应用数学导论大作业

敖睿成

1 摘要

考虑方程

$$\begin{cases} \dfrac{\partial u}{\partial t} = \Delta u \\ \text{边值条件} \end{cases}$$

本次实验中，依次针对一维情形，$\Omega = [0,1] \times [0,1]$，L-型区域上的泊松问题上的泊松问题，分别使用有限元方法，和中心差分方法求解方程, 并给出相关理论分析和实验结果。

2 一维自适应有限元方法

对一维问题

$$\begin{cases} -u'' = f \qquad \Omega = (0,L) \\ u'(0) = \kappa_0(u(0) - g_0) \\ u'(L) = \kappa_L(u(L) - g_L) \end{cases}$$

对于 $N+1$ 个结点 $x_0 = 0, x_1 = \frac{L}{N}, ...x_N = L$, 在子区间 $[x_{i-1}, x_{i+1}](i = 1, 2, ..., N-1)$ 上取分段线性函数:

$$\phi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{h_i}, \qquad x \in [x_{i-1}, x_i] \\ \dfrac{x_{i+1} - x}{h_{i+1}}, \qquad x \in [x_i, x_{i+1}] \\ 0, \end{cases}$$

其中 $h_i = x_i - x_{i-1}$, 对于边界两个结点, 考虑 $x_0, x_1; x_{N-1}, x_N$ 对应的两个两点线性插值函数，这样得到 $N+1$ 个基函数 $\phi_0, \phi_1, ..., \phi_N$, 现求解 $v(x) = \sum_{j=0}^{N} u_j\phi_j(x)$, 使得

$$-\int_0^1 u''(x)v(x)dx = \int_0^1 f(x)v(x)dx$$

成立，通过变分方法得到:

$$\begin{cases} -\dfrac{u_{i-1}}{h_i} + (\dfrac{1}{h_i} + \dfrac{1}{h_{i+1}})u_i - \dfrac{u_{i+1}}{h_{i+1}} = \displaystyle\int_{x_{i-1}}^{x_{i+1}} f\phi'(x)dx \qquad i = 1, 2, ..., N-1 \\ (\dfrac{1}{h_1} + \kappa_0)u_0 - \dfrac{u_1}{h_1} = f_0 \\ (\dfrac{1}{h_N} + \kappa_L)u_N - \dfrac{u_{N-1}}{h_N} = f_N \end{cases}$$

令 $a_i = \frac{1}{h_i} + \frac{1}{h_{i+1}}, i = 1, 2, ..., N-1,$

$$A = \begin{pmatrix} \frac{1}{h_1} + \kappa_0 & -\frac{1}{h_1} & & & \\ -\frac{1}{h_1} & a_1 & -\frac{1}{h_2} & & \\ & -\frac{1}{h_2} & a_2 & \ddots & \\ & & \ddots & \ddots & -\frac{1}{h_N} \\ & & & -\frac{1}{h_N} & \frac{1}{h_N} + \kappa_L \end{pmatrix}$$

得到方程

$$Au = F$$

。分别应用均匀剖分和自适应方法，在总点数 $N = 10, 20, 80$ 时得到如下结果:
$\kappa_0 = 10^6, k_1 = 0, g_0 = 0, f(x) = e^{-100(x-0.5)^2}$ 时，

