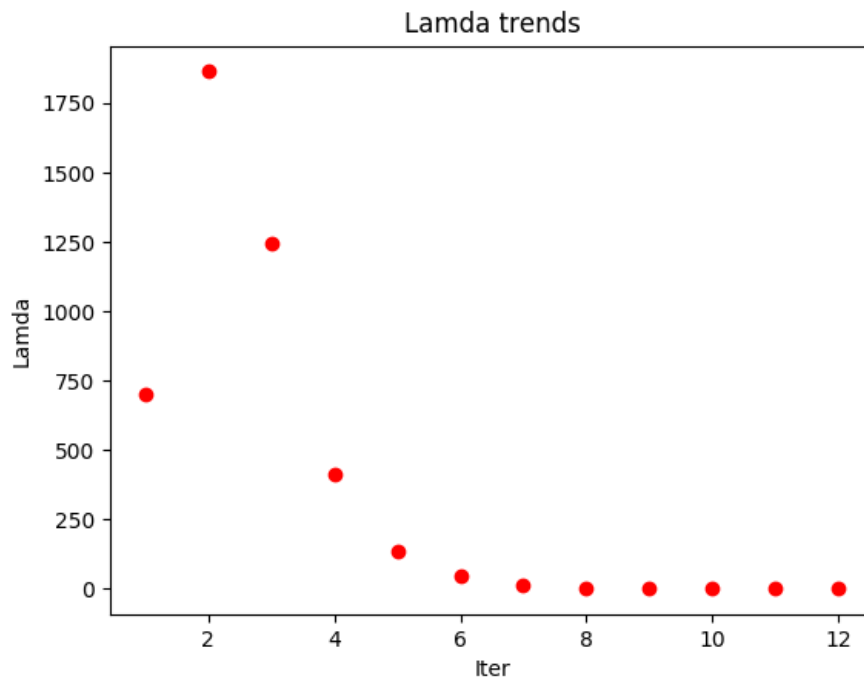


图 2:  $\mu$  随迭代变化的曲线图

1.2 将曲线函数改为  $y = ax^2 + bx + c$ , 请修改样例代码中残差计算, 雅可比计算等函数, 完成曲线估计

将  $y = \exp(ax^2 + bx + c)$  改成  $y = ax^2 + bx + c$

残差为:

$$residual = \check{a}x^2 + \check{b}x + \check{c} - y \quad (1)$$

其中 $\hat{y}$ 上标表示预测值

对  $a$   $b$   $c$  求偏导得雅可比矩阵为:

$$\begin{bmatrix} 2x & x & 1 \end{bmatrix} \quad (2)$$

修改代码如下:

```
1 residual_(0) = abc(0)*x_0*x_0 + abc(1)*x_0 + abc(2) - y_0; // 构建残差 = 预测值-测量值
2
3 jaco_abc << x_0 * x_0, x_0, 1; // y = a*x*x + b*x + c 的雅可比矩阵
```

【注：由于对于此方程，原始设置的噪声过大，不好收敛，故采用增大观测数或者减小噪声，考虑实际情况，选择增大观测次数  $N=1500$ 】

得到结果如下:

```
why@why-desktop:~/Desktop/深鉴VIO课程内容/作业3/CurveFitting_LM/cmake-build-debug/app$ ./testCurveFitting

Test CurveFitting start...
iter: 0 , chi= 2.09431e+07 , Lambda= 151.622
iter: 1 , chi= 1514.88 , Lambda= 50.5407
iter: 2 , chi= 1494.8 , Lambda= 16.8469
iter: 3 , chi= 1494.35 , Lambda= 5.61563
iter: 4 , chi= 1494.34 , Lambda= 3.74375
problem solve cost: 125.062 ms
makeHessian cost: 84.461 ms
-----After optimization, we got these parameters :
1.00074 1.99288 0.991964
-----ground truth:
1.0, 2.0, 1.0
```

