HW3

sgmliu9

July 2019

1 Q1

1.1 Q1.1

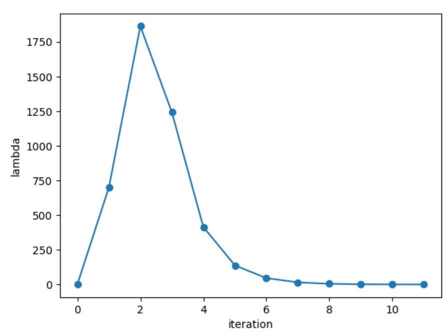

Figure 1: Lambda updates with successful iterations

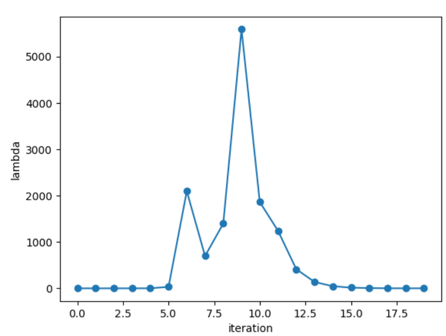

Figure 2: All the lambda updates with iterations

1.2 Q1.2

Assume

$$y' = x^2 + 2x + 1 \tag{1}$$

The residual is

$$r = x^2 + 2x + 1 - y \tag{2}$$

where y is the measurement data.
Then the Jacobian with respect to three coefficients is $[x^2, x, 1]$

```
Test CurveFitting start...
iter: 0 , chi= 7114.25 , Lambda= 0.01
iter: 1 , chi= 973.88 , Lambda= 0.00333333
iter: 2 , chi= 973.88 , Lambda= 0.00222222
problem solve cost: 1.17062 ms
   makeHessian cost: 0.916768 ms
-------After optimization, we got these parameters :
0.958923  2.06283 0.968821
-------ground truth:
1.0,  2.0,  1.0
```

Figure 3: Result

1.3 Q1.3

λ is initialized using

$$\lambda_0 = \tau * max\{(\mathbf{J}^T\mathbf{J})_{ii}\} \tag{3}$$

where τ is a user-specified constant.

The parameter λ is updated using

2. $\lambda_0 = \lambda_o \max\left[\mathsf{diag}[\boldsymbol{J}^\mathsf{T}\boldsymbol{W}\boldsymbol{J}]\right]$; λ_o is user-specified.
   use eq'n (12) for $\boldsymbol{h}_{\mathsf{lm}}$ and eq'n (15) for ρ
   $\alpha = \left(\left(\boldsymbol{J}^\mathsf{T}\boldsymbol{W}(\boldsymbol{y}-\hat{\boldsymbol{y}}(\boldsymbol{p}))\right)^\mathsf{T}\boldsymbol{h}\right) / \left((\chi^2(\boldsymbol{p}+\boldsymbol{h})-\chi^2(\boldsymbol{p}))/2 + 2\left(\boldsymbol{J}^\mathsf{T}\boldsymbol{W}(\boldsymbol{y}-\hat{\boldsymbol{y}}(\boldsymbol{p}))\right)^\mathsf{T}\boldsymbol{h}\right)$;
   if $\rho_i(\alpha\boldsymbol{h}) > \epsilon_4$: $\boldsymbol{p} \leftarrow p + \alpha\boldsymbol{h}$; $\lambda_{i+1} = \max[\lambda_i/(1+\alpha), 10^{-7}]$;
   otherwise: $\lambda_{i+1} = \lambda_i + |\chi^2(\boldsymbol{p}+\alpha\boldsymbol{h}) - \chi^2(\boldsymbol{p})|/(2\alpha)$;