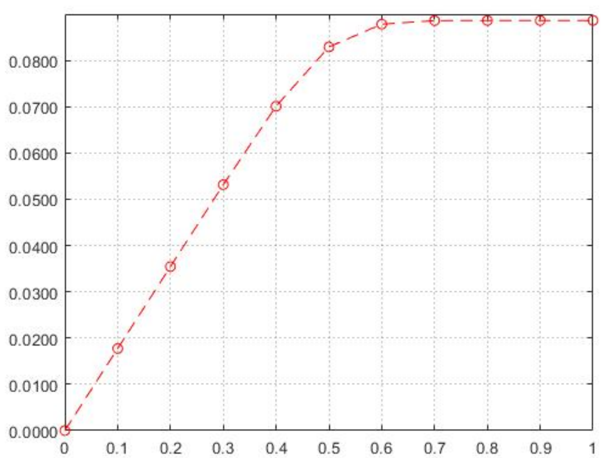

图 1: 均匀剖分 10

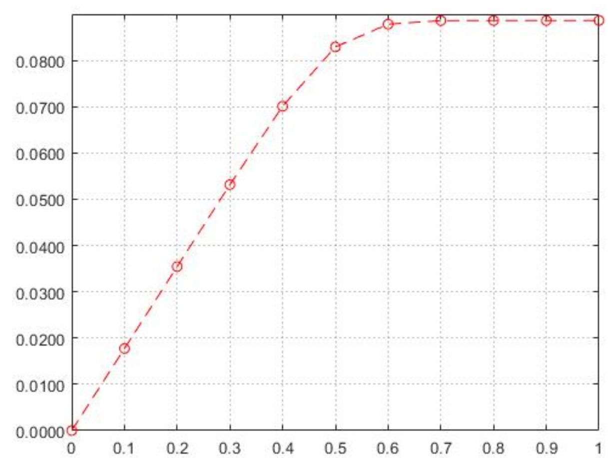

图 2: 自适应 10

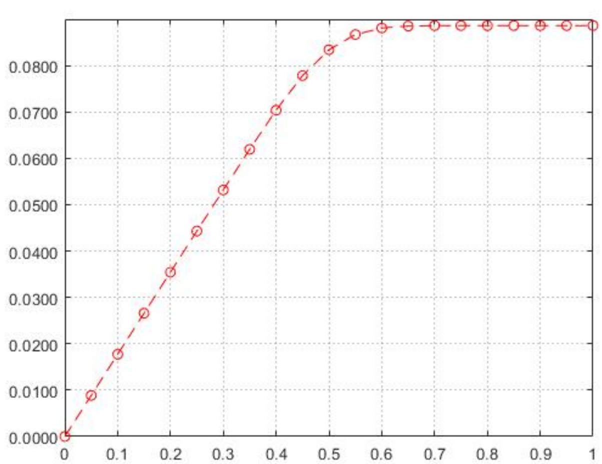

图 3: 均匀剖分 20

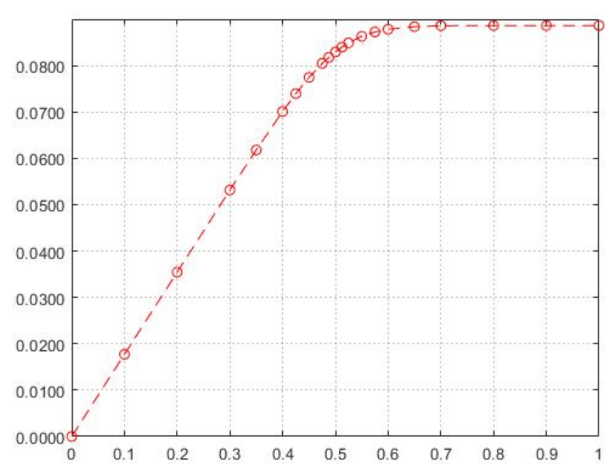

图 4: 自适应 20

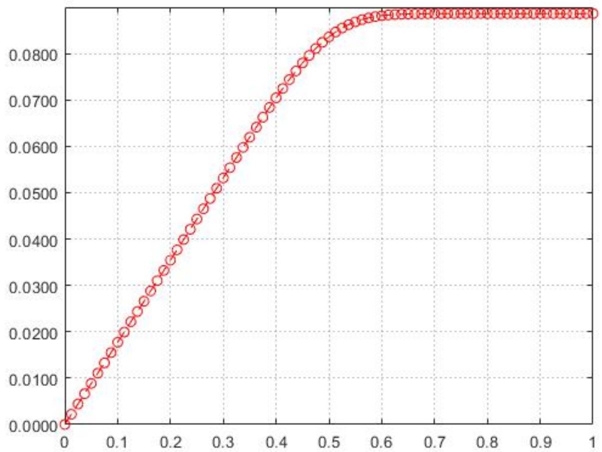

图 5: 均匀剖分 80

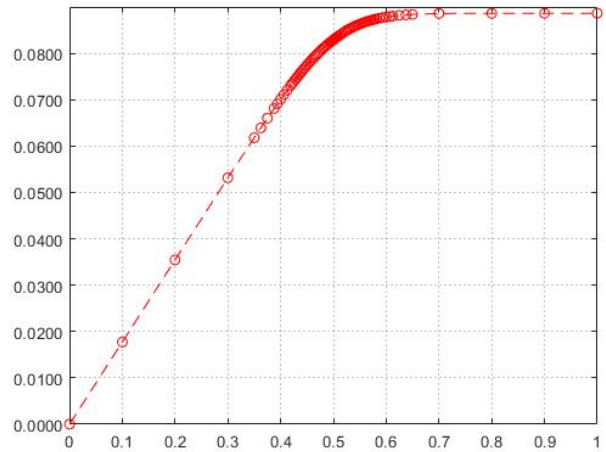

图 6: 自适应 80

$\kappa_0 = 10^6, k_1 = 10^5, g_0 = 0, g_L = 0, f(x) = e^{-100(x-0.5)^2}$ 时，

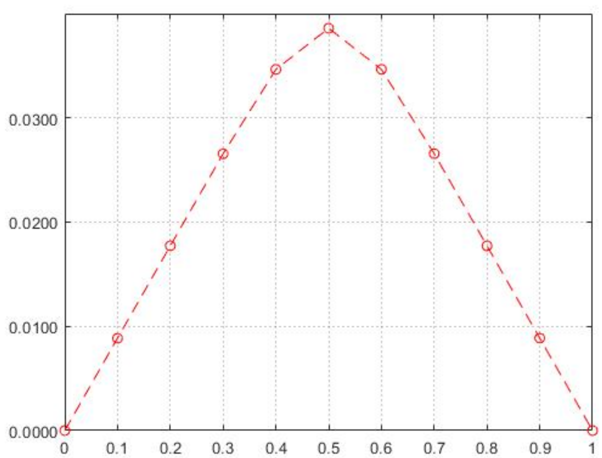

图 7: 均匀剖分 10

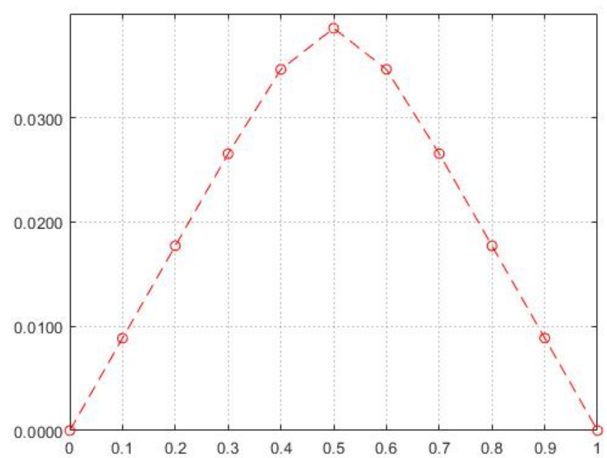

图 8: 自适应 10

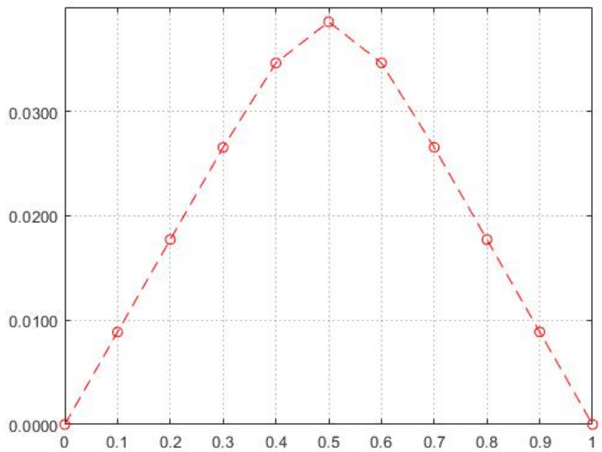