图 3: 修改后输出结果

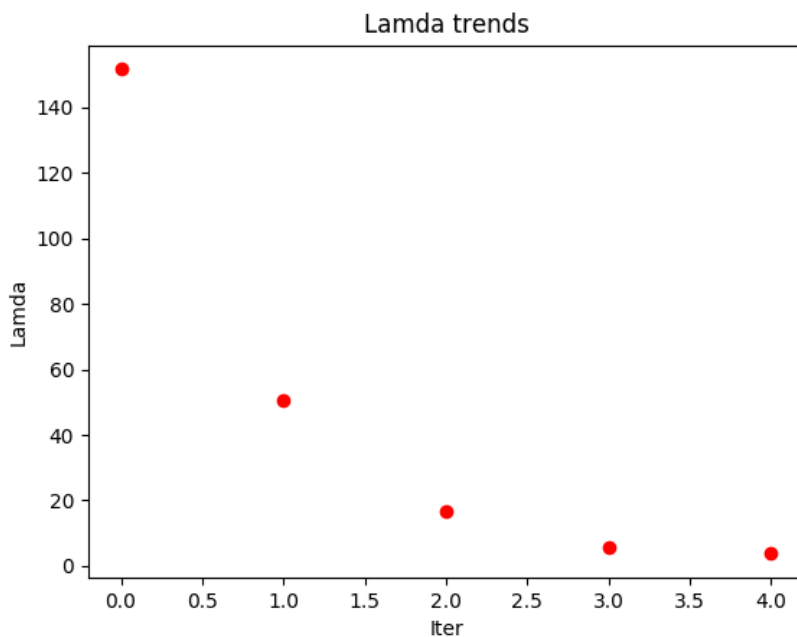


图 4: 修改后  $\lambda$  曲线图

### 1.3 实现其它阻尼因子更新策略

参考 ‘The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems’ 第一种阻尼因子更新策略：

1.  $\lambda_0 = \lambda_o$ ;  $\lambda_o$  is user-specified [8].  
 use eq'n (13) for  $\mathbf{h}_{lm}$  and eq'n (16) for  $\rho$   
 if  $\rho_i(\mathbf{h}) > \epsilon_4$ :  $\mathbf{p} \leftarrow \mathbf{p} + \mathbf{h}$ ;  $\lambda_{i+1} = \max[\lambda_i/L_{\downarrow}, 10^{-7}]$ ;  
 otherwise:  $\lambda_{i+1} = \min[\lambda_i L_{\uparrow}, 10^7]$ ;

其中  $L_{\downarrow}$  取 9,  $L_{\uparrow}$  取 11 时效果最好。

修改代码如下：

```

1  double maxDiagonal = 0;
2  ulong size = Hessian_.cols();
3  assert(Hessian_.rows() == Hessian_.cols() && "Hessian is not square");
4  for (ulong i = 0; i < size; ++i) {
5      maxDiagonal = std::max(fabs(Hessian_(i, i)), maxDiagonal); //Hessian矩阵对角元素最大值
6  }
7
8  scale = delta_x_.transpose() * (currentLambda_ * maxDiagonal * delta_x_ + b_);
9  scale += 1e-3;
10 double rho = (currentChi_ - tempChi) / scale;
11 if(rho > 0 && isfinite(tempChi))
12 {
13     double scaleFactor = (std::max)(currentLambda_/9, 10e-7);
14     currentLambda_ = scaleFactor;
15     currentChi_ = tempChi;
16     return true;
17 } else{
18     double scaleFactor = (std::min)(currentLambda_/11, 10e+7);
19     currentLambda_ = scaleFactor;
20     return false;
21 }
```

相比于第三种阻尼因子更新策略，第一种只需要迭代两次即可收敛，效果较好。结果如图 5：

```

why@why-desktop:~/Desktop/深蓝VIO课程内容/作业3/CurveFitting
Test CurveFitting start...
iter: 0 , chi= 2.09431e+07 , Lambda= 1e-05
iter: 1 , chi= 1494.34 , Lambda= 1.11111e-06
problem solve cost: 43.1952 ms
      makeHessian cost: 37.1302 ms
-----After optimization, we got these parameters :
      1.00075  1.99283  0.992141
-----ground truth:
      1.0,  2.0,  1.0
```