Figure 4: LM update method

where $\mathbf{b} = -J^TW * (y' - y)^T = J^TW * (y - y')^T$ and ρ is calculated using

$$\begin{aligned}
\rho_i(\boldsymbol{h}_{\mathsf{lm}}) &= \frac{\chi^2(\boldsymbol{p}) - \chi^2(\boldsymbol{p}+\boldsymbol{h}_{\mathsf{lm}})}{(\boldsymbol{y}-\hat{\boldsymbol{y}})^\mathsf{T}\boldsymbol{W}(\boldsymbol{y}-\hat{\boldsymbol{y}}) - (\boldsymbol{y}-\hat{\boldsymbol{y}}-\boldsymbol{J}\boldsymbol{h}_{\mathsf{lm}})^\mathsf{T}\boldsymbol{W}(\boldsymbol{y}-\hat{\boldsymbol{y}}-\boldsymbol{J}\boldsymbol{h}_{\mathsf{lm}})} && (14)\\
&= \frac{\chi^2(\boldsymbol{p}) - \chi^2(\boldsymbol{p}+\boldsymbol{h}_{\mathsf{lm}})}{\boldsymbol{h}_{\mathsf{lm}}^\mathsf{T}\left(\lambda_i\boldsymbol{h}_{\mathsf{lm}} + \boldsymbol{J}^\mathsf{T}\boldsymbol{W}\left(\boldsymbol{y}-\hat{\boldsymbol{y}}(\boldsymbol{p})\right)\right)} \quad \text{if using eq'n (12) for } \boldsymbol{h}_{\mathsf{lm}} && (15)\\
&= \frac{\chi^2(\boldsymbol{p}) - \chi^2(\boldsymbol{p}+\boldsymbol{h}_{\mathsf{lm}})}{\boldsymbol{h}_{\mathsf{lm}}^\mathsf{T}\left(\lambda_i\mathsf{diag}(\boldsymbol{J}^\mathsf{T}\boldsymbol{W}J)\boldsymbol{h}_{\mathsf{lm}} + \boldsymbol{J}^\mathsf{T}\boldsymbol{W}\left(\boldsymbol{y}-\hat{\boldsymbol{y}}(\boldsymbol{p})\right)\right)} \quad \text{if using eq'n (13) for } \boldsymbol{h}_{\mathsf{lm}} && (16)
\end{aligned}$$

The result for updating λ using this method.

```
Test CurveFitting start...
iter: 0 , chi= 2.19457e+07 , Lambda= 117.062
iter: 1 , chi= 713.9 , Lambda= 81.0421
iter: 2 , chi= 698.396 , Lambda= 54.1523
iter: 3 , chi= 697.31 , Lambda= 34.2503
iter: 4 , chi= 696.942 , Lambda= 20.1081
iter: 5 , chi= 696.75 , Lambda= 12.1351
iter: 6 , chi= 696.686 , Lambda= 7.60512
iter: 7 , chi= 696.674 , Lambda= 4.90785
iter: 8 , chi= 696.673 , Lambda= 3.23527
iter: 9 , chi= 696.673 , Lambda= 2.16495
problem solve cost: 3.18822 ms
   makeHessian cost: 2.41581 ms
-------After optimization, we got these parameters :
0.998222  2.02054 0.936674  1.01939
-------ground truth:
1.0,  2.0,  1.0 , 1.0
```

Figure 5: Result

The result for updating λ using the given code.

```
Test CurveFitting start...
iter: 0 , chi= 2.19457e+07 , Lambda= 117.062
iter: 1 , chi= 713.9 , Lambda= 39.0205
iter: 2 , chi= 697.63 , Lambda= 13.0068
iter: 3 , chi= 696.811 , Lambda= 4.33561
iter: 4 , chi= 696.678 , Lambda= 1.4452
iter: 5 , chi= 696.673 , Lambda= 0.96347
iter: 6 , chi= 696.673 , Lambda= 0.642313
problem solve cost: 2.52152 ms
   makeHessian cost: 2.09793 ms
-------After optimization, we got these parameters :
0.998218  2.02059 0.936515  1.01952
-------ground truth:
1.0,  2.0,  1.0 , 1.0
```

Figure 6: Result

The result for updating λ using the method below

3. $\lambda_0 = \lambda_o \max\left[\mathsf{diag}[\boldsymbol{J}^\mathsf{T}\boldsymbol{W}\boldsymbol{J}]\right]$; λ_o is user-specified [9].
use eq'n (12) for $\boldsymbol{h}_{\mathsf{lm}}$ and eq'n (15) for ρ
if $\rho_i(\boldsymbol{h}) > \epsilon_4$: $\boldsymbol{p} \leftarrow \boldsymbol{p} + \boldsymbol{h}$; $\lambda_{i+1} = \lambda_i \max\left[1/3, 1-(2\rho_i - 1)^3\right]; \nu_i = 2$;
otherwise: $\lambda_{i+1} = \lambda_i \nu_i; \quad \nu_{i+1} = 2\nu_i$;

