Final Homework

苑闻 2200010833

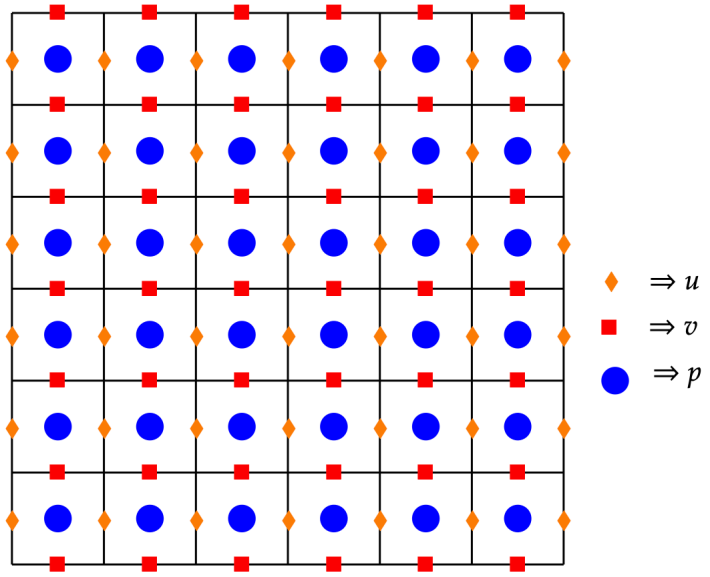
求解Stokes方程:

$$\begin{cases} -\Delta\vec{u} + \nabla p = \vec{F}, & (x,y) \in (0,1)\times(0,1) \\ \text{div}\vec{u} = 0, & (x,y) \in (0,1)\times(0,1) \end{cases} \tag{1}$$

边界条件

$$\begin{aligned} \frac{\partial u}{\partial \vec{n}} &= b, y = 0, & \frac{\partial u}{\partial \vec{n}} &= t, y = 1 \\ \frac{\partial v}{\partial \vec{n}} &= l, x = 0, & \frac{\partial v}{\partial \vec{n}} &= r, x = 1 \\ u &= 0, x = 0, 1, & v &= 0, y = 0, 1 \end{aligned}$$

其中$\vec{u} = (u, v)$为速度, p为压力, $\vec{F} = (f, g)$为外力, $\vec{n}$为外法向方向



stokes方程数值形式为($h=\frac{1}{N}$):

$$
\begin{aligned}
-\frac{u_{i+1,j-\frac{1}{2}}-2u_{i,j-\frac{1}{2}}+u_{i-1,j-\frac{1}{2}}}{h^2}-\frac{u_{i,j+\frac{1}{2}}-2u_{i,j-\frac{1}{2}}+u_{i,j-\frac{3}{2}}}{h^2}+\frac{p_{i+\frac{1}{2},j-\frac{1}{2}}-p_{i-\frac{1}{2},j-\frac{1}{2}}}{h}&=f_{i,j-\frac{1}{2}}\\
-\frac{v_{i-\frac{1}{2},j+1}-2v_{i-\frac{1}{2},j}+v_{i-\frac{1}{2},j-1}}{h^2}-\frac{v_{i+\frac{1}{2},j}-2v_{i-\frac{1}{2},j}+v_{i-\frac{3}{2},j}}{h^2}+\frac{p_{i-\frac{1}{2},j+\frac{1}{2}}-p_{i-\frac{1}{2},j-\frac{1}{2}}}{h}&=g_{i-\frac{1}{2},j}\\
\frac{u_{i,j-\frac{1}{2}}-u_{i-1,j-\frac{1}{2}}}{h}+\frac{v_{i-\frac{1}{2},j}-v_{i-\frac{1}{2},j-1}}{h}&=0
\end{aligned}
$$