图 5: 采用第一种阻尼因子更新策略结果

## 2 题二

## 2.1 根据课程知识, 完成 F, G 推导

答: 根据课程 PPT 内容推导如下:

$$(1) \mathbf{f}_{15} = \frac{\partial \alpha_{b_i b_{k+1}}}{\partial \delta \mathbf{b}_k^g}$$

$$\mathbf{a} = \frac{1}{2} \left( \mathbf{q}_{b_i b_k} (a^{b_k} - b_k^a) + \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \boldsymbol{\omega} \delta t \end{bmatrix} (\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a) \right)$$

$$\boldsymbol{\omega} = \frac{1}{2} \left( (\boldsymbol{\omega}^{b_k} + \mathbf{n}_k^g - b_k^g) + (\boldsymbol{\omega}^{b_{k+1}} + \mathbf{n}_{k+1}^g - \mathbf{b}_k^g) \right) = \frac{1}{2} \left( (\boldsymbol{\omega}^{b_k} + \mathbf{n}_k^g) + (\boldsymbol{\omega}^{b_{k+1}} + \mathbf{n}_{k+1}^g) \right) - \mathbf{b}_k^g$$

$$\alpha_{b_i b_{k+1}} = \alpha_{b_i b_k} + \beta_{b_i b_k} + \frac{1}{2} \mathbf{a} \delta t^2$$

$\mathbf{b}_k^g$  与  $\boldsymbol{\omega}$  相关, 误差传递过程, 对  $\boldsymbol{\omega}$  右乘微小扰动  $\delta \mathbf{b}_k^g \delta t$  求以下偏导:

$$\mathbf{f}_{15} = \frac{\partial \alpha_{b_i b_{k+1}}}{\partial \delta \mathbf{b}_k^g} \quad (3)$$

$$= \frac{\partial \frac{1}{2} \cdot \frac{1}{2} \mathbf{a} \delta t^2}{\partial \delta \mathbf{b}_k^g} \quad (4)$$

$$= \frac{\partial \frac{1}{2} \cdot \frac{1}{2} \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \boldsymbol{\omega} \delta t \end{bmatrix} (a^{b_{k+1}} - b_k^a) \delta t^2}{\partial \delta \mathbf{b}_k^g} \quad (5)$$

$$= \frac{1}{4} \frac{\partial \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \boldsymbol{\omega} \delta t \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -\frac{1}{2} \delta \mathbf{b}_k^g \delta t \end{bmatrix} (\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a) \delta t^2}{\partial \delta \mathbf{b}_k^g} \quad (6)$$

$$= \frac{1}{4} \frac{\partial \mathbf{R}_{b_i b_{k+1}} \exp([- \frac{1}{2} \delta \mathbf{b}_k^g \delta t]_{\times}) (\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a) \delta t^2}{\partial \delta \mathbf{b}_k^g} \quad (7)$$

$$= \frac{1}{4} \frac{\partial \mathbf{R}_{b_i b_{k+1}} (\mathbf{I} + [- \frac{1}{2} \delta \mathbf{b}_k^g \delta t]_{\times}) (\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a) \delta t^2}{\partial \delta \mathbf{b}_k^g} \quad (8)$$

$$= -\frac{1}{4} \frac{\partial \mathbf{R}_{b_i b_{k+1}} [(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)]_{\times} \delta t^2 (-\delta \mathbf{b}_k^g \delta t)}{\partial \delta \mathbf{b}_k^g} \quad (9)$$

$$= -\frac{1}{4} \left( \mathbf{R}_{b_i b_{k+1}} [(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)]_{\times} \delta t^2 \right) (-\delta t) \quad (10)$$

$$(11)$$

$$(2) \mathbf{g}_{12} = \frac{\partial \alpha_{b_i b_{k+1}}}{\partial \delta \mathbf{n}_k^g}$$

$$\mathbf{a} = \frac{1}{2} \left( \mathbf{q}_{b_i b_k} (a^{b_k} - b_k^a) + \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \boldsymbol{\omega} \delta t \end{bmatrix} (\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a) \right)$$

$$\boldsymbol{\omega} = \frac{1}{2} \left( (\boldsymbol{\omega}^{b_k} + \mathbf{n}_k^g - b_k^g) + (\boldsymbol{\omega}^{b_{k+1}} + \mathbf{n}_{k+1}^g - b_{k+1}^g) \right) = \frac{1}{2} \left( (\boldsymbol{\omega}^{b_k} + \mathbf{n}_k^g) + (\boldsymbol{\omega}^{b_{k+1}} + \mathbf{n}_{k+1}^g) \right) - \mathbf{b}_k^g$$

$$\alpha_{b_i b_{k+1}} = \alpha_{b_i b_k} + \beta_{b_i b_k} + \frac{1}{2} \mathbf{a} \delta t^2$$