Figure 7: Method for updating lambda

```
Test CurveFitting start...
iter: 0 , chi= 2.19457e+07 , Lambda= 117.062
iter: 1 , chi= 713.9 , Lambda= 39.0205
iter: 2 , chi= 697.63 , Lambda= 13.0068
iter: 3 , chi= 696.811 , Lambda= 4.33561
iter: 4 , chi= 696.678 , Lambda= 1.4452
iter: 5 , chi= 696.673 , Lambda= 0.980577
iter: 6 , chi= 696.673 , Lambda= 1.79457
problem solve cost: 2.09265 ms
   makeHessian cost: 1.74317 ms
-------After optimization, we got these parameters :
0.998218  2.02059 0.936515  1.01952
-------ground truth:
1.0,  2.0,  1.0 , 1.0
```

Figure 8: Method for updating lambda

The original method limits the decrease of lambda, compared with the third method. The original method achieves this using

```
alpha = std::min(alpha, 2. / 3.);//Note: 这里加上2/3 是让他别下降太快
```

2 Q2

2.1 Q2.1

Prove:

$$\mathbf{f}_{15} = \frac{\partial \boldsymbol{\alpha}_{\boldsymbol{b_i b_{k+1}}}}{\partial \delta \mathbf{b}_k^g} = -\frac{1}{4}\left(\mathbf{R}_{b_i b_{k+1}}\left[\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right]_\times \delta t^2\right)(-\delta t) \tag{4}$$

As we know,

$$\boldsymbol{\alpha}_{\boldsymbol{b_i b_{k+1}}} = \boldsymbol{\alpha}_{b_i b_k} + \boldsymbol{\beta}_{b_i b_k}\delta t + \frac{1}{2}\mathbf{a}\delta t^2 \tag{5}$$

where,

$$\mathbf{a} = \frac{1}{2}\left(\mathbf{q}_{b_i b_k}\left(\mathbf{a}^{b_k} + \mathbf{n}_k^a - \mathbf{b}_k^a\right) + \mathbf{q}_{b_i b_{k+1}}\left(\mathbf{a}^{b_{k+1}} + \mathbf{n}_{k+1}^a - \mathbf{b}_k^a\right)\right) \tag{6}$$

Furthermore, the bias $\partial \delta \mathbf{b}_k^g$ of the angular velocity at time k is

$$\boldsymbol{\omega} = \frac{1}{2}\left(\left(\boldsymbol{\omega}^{b_k} - \mathbf{b}_k^g\right) + \left(\boldsymbol{\omega}^{b_{k+1}} - \mathbf{b}_k^g\right)\right) = \frac{1}{2}\left(\boldsymbol{\omega}^{b_k} + \boldsymbol{\omega}^{b_{k+1}}\right) - \mathbf{b}_k^g \tag{7}$$

As a result,

$$
\begin{aligned}
\frac{\partial \boldsymbol{\alpha}_{\boldsymbol{b_i b_{k+1}}}}{\partial \delta \mathbf{b}_k^g} =& \frac{1}{4} \frac{\partial \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\omega \delta t \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -\frac{1}{2}\delta \mathbf{b}_k^g \delta t \end{bmatrix} \left(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right) \delta t^2}{\partial \delta \mathbf{b}_k^g} \\
=& \frac{1}{4} \frac{\partial \mathbf{R}_{b_i b_{k+1}} \exp\left(\left[-\delta \mathbf{b}_k^g \delta t\right]_\times\right) \left(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right) \delta t^2}{\partial \delta \mathbf{b}_k^g} \\
=& \frac{1}{4} \frac{\partial \mathbf{R}_{b_i b_{k+1}} \left(\mathbf{I} + \left[-\delta \mathbf{b}_k^g \delta t\right]_\times\right) \left(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right) \delta t^2}{\partial \delta \mathbf{b}_k^g} \\
=& -\frac{1}{4} \frac{\partial \mathbf{R}_{b_i b_{k+1}} \left(\left[\left(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right) \delta t^2\right]_\times\right) \left(-\delta \mathbf{b}_k^g \delta t\right)}{\partial \delta \mathbf{b}_k^g} \\
=& -\frac{1}{4} \left(\mathbf{R}_{b_i b_{k+1}} \left[\left(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right)\right]_\times \delta t^2\right) (-\delta t)
\end{aligned} \tag{8}
$$