图 9: 均匀剖分 20

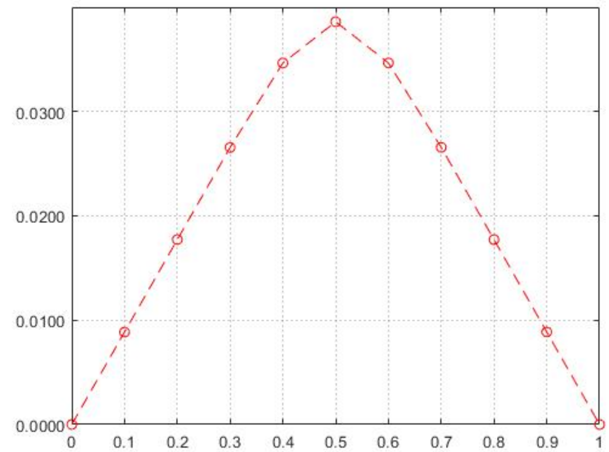

图 10: 自适应 20

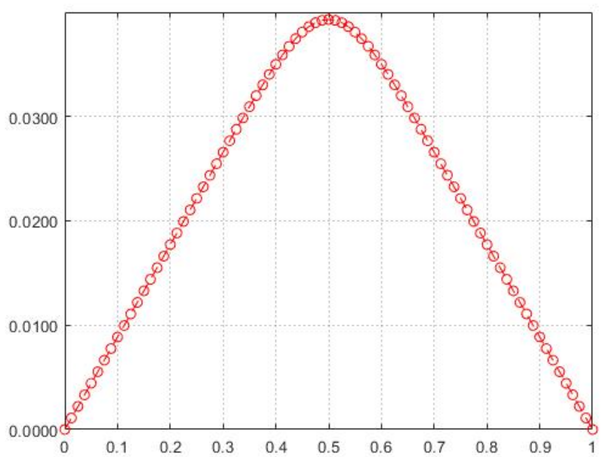

图 11: 均匀剖分 80

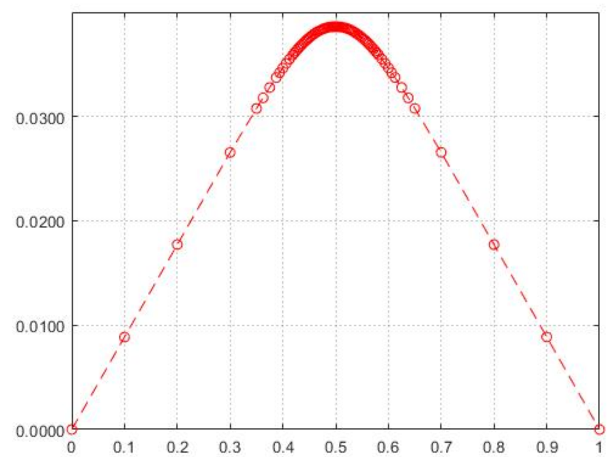

图 12: 自适应 80

$\kappa_0 = 10^6, k_1 = 0, g_0 = -1, f(x) = e^{-100(x-0.5)^2}$ 时，

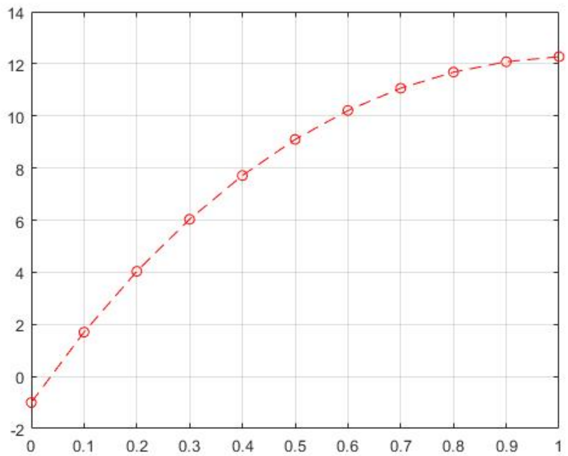

图 13: 均匀剖分 10

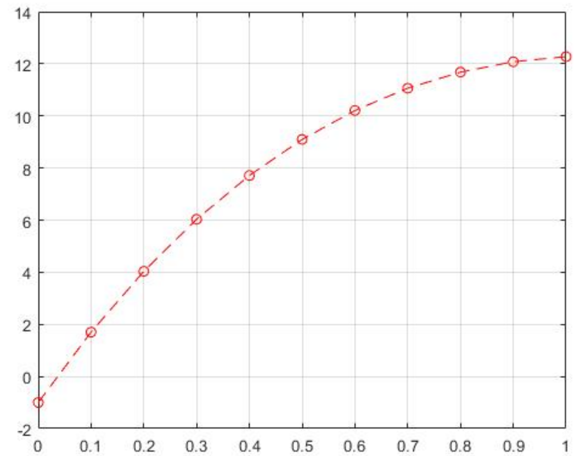

图 14: 自适应 10

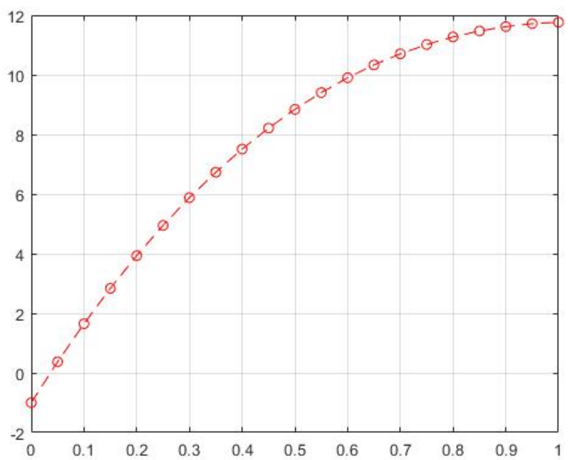

图 15: 均匀剖分 20

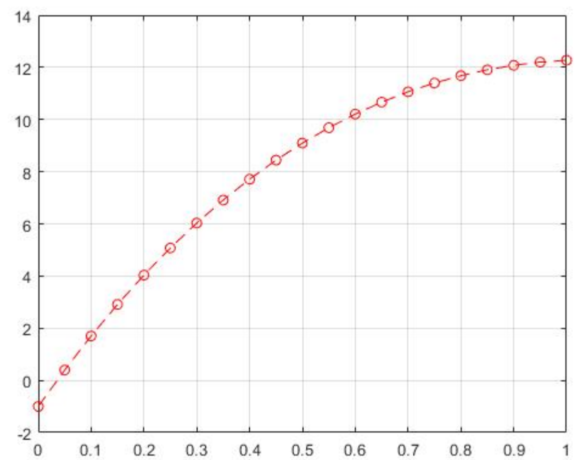

图 16: 自适应 20

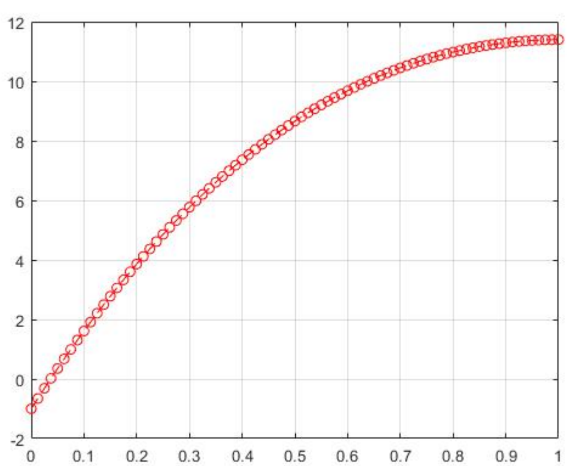

图 17: 均匀剖分 80