同理：对  $\boldsymbol{\omega}$  右乘微小扰动  $\frac{1}{2} \delta \mathbf{n}_k^g \delta t$  得：

$$\mathbf{g}_{12} = \frac{\partial \alpha_{b_i b_{k+1}}}{\partial \delta \mathbf{n}_k^g} \quad (12)$$

$$= \frac{\partial \frac{1}{2} \mathbf{a} \delta t^2}{\partial \delta \mathbf{n}_k^g} \quad (13)$$

$$= \frac{\partial \frac{1}{2} \cdot \frac{1}{2} \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \boldsymbol{\omega} \delta t \end{bmatrix} (a^{b_{k+1}} - b_k^a) \delta t^2}{\partial \delta \mathbf{n}_k^g} \quad (14)$$

$$= \frac{1}{4} \frac{\partial \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \boldsymbol{\omega} \delta t \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \cdot \frac{1}{2} \delta \mathbf{n}_k^g \delta t \end{bmatrix} (\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a) \delta t^2}{\partial \delta \mathbf{n}_k^g} \quad (15)$$

$$= \frac{1}{4} \frac{\partial \mathbf{R}_{b_i b_{k+1}} \exp([\frac{1}{2} \delta \mathbf{n}_k^g \delta t]_{\times}) (\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a) \delta t^2}{\partial \delta \mathbf{n}_k^g} \quad (16)$$

$$= \frac{1}{4} \frac{\partial \mathbf{R}_{b_i b_{k+1}} (\mathbf{I} + [\frac{1}{2} \delta \mathbf{n}_k^g \delta t]_{\times}) (\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a) \delta t^2}{\partial \delta \mathbf{n}_k^g} \quad (17)$$

$$= -\frac{1}{4} \frac{\partial \mathbf{R}_{b_i b_{k+1}} [\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a]_{\times} \delta t^2 (\frac{1}{2} \delta \mathbf{n}_k^g \delta t)}{\partial \delta \mathbf{n}_k^g} \quad (18)$$

$$= -\frac{1}{4} (\mathbf{R}_{b_i b_{k+1}} [(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)_{\times} \delta t^2] (\frac{1}{2} \delta t)) \quad (19)$$

$$(20)$$

### 3 题三

#### 3.1 证明式 (9)

$$\Delta \mathbf{x}_{lm} = - \sum_{j=1}^n \frac{\mathbf{v}_j^T \mathbf{F}'^T}{\lambda_j + \mu} \mathbf{v}_j \quad (21)$$

根据 GN-LM 论文推导:

$$(\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}) \Delta \mathbf{x}_{lm} = -\mathbf{J}^T \mathbf{f} = -\mathbf{F}'^T$$

由 SVD 分解

$$\mathbf{A} = \mathbf{J}^T \mathbf{J} = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^T \quad (22)$$

其中  $\mathbf{U} \in \mathbf{R}^{m \times m}$  和  $\mathbf{V} \in \mathbf{R}^{n \times n}$ 。

$\{\mathbf{u}_j\}_{j=1}^m$  和  $\{\mathbf{v}_j\}_{j=1}^n$  分别表示的  $\mathbf{U}, \mathbf{V}$  的列。

因为矩阵是正交的, 矢量形成两个正交基:

$$\mathbf{n}_i^T \mathbf{u}_j = \mathbf{v}_i^T \mathbf{v}_i = 1, i = j \quad (23)$$

由 (22) (23) 得:

$$\mathbf{A} = \sum_{j=1}^p (\lambda_j + \mu) \mathbf{u}_j \mathbf{v}_j^T, p \text{ 为矩阵 } \mathbf{A} \text{ 的秩} \quad (24)$$

$$-\mathbf{F}'^T = -\mathbf{J}^T \mathbf{f} = \sum_{j=1}^m \beta_j \mathbf{u}_j \quad (25)$$

$$\Delta \mathbf{x}_{lm} = \sum_{i=1}^n \eta_i \mathbf{v}_i \quad (26)$$

联立 (24-26) 得:

$$\mathbf{r} = -\mathbf{F}'^T - \mathbf{A} \Delta \mathbf{x}_{lm} \quad (27)$$

$$= \sum_{j=1}^p (\beta_j - (\lambda_j + \mu) \eta_j) \mathbf{u}_j + \sum_{j=p+1}^m \beta_j \mathbf{u}_j \quad (28)$$