2.2 Q2.2

Prove:

$$
\mathbf{g_{12}} = \frac{\partial \alpha_{b_i b_{k+1}}}{\partial \mathbf{n}_k^g} = -\frac{1}{4} \left(\mathbf{R}_{b_i b_{k+1}} \left[\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right]_\times \delta t^2\right) \left(\frac{1}{2}\delta t\right) \tag{9}
$$

Similarly,

$$
\begin{aligned}
\frac{\partial \boldsymbol{\alpha}_{\boldsymbol{b_i b_{k+1}}}}{\partial \mathbf{n}_k^g} =& \frac{1}{4} \frac{\partial \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\omega \delta t \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\frac{1}{2}\mathbf{n}_k^g \delta t \end{bmatrix} \left(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right) \delta t^2}{\partial \mathbf{n}_k^g} \\
=& \frac{1}{4} \frac{\partial \mathbf{R}_{b_i b_{k+1}} \exp\left(\left[\frac{1}{2}\mathbf{n}_k^g \delta t\right]_\times\right) \left(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right) \delta t^2}{\partial \mathbf{n}_k^g} \\
=& \frac{1}{4} \frac{\partial \mathbf{R}_{b_i b_{k+1}} \left(\mathbf{I} + \frac{1}{2}\left[\mathbf{n}_k^g \delta t\right]_\times\right) \left(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right) \delta t^2}{\partial \mathbf{n}_k^g} \\
=& -\frac{1}{4} \frac{\partial \mathbf{R}_{b_i b_{k+1}} \left(\left[\left(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right) \delta t^2\right]_\times\right) \frac{1}{2}\left(\mathbf{n}_k^g \delta t\right)}{\partial \mathbf{n}_k^g} \\
=& -\frac{1}{4} \left(\mathbf{R}_{b_i b_{k+1}} \left[\left(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right)\right]_\times \delta t^2\right) \left(\frac{1}{2}\delta t\right)
\end{aligned} \tag{10}
$$

3 Q3

Prove:

$$
\Delta \mathbf{x}_{lm} = -\sum_{j=1}^{n} \frac{\mathbf{v}_j^{\mathrm{T}} F'^{\mathrm{T}}}{\lambda_j + \mu} \mathbf{v}_j \tag{11}
$$

As we know,

$$\left(\mathbf{J}^{\top}\mathbf{J}+\mu\mathbf{I}\right)\Delta\mathbf{x}_{\mathrm{lm}}=-\mathbf{J}^{\top}\mathbf{f}=-\mathbf{F}^{'}(\mathbf{x})^{T} \tag{12}$$

The SVD of $\mathbf{J^TJ}$ is

$$\mathbf{J^TJ}=\mathbf{V\Lambda V}^{\top} \tag{13}$$

$$\mathbf{VV}^{\top}=\mathbf{I} \tag{14}$$

Then we have,

$$\begin{aligned}
(\mathbf{V\Lambda V}^{\top}+\mu\mathbf{VV}^{\top})\Delta\mathbf{x}_{\mathrm{lm}} &= -\mathbf{F}^{'}(\mathbf{x})^{T} \\
\mathbf{V}(\mathbf{\Lambda}+\mu\mathbf{I})\mathbf{V}^{\top}\Delta\mathbf{x}_{\mathrm{lm}} &= -\mathbf{F}^{'}(\mathbf{x})^{T} \\
(\mathbf{\Lambda}+\mu\mathbf{I})\Delta\mathbf{x}_{\mathrm{lm}} &= -\mathbf{V}^{\top}\mathbf{F}^{'}(\mathbf{x})^{T}\mathbf{V} \\
\Delta\mathbf{x}_{\mathrm{lm}} &= -\sum_{j=1}^{n}\frac{\mathbf{v}_j^{\mathrm{T}}F'^{\mathrm{T}}}{\lambda_j+\mu}\mathbf{v}_j
\end{aligned} \tag{15}$$