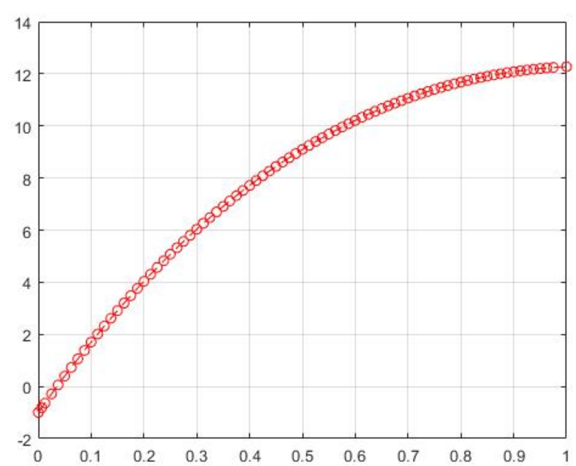

图 18: 自适应 80

可以看到，相比较均匀剖分，自适应方法在曲率大的点附近加密得较细，所得函数也更为平滑。

3 正方形区域泊松方程

3.1 问题

考虑热传导方程:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \Delta u \\ u|_{\partial\Omega\times[0,1]} = 0, \Omega = [0,1]\times[0,1] \\ u|_{t=0} = \sin(\pi x)\sin(\pi y) \end{cases}$$

它有解析解 $u = e^{-2\pi^2 t}\sin(\pi x)\sin(\pi y)$，我们使用不同数值方法计算方程近似解，并给出相关分析。

3.2 稳定性分析

考察 Laplace 算子

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

，我们使用中心差分方法

$$L_{h_x,h_y}U_{i,j}^m \overset{\text{def}}{=\!=} \frac{U_{i-1,j}^m - 2U_{i,j}^m + U_{i+1,j}^m}{h_x^2} + \frac{U_{i,j-1}^m - 2U_{i,j}^m + U_{i,j+1}^m}{h_y^2}$$