边界条件的数值形式为:

$$
\begin{aligned}
-\frac{u_{i+1,\frac{1}{2}}-2u_{i,\frac{1}{2}}+u_{i-1,\frac{1}{2}}}{h^2}-\frac{u_{i,\frac{3}{2}}-u_{i,\frac{1}{2}}}{h^2}+\frac{p_{i+\frac{1}{2},\frac{1}{2}}-p_{i-\frac{1}{2},\frac{1}{2}}}{h}&=f_{i,\frac{1}{2}}+\frac{b_{i,0}}{h}\\
-\frac{u_{i+1,N-\frac{1}{2}}-2u_{i,N-\frac{1}{2}}+u_{i-1,N-\frac{1}{2}}}{h^2}+\frac{u_{i,N-\frac{1}{2}}-u_{i,N-\frac{3}{2}}}{h^2}+\frac{p_{i+\frac{1}{2},N-\frac{1}{2}}-p_{i-\frac{1}{2},N-\frac{1}{2}}}{h}&=f_{i,N-\frac{1}{2}}+\frac{t_{i,N}}{h}\\
-\frac{v_{\frac{1}{2},j+1}-2v_{\frac{1}{2},j}+v_{\frac{1}{2},j-1}}{h^2}-\frac{v_{\frac{3}{2},j}-v_{\frac{1}{2},j}}{h^2}+\frac{p_{\frac{1}{2},j+\frac{1}{2}}-p_{\frac{1}{2},j-\frac{1}{2}}}{h}&=g_{\frac{1}{2},j}+\frac{l_{0,j}}{h}\\
-\frac{v_{N-\frac{1}{2},j+1}-2v_{N-\frac{1}{2},j}+v_{N-\frac{1}{2},j-1}}{h^2}+\frac{v_{N-\frac{1}{2},j}-v_{N-\frac{3}{2},j}}{h^2}+\frac{p_{N-\frac{1}{2},j+\frac{1}{2}}-p_{N-\frac{1}{2},j-\frac{1}{2}}}{h}&=g_{N-\frac{1}{2},j}+\frac{r_{N,j}}{h}\\
u_{0,j-\frac{1}{2}}=u_{N,j-\frac{1}{2}}=v_{i-\frac{1}{2},0}=v_{i-\frac{1}{2},N}&=0
\end{aligned}
$$

将第k次迭代的$u_{i,j-\frac{1}{2}}$记为$u^k_{i,j}$, $v_{i-\frac{1}{2},j}$记为$v^k_{i,j}$, $p_{i-\frac{1}{2},j-\frac{1}{2}}$记为$p^k_{i,j}$
则我们需要解线性方程:

$$
\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix}\begin{bmatrix} U \\ P \end{bmatrix}=\begin{bmatrix} F \\ 0 \end{bmatrix}
$$

1 DGS+Vcycle

1.1 Algorithm

1.1.1 单次DGS迭代

Gauss迭代更新速度分量

stokes方程的Gauss迭代可以写成:

$$
\begin{aligned}
-\frac{u^k_{i+1,j}-2u^{k+1}_{i,j}+u^{k+1}_{i-1,j}}{h^2}-\frac{u^k_{i,j+1}-2u^{k+1}_{i,j}+u^{k+1}_{i,j-1}}{h^2}+\frac{p^k_{i+1,j}-p^k_{i,j}}{h}&=f_{i,j-\frac{1}{2}}\\
-\frac{v^k_{i,j+1}-2v^{k+1}_{i,j}+v^{k+1}_{i,j-1}}{h^2}-\frac{v^k_{i+1,j}-2v^{k+1}_{i,j}+v^{k+1}_{i-1,j}}{h^2}+\frac{p^k_{i,j+1}-p^k_{i,j}}{h}&=g_{i-\frac{1}{2},j}
\end{aligned}
$$

边界条件的Gauss迭代可以写成:

$$
\begin{aligned}
-\frac{u_{i+1,1}^{k}-2u_{i,1}^{k+1}+u_{i-1,1}^{k+1}}{h^2}-\frac{u_{i,2}^{k}-u_{i,1}^{k}}{h^2}+\frac{p_{i+1,1}^{k}-p_{i,1}^{k}}{h}&=f_{i,\frac{1}{2}}+\frac{b_{i,0}}{h}\\
-\frac{u_{i+1,N}^{k}-2u_{i,N}^{k+1}+u_{i-1,N}^{k+1}}{h^2}+\frac{u_{i,N}^{k+1}-u_{i,N-1}^{k+1}}{h^2}+\frac{p_{i+1,N}^{k}-p_{i,N}^{k}}{h}&=f_{i,N-\frac{1}{2}}+\frac{t_{i,N}}{h}\\
-\frac{v_{1,j+1}^{k}-2v_{1,j}^{k+1}+v_{1,j-1}^{k+1}}{h^2}-\frac{v_{2,j}^{k}-v_{1,j}^{k+1}}{h^2}+\frac{p_{1,j+1}^{k}-p_{1,j}^{k}}{h}&=g_{\frac{1}{2},j}+\frac{l_{0,j}}{h}\\
-\frac{v_{N,j+1}^{k}-2v_{N,j}^{k+1}+v_{N,j-1}^{k+1}}{h^2}+\frac{v_{N,j}^{k+1}-v_{N-1,j}^{k+1}}{h^2}+\frac{p_{N,j+1}^{k}-p_{N,j}^{k}}{h}&=g_{N-\frac{1}{2},j}+\frac{r_{N,j}}{h}\\
u_{0,j}^{k+1}=u_{N,j}^{k+1}=v_{i,0}^{k+1}=v_{i,N}^{k+1}&=0
\end{aligned}
$$