对上式取模得最小值时满足以下:

$$\beta_j - (\lambda_j + \mu) \eta_j = 0 \quad (29)$$

$$\frac{\beta_j}{\lambda_j + \mu} = \eta_j, j = 1, \dots, p. \quad (30)$$

最小二乘结果表示为:

$$\Delta \mathbf{x}_{lm} = \sum_{j=1}^p \left( \frac{\beta_j}{\lambda_j + \mu} \right) \mathbf{v}_j + \sum_{j=p+1}^n \eta_j \mathbf{v}_j \quad (31)$$

$$\| \Delta \mathbf{x}_{lm} \|^2 = \sum_{j=1}^p \left( \frac{\beta_j}{\lambda_j + \mu} \right)^2 + \sum_{j=p+1}^n \eta_j^2 \quad (32)$$

当  $\Delta \mathbf{x}_{lm}$  取最小值时,  $\eta_{p+1} = \dots = \eta_n = 0$ , 由式 (25)(32), 得下式:

$$\Delta \mathbf{x}_{lmmin} = \sum_{j=1}^p \left( \frac{\beta_j}{\lambda_j + \mu} \right) \mathbf{v}_j = - \sum_{j=1}^p \frac{\mathbf{u}_j^T \mathbf{F}'^T}{\lambda_j + \mu} \mathbf{v}_j \quad (33)$$

根据矩阵运算推导:

i. 由于  $\mathbf{V}$  是正交阵, 具有性质:  $\mathbf{V}^T = \mathbf{V}^{-1} \Rightarrow (\mathbf{V}(\boldsymbol{\Lambda})\mathbf{V}^T)^{-1} = \mathbf{V}(\boldsymbol{\Lambda})^{-1}\mathbf{V}^T$

ii. 由课件推导可知:

$\mathbf{J}^T \mathbf{J}$  的特征值为  $\lambda_j$  和对应的特征向量为  $\mathbf{v}_j$

对  $\mathbf{J}^T \mathbf{J}$  特征分解后得:

$\mathbf{J}^T \mathbf{J} = \mathbf{V}(\boldsymbol{\Lambda} + \mu \mathbf{I})\mathbf{V}^T$  代入下式:

$$(\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}) \Delta \mathbf{x}_{lm} = -\mathbf{J}^T \mathbf{f} \quad (34)$$

$$\Delta \mathbf{x}_{lm} = (\mathbf{J}^T \mathbf{J} + \mu \mathbf{I})^{-1} (-\mathbf{J}^T \mathbf{f}) \quad (35)$$

$$= (\mathbf{J}^T \mathbf{J} + \mu \mathbf{I})^{-1} (-\mathbf{F}'^T) \quad (36)$$

$$= (\mathbf{V}(\boldsymbol{\Lambda} + \mu \mathbf{I})\mathbf{V}^T)^{-1} (-\mathbf{F}'^T) \quad (37)$$

$$= \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n \end{bmatrix} \begin{bmatrix} \frac{1}{\lambda_1 + \mu} & 0 & \dots & 0 \\ 0 & \frac{1}{\lambda_2 + \mu} & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & \frac{1}{\lambda_n + \mu} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \cdot \\ \cdot \\ \mathbf{v}_n^T \end{bmatrix} (-\mathbf{F}'^T) \quad (38)$$

$$= \begin{bmatrix} \frac{\mathbf{v}_1}{\lambda_1 + \mu} & \frac{\mathbf{v}_2}{\lambda_2 + \mu} & \dots & \frac{\mathbf{v}_n}{\lambda_n + \mu} \end{bmatrix} \begin{bmatrix} -\mathbf{v}_1^T \mathbf{F}'^T \\ -\mathbf{v}_2^T \mathbf{F}'^T \\ \cdot \\ \cdot \\ -\mathbf{v}_n^T \mathbf{F}'^T \end{bmatrix} \quad (39)$$

$$= - \sum_{j=1}^n \frac{\mathbf{v}_j \mathbf{v}_j^T \mathbf{F}'^T}{\lambda_j + \mu} \quad (40)$$