这里 h_x, h_y 为对应分量的区间步长，对于 $\frac{\partial u}{\partial t}$，我们使用一阶向前差分方法:

$$D_k U_{i,j}^m \overset{\text{def}}{=\!=} \frac{U_{i,j}^{m+1} - U_{i,j}^m}{k}$$

这里 k 为时间步长。现在考虑等式:

$$(1-\theta)L_{h_x,h_y}U_{i,j}^m + \theta L_{h_x,h_y}U_{i,j}^{m+1} = D_k U_{i,j}^m$$

当 $\theta = 1$ 时，为隐式格式，$\theta = 0.5$ 时，为 Crank-Nicolson 格式，$\theta = 0$ 时，为显式格式。令

$$\tilde{U} = u\left(ih_x, jh_y, \left(m + \frac{1}{2}\right)k\right)$$

应用 Taylor 公式，可以得到

$$\begin{aligned} L_{hx,hy}U_{i,j}^m =& \tilde{U}_{xx} + \tilde{U}_{yy} + \frac{2}{3!}\left(3\tilde{U}_{txx}\left(-\frac{1}{2}k\right) + 3\tilde{U}_{tyy}\left(-\frac{1}{2}k\right)\right) \\ &+ \frac{2}{4!}\left(\tilde{U}_{xxxx}h_x^2 + \tilde{U}_{yyyy}h_y^2\right) + O(k^2 + h_x^4 + h_y^4) \end{aligned}$$

$$\begin{aligned} L_{hx,hy}U_{i,j}^{m+1} =& \tilde{U}_{xx} + \tilde{U}_{yy} + \frac{2}{3!}\left(3\tilde{U}_{txx}\frac{1}{2}k + 3\tilde{U}_{tyy}\frac{1}{2}k\right) \\ &+ \frac{2}{4!}\left(\tilde{U}_{xxxx}h_x^2 + \tilde{U}_{yyyy}h_y^2\right) + O(k^2 + h_x^4 + h_y^4) \end{aligned}$$