对每个内部单元(i,j)计算散度残量

$$r_{i,j}=-\frac{u_{i,j}^{k+1}-u_{i-1,j}^{k+1}}{h}-\frac{v_{i,j}^{k+1}-v_{i,j-1}^{k+1}}{h}$$

令$delta=r_{i,j}h/4$

更新内部单元速度

$$
\begin{aligned}
u_{i-1,j}^{k+2}=u_{i-1,j}^{k+1}-\delta, u_{i,j}^{k+2}=u_{i,j}^{k+1}+\delta\\
v_{i,j-1}^{k+2}=v_{i,j-1}^{k+1}-\delta, v_{i,j}^{k+2}=v_{i,j}^{k+1}+\delta
\end{aligned}
$$

更新内部单元压力

$$
\begin{aligned}
p_{i,j}^{k+2}&=p_{i,j}^{k}+r_{i,j},\\
p_{i+1,j}^{k+2}=p_{i+1,j}^{k}-r_{i,j}/4, p_{i-1,j}^{k+2}&=p_{i-1,j}^{k}-r_{i,j}/4,\\
p_{i,j+1}^{k+2}=p_{i,j+1}^{k}-r_{i,j}/4, p_{i,j-1}^{k+2}&=p_{i,j-1}^{k}-r_{i,j}/4,
\end{aligned}
$$

更新边界单元和顶点单元

与前面类似, 只是将分子的4换成3和2

1.1.2 Vcycle

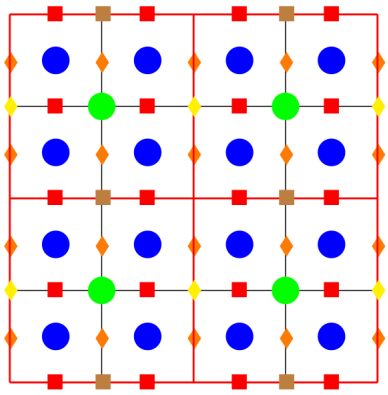
Vcycle算法框架如下 (求解$Ax=b$):
1) 每一轮开始时, 如果是最上层循环, 以上一轮得到的x_0作为初始值, 否则用0作为初始值
2) 先做ν_1次DGS得到x_0
3) 再计算当今误差$b_0=b-Ax_0$, 对其做一次限制算子降低维数得到b_1
4) 重复Vcycle, 近似求解$Ax_1=b_1$
5) 对求解到的x_1做一次提升算子得到x_2

6) 更新$x_0 < -x_0 + x_2$ 7) 再做ν_2次DGS得到x_0
8) 回溯
以下为课堂上的限制和提升算子:

限制算子：

速度：♦=0.25*最近两个♦+0.125*周围四个♦

压力：● =0.25*最近的四个●

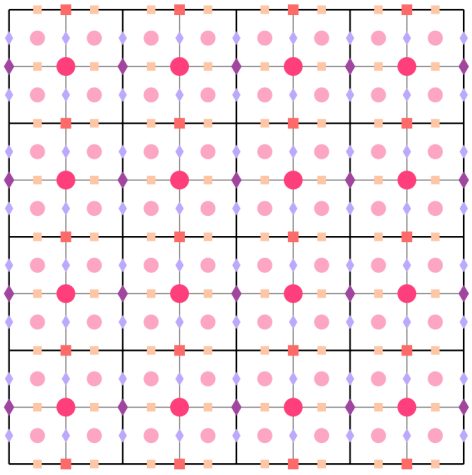


提升算子：

内部粗网格线上：♦=0.25*远♦+0.75*近♦　上、下边界粗网格线上：♦=0.5*最近♦

细网格线上：♦=0.5*最近♦+0.5*最近♦　压力：● =最近●

以下为作业pdf中新加上的限制和提升算子:



限制算子

♦ = 最近两个♦/4 + 次近四个♦/8

■ = 最近两个■/4 + 次近四个■/8

● = 最近四个●/4

提升算子

♦ (粗网格线) = 最近♦

♦ (细网格线) = 最近两个♦/2

■ (粗网格线) = 最近■

■ (细网格线) = 最近两个■/2

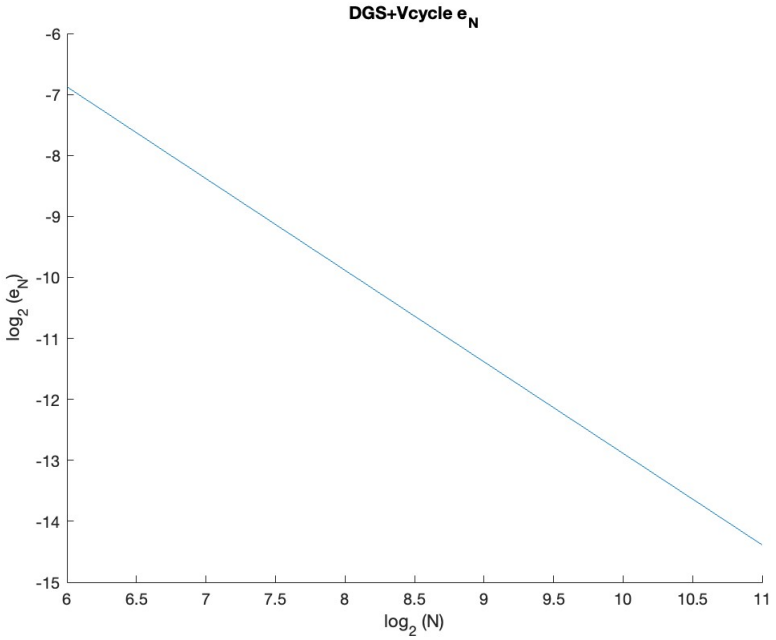
● = 最近●

1.2 Implementation

为了加快速度,采用作业pdf上的限制提升算子,如果想测试课上讲的限制提升算子,只需将get_para()函数中的type改成class即可
具体函数见code里的README.md

1.3 Result

采用参数$\nu_1 = 10, \nu_2 = 5, L = 2$



N	time	e_N	steps
64	0.14275	0.0085233	6
128	0.12871	0.0030007	6
256	0.57573	0.0010587	7
512	2.4236	0.00037395	7
1024	18.5366	0.00013215	7
2048	94.1432	4.6709e-05	7

由图易见$e_N = O(\frac{1}{N^2})$(斜率为-2)
以下给出了一些其他参数的情况的表格以及解释:
若$\nu_1 = 20, \nu_2 = 5, L = 2$

N	time	e_N	steps
64	0.1059	0.0085232	5
128	0.1971	0.0030006	5
256	0.9526	0.0010587	6
512	3.9393	0.00037395	6
1024	26.6513	0.00013214	6
2048	133.3457	4.6707e-05	6

可以看出ν_1的选取若太大, 降低速度, 但误差基本不改变
若$\nu_1 = 10, \nu_2 = 10, L = 2$

N	time	e_N	steps
64	0.047773	0.0085233	6
128	0.19007	0.0030007	6
256	0.73949	0.0010587	6
512	3.329	0.00037395	6
1024	20.7469	0.00013214	6
2048	102.7992	4.6701e-05	6

可以看出ν_2的选取若不算太大, 对速度的影响就较小, 当然误差基本无影响
若$\nu_1 = 10, \nu_2 = 5, L = 3$

N	time	e_N	steps
64	0.11506	0.0085233	18
128	0.46177	0.0030007	18
256	1.8105	0.0010587	17
512	7.7597	0.00037394	17
1024	49.5902	0.00013213	17
2048	245.2495	4.6683e-05	17

可以看出L必须取很小才能保证迭代收敛的速度
咨询助教后, 助教说L可以取很大, 猜测是我写的方程的问题, 我直接采用stokes数值形式写数值方程,
但还可以消除分母, 也就是将stokes压力梯度的方程乘h^2, 速度散度的方程乘h, 我的做法可能对方程产生影响,
导致DGS迭代收敛速度太慢, 从而需要更小的L保证收敛速度

2 Uzawa Iteration Method + CG

2.1 Algorithm

算法框架如下:
1) 每一轮开始时, 用共轭梯度法求解$AU_{k+1} = F - BP_k$
2) 然后对压力进行更新$P_{k+1} = P_k + \alpha(B^T U_{k+1})$
3) 判断误差是否小于允许的值, 从而判断是否回到第一步
上面α选取为

$$\alpha_\star = \frac{2}{\lambda_{min}(B^T A^{-1} B) + \lambda_{max}(B^T A^{-1} B)}$$

2.2 Implementation

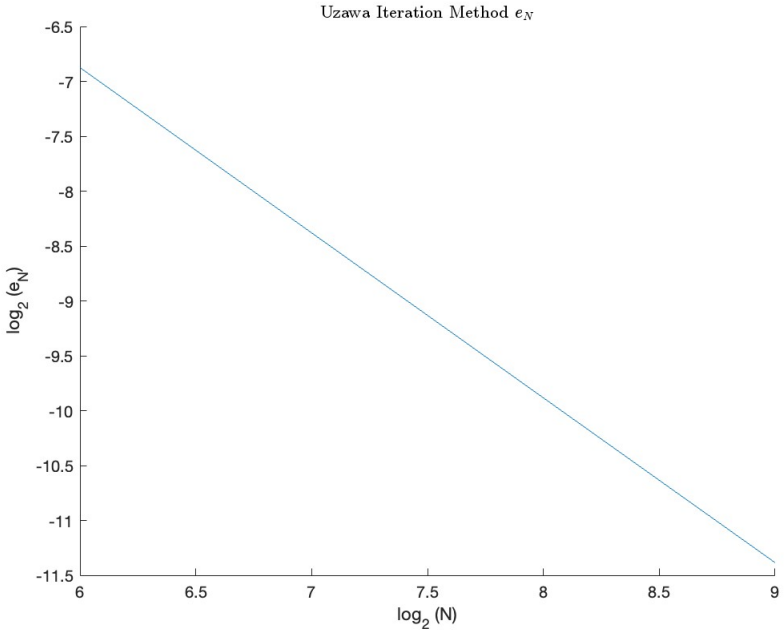
α选取需要做特征值的计算,所以我写了基本的计算最大特征值的函数,但发现最大特征值为1,数值实验表明α取1基本最优,故猜测最大与最小 特征值均等于或接近1
为了加快速度,经过数值实验过后(输出每一步迭代后的误差),发现刚开始误差减少飞快,到了10步左右误差就几乎不动弹,故得知是α太大,采用阶梯形α如下:

$$\alpha_k = \frac{\alpha}{max(k, 50) - 49}$$

从而大幅度提高速度(可能采用其他的α_k更快)
具体函数见code里的README.md

2.3 Result

α取初值为1,后面按照前面的方法阶梯下降



N	time	e_N	steps
64	0.13476	0.0085233	21
128	0.44345	0.0030007	36
256	4.7654	0.0010587	59
512	28.6095	0.00037395	55

由图易见$e_N = O(\frac{1}{N^2})$

3 Inexact Uzawa Iteration Method + Vcycle

3.1 Algorithm

算法框架如下:
1) 每一轮开始时, 用预条件共轭梯度法近似求解$AU_{k+1} = F - BP_k$, 使用$M = A$作为预条件矩阵, 求解预条件方程的时候使用Vcycle方法求解
2) 然后对压力进行更新$P_{k+1} = P_k + \alpha(B^T U_{k+1})$
3) 判断误差是否小于允许的值, 从而判断是否回到第一步
上面α选取为

$$\alpha_\star = \frac{2}{\lambda_{min} B^T A^{-1} B + \lambda_{max} B^T A^{-1} B}$$