从而有：

$$(1-\theta)L_{h_x,h_y}U_{i,j}^m+\theta L_{h_x,h_y}U_{i,j}^{m+1}-\Delta\tilde{U}=\left(\left(\theta-\frac{1}{2}\right)k+\frac{h_x^2}{12}\right)\tilde{U}_{xxxx}+\left(\left(\theta-\frac{1}{2}\right)k+\frac{h_y^2}{12}\right)\tilde{U}_{yyyy}$$
$$+\left(2\theta-1\right)k\tilde{U}_{xxyy}+O(k^2+h_x^4+h_y^4)$$

从而

$$(1-\theta)L_{h_x,h_y}U_{i,j}^m+\theta L_{h_x,h_y}U_{i,j}^{m+1}-\Delta\tilde{U}=\left(\left(\theta-\frac{1}{2}\right)k+\frac{h_x^2}{12}\right)\tilde{U}_{xxxx}+\left(\left(\theta-\frac{1}{2}\right)k+\frac{h_y^2}{12}\right)\tilde{U}_{yyyy}$$
$$+\left(2\theta-1\right)k\tilde{U}_{xxyy}+O(k^2+h_x^4+h_y^4)$$

这样截断误差 E 满足

$$E=\begin{cases}O(k^2+h_x^2+h_y^2) & \theta=0.5\\ O(k+h_x^2+h_y^2) & \theta\neq 0.5\end{cases}$$

可以看出，C-N 方法具有较高的截断误差阶数，现在考虑 Fourier 函数

$$U_{j,k}^m=\lambda_\alpha^m e^{i(\alpha_x x_j+\alpha_y y_k)},\quad \alpha=(\alpha_x,\alpha_y)$$

解得

$$\lambda_k=\frac{1-4(1-\theta)\left(\mu_x\sin^2(\alpha_x h_x/2)+\mu_y\sin^2(\alpha_y h_y/2)\right)}{1+4\theta\left(\mu_x\sin^2(\alpha_x h_x/2)+\mu_y\sin^2(\alpha_y h_y/2)\right)}$$

因此，我们有稳定性条件

$$\begin{cases}2(\mu_x+\mu_y)(1-2\theta)\leq 1 & 0\leq\theta<1/2\\ \text{无条件收敛} & 1/2\leq\theta\leq 1\end{cases}$$

这表明当 $h_x=h_y=h$ 时，对于显式格式，我们需要选取 $\mu<\frac{1}{4}$, 即 $1/h\geq 1/4k$，才能得到收敛结果。

3.3 数值实验

考虑由中间 $(N-1)\times(N-1)$ 个点，其中 $N=\frac{1}{h}$ 为单元网格长，则对应的差分矩阵

$$_h=\begin{pmatrix}A_h & I_{N-1}/h & & & \\ I_{N-1}/h^2 & A_h & I_{N-1}/h^2 & & \\ & I_{N-1}/h^2 & A_h & \ddots & \\ & & \ddots & \ddots & I_{N-1}/h^2\\ & & & I_{N-1}/h^2 & A_h\end{pmatrix}$$

$$A_h=\begin{pmatrix}-2/h^2-2/h^2 & 1/h^2 & & & \\ 1/h^2 & -2/h^2-2/h^2 & 1/h^2 & & \\ & 1/h^2 & -2/h^2-2/h^2 & \ddots & \\ & & \ddots & \ddots & 1/h^2\\ & & & 1/h^2 & -2/h^2-2/h^2\end{pmatrix}$$