近似求解$AU_{k+1} = F - BP_k$得到的$\hat{U}_{k+1}$需要满足若定义

$$\delta_k = A\hat{U}_{k+1} - F + BP_k$$

则

$$||\delta_k||_2 \leq \tau ||B^T \hat{U}_k||$$

3.2 Implementation

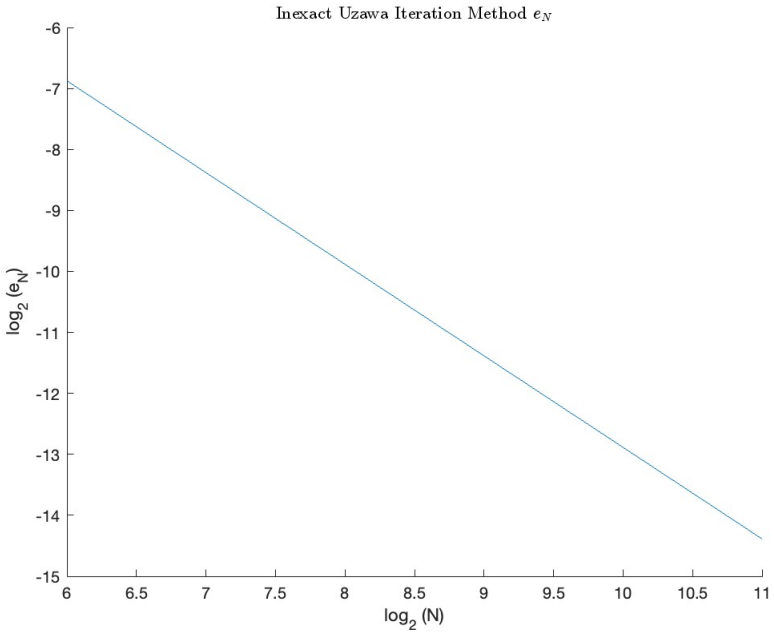
下面结果可以看见外层迭代次数较少, 故无需采用前面的α阶段下降方法
注意Vcycle作为预条件子, 其实求解的就是原先方程, 为了保证速度, 或者Vcycle预条件子求解的精度不能太高, 下面给出了我写的Vcycle精度要求

$$\frac{||r_h||}{||r_0||} \leq \max(\tau ||B^T \hat{U}_k||, 10^{-4}) \quad or \quad ||r_h|| \leq 10^{-8}$$

具体函数见code里的README.md

3.3 Result

采用参数$\nu_1 = 5, \nu_2 = 3, L = 2, \alpha = 1, \tau = 0.01$



N	time	e_N	steps
64	0.065143	0.0085233	4
128	0.25165	0.0030007	4
256	0.99247	0.0010587	4
512	4.0304	0.00037395	4
1024	24.4011	0.00013212	4
2048	129.3766	4.6674e-05	4

由图易见$e_N = O(\frac{1}{N^2})$
以下给出了一些其他参数的情况的表格以及解释:
若$\nu_1 = 10, \nu_2 = 3, L = 2, \alpha = 1, \tau = 0.01$

N	time	e_N	steps
64	0.0705	0.0085233	4
128	0.18218	0.0030007	4
256	0.74884	0.0010587	4
512	3.3119	0.00037395	4
1024	24.7459	0.00013215	4
2048	132.101	4.6709e-05	4

可以看见当ν_1不算太大的时候, 对效率和误差基本无影响
若$\nu_1 = 5, \nu_2 = 10, L = 2, \alpha = 1, \tau = 0.01$

N	time	e_N	steps
64	0.047336	0.0085233	4
128	0.192	0.0030007	4
256	0.7621	0.0010587	4
512	3.582	0.00037395	4
1024	27.3802	0.00013215	4

其中$N = 2048$无法收敛,可以看出虽然ν_2增大,且效率误差影响都很小， 但对收敛性有一定影响,说明τ取的不够小,或者Vcycle预条件子求解时 的不够精确导致共轭梯度没法收敛

若$\nu_1 = 5, \nu_2 = 3, L = 2, \alpha = 0.9, \tau = 0.01$

N	time	e_N	steps
64	0.063288	0.0085233	7
128	0.26009	0.0030007	7
256	1.1275	0.0010587	7
512	4.5841	0.00037395	7
1024	31.504	0.00013224	6
2048	171.6888	4.684e-05	6

可以看出α的小改动会对效率造成较大的影响

至于L和τ改动的影响,一个在section1已经说过类似的,一个比较显然