这样，得到如下方程

$$(I-k\theta L_h)U^{m+1}=(I+k(1-\theta)L_h)U^m$$

其中 U 为对应的网格向量化。下面考虑几种求解对应方程的方法。

Cholesky 分解

考察方程 $Ax=b$, 其中 A 为对称正定矩阵，则我们有如下 Cholesky 分解用以求解方程：

Algorithm 1: 利用向量外积的 cholesky 分解

Input: $A\in\mathbb{R}^{n\times n}$

Output: $L\in\mathbb{R}^{n\times n}$, $LL^{\mathrm{T}}=A$

for $j=1:n$ **do**
  if $j>1$ **then**
    $A(j:n,j)=A(j:n,j)-A(j:n,1:j-1)A(j,1:j-1)^{\mathrm{T}}$
  $A(j:n,j)=A(j:n,j)/\sqrt{A(j,j)}$
return $tril(A)$;

计算量约为 $O(n^3/3)$, 是直接进行 Gauss 消元法的一半，在矩阵阶数较小时具有很快速度，但当矩阵阶数较大时，可以使用分块的方法加快速度。

Gauss-Seidel 迭代法

令 $A=D-L-U$, 其中 D,L,U 分别为对角矩阵、下三角矩阵和上三角矩阵，则我们有如下算法：

Algorithm 2: G-S 迭代法

Input: $A\in\mathbb{R}^{n\times n}$,$b\in\mathbb{R}^n$, **TOLERENCE** tol, **INITIALVALUE** x_0 **MAXITERATION** ite

Output: $x\in\mathbb{R}^n$, $Ax\approx b$

Divide A into D,L and U
$\mathbf{Ux}\leftarrow Ux_0$
while $norm(res)/norm(res_0)>tol\&\&ite_number\leq ite$ **do**
  Get x by solving $(D-L)x=\mathbf{Ux}+b$
  Update $res=Ux-\mathbf{Ux}$
  Update $\mathbf{Ux}\leftarrow res+\mathbf{Ux}$
  if $norm(res)\leq tol*res_0$ **then**
    return x
return x;

这里利用到了 $\mathrm{res}=b-Ax^{m+1}=b-(D-L)x^{m+1}+Ux^{m+1}=Ux^{m+1}-Ux^m$。可以证明，当 A 为对称正定矩阵时，G-S 迭代法是收敛的。

多重网格法

对于单元格长为 $h=1/N$ 的细网格，我们希望找到一个较好的初始值，从而加快迭代法收敛的速度，故考虑在细网格上先进行若干次迭代，再将误差限制在粗网格上，在粗网格解关于误差的方程，再把提升回细网格，这样两者相加，可以认为得到了更近的初值，再重复这样的操作，直到误差满足要求，这里可以重复限制、提升多层，算法如下：

Algorithm 3: Multi Grid V

Input: $A \in \mathbb{R}^{n\times n}$,$b \in \mathbb{R}^n$, **LEVEL** l, **MAXITERATION** ite_1, **INITIALVALUE** x_0

Output: $x \in \mathbb{R}^n$, $Ax \approx b$

if $size(x) < threshold$ **then**

  Solve $Ax = b$ by **G-S** with **INITIALVALUE** x_0

else

  Solve $Au = b$ by **G-S** with **INITIALVALUE** x_0 and **MAXITERATION** ite_1

  Get residue: $res \leftarrow b - Au$

  Get coarse residue: $\widehat{res} \leftarrow I_h^{2h} res$

  $\widehat{A} \leftarrow I_h^{2h} A\, I_{2h}^h$

  Solve $\widehat{A}\hat{e} = \widehat{res}$ with **EDGE** $2 * h$, **INITIALVALUE** 0 and **MAXITERATION** ite by **G-S**

  $e \leftarrow I_{2h}^h \hat{e}$

  Update $u \leftarrow u + e$

  Solve $Ax = b$ by **G-S** with **INITIALVALUE** u and **MAXITERATION** ite_2

return x;

这里 I_h^{2h}, I_{2h}^h 分别为限制和提升矩阵。当初始值较差时，多重网格的速度比直接 G-S 迭代快。