

应用数学导论大作业

吴睿林 2200010914
北京大学 数学科学学院



目录

2.1 有限元方法	1
2.1.a	1
2.1.b	1
2.1.c	1
2.2 自适应有限元方法	3
2.2.1	3
2.2.2	3
2.2.3	3
3 二维热传导方程	6

2.1 有限元方法

2.1.a

证明: 若 $f \in C^0([0, L]), u \in V \cap C^2([0, L])$ 是问题 (1.3) 的解, 则它也是问题 (1.1) 的解. 称 (1.3) 为 (1.1) 的变分问题.

解. 由分部积分公式可知

$$\int_0^L f v dx = \int_0^L \frac{du}{dx} \frac{dv}{dx} dx = \frac{du}{dx} v \Big|_0^L - \int_0^L \frac{d^2 u}{dx^2} v dx = - \int_0^L \frac{d^2 u}{dx^2} v dx$$

于是得到

$$\int_0^L \left(f + \frac{d^2 u}{dx^2} \right) v dx = 0, \quad \forall v \in V_0$$

由于 $f + \frac{d^2 u}{dx^2} \in C^0([0, L])$, 而 V_0 中的函数可逼近阶梯函数, 故有

$$f + \frac{d^2 u}{dx^2} = 0$$

2.1.b

证明: 有限元问题 (1.6) 的解 u_h 满足

$$\int_{\Omega} \frac{d(u - u_h)}{dx} \frac{dv_h}{dx} dx = 0 \quad \forall v_h \in V_{h,0}$$

解. 由于

$$\int_0^L \frac{du}{dx} \frac{dv_h}{dx} dx = \int_0^L f v_h dx = \int_0^L \frac{du_h}{dx} \frac{dv_h}{dx} dx$$

于是有

$$\int_0^L \frac{d(u - u_h)}{dx} \frac{dv_h}{dx} dx = 0 \quad \forall v_h \in V_{h,0}$$

2.1.c

当 $L = 1, f = 1$ 时, 写出问题 (1.1) 的变分形式; 使用有限元空间 $V_{h,0}$ 进行离散, 写出相应线性代数系统的刚度矩阵 A , 并使用追赶法求解 $Au = b$, 要求:

1. 写出变分形式;
2. 写出单元个数为 5 的时候 A 的具体形式;

3. 对 $n = 10, 20, 40, 80$ 的均匀剖分, 写一个有限元程序计算相应的矩阵 A , 并用追赶法求解线性系统 $Au = b$, 画出相应有限元解 u_h 的图像 (数值积分公式仍然使用梯形公式);
4. 当真解 $u = x(1-x)/2$, 对 $n = 2, 3, 8, 16, 32, 64, 128$ 的均匀剖分, 计算相应的误差 $\|\frac{d(u-u_h)}{dx}\|_{L^2(\Omega)}^2$, 观察 n 增加时, 误差变化的情况, 并做表格, 其中

$$\left\| \frac{d(u-u_h)}{dx} \right\|_{L^2(\Omega)}^2 = \left(\int_0^1 \left(\frac{d(u-u_h)}{dx} \right)^2 dx \right)^{\frac{1}{2}}$$

解. 1. 变分形式: 求 $u \in V_{h,0}$, 使得

$$\int_0^1 \frac{du}{dx} \frac{dv}{dx} dx = \int_0^1 v dx \quad \forall v \in V_{h,0}$$

2. 单元个数为 5 时 A 的具体形式为:

$$\begin{bmatrix} 10 & -5 & 0 & 0 \\ -5 & 10 & -5 & 0 \\ 0 & -5 & 10 & -5 \\ 0 & 0 & -5 & 10 \end{bmatrix}$$

3. 此时利用 numpy 中的 diag 函数可快速创建三对角矩阵 A , 利用追赶法可解得有限元解 u_h 的图像如下:

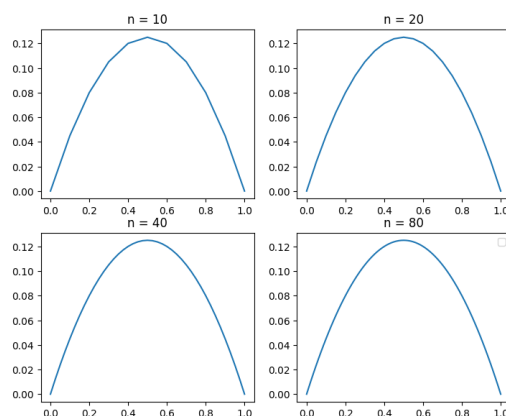


图 1: 不同的 n 对应有限元解 u_h 的图像

4. 利用梯形公式可算出误差如下:

$n = 2$	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$	$n = 128$
6.25×10^{-2}	1.56×10^{-2}	3.90×10^{-3}	9.76×10^{-4}	2.44×10^{-4}	6.10×10^{-5}	1.52×10^{-5}

2.2 自适应有限元方法

2.2.1

第一步: 已知 $L = 1, f = e^{-100(x-0.5)^2}$, 当 $0 < x < 1$ 时, 且 $u(0) = u(1) = 0$, 对 $n = 10$ 的均匀剖分, 用有限元方法求解该问题, 并画出有限元解.

解. 利用数值积分公式计算函数 f 与基底函数在对应区间上的积分, 得到 $n = 10$ 的有限元解如下:

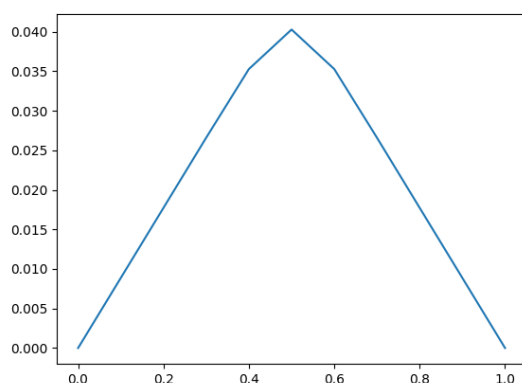


图 2: $n = 10$ 的有限元解

2.2.2

第二步: 当 $n < 20$, 编程计算各个单元 (小区间 $[x_i, x_{i+1}]$) 上的 $\eta_i = h_i(\int_{x_i}^{x_{i+1}} f^2(x)dx)^{1/2}$ 的值 (数值积分仍然使用梯形公式), 将所有 η_i 存入向量 $\eta, \eta = [\eta_1, \eta_2, \dots, \eta_n]$.

解. 利用数值积分梯形公式得到

$$\eta_i \approx h_i^{\frac{3}{2}} \left(\frac{f^2(x_i) + f^2(x_{i+1})}{2} \right)^{\frac{1}{2}}$$

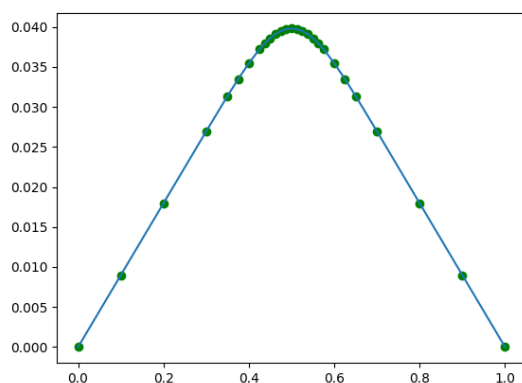
2.2.3

第三步: 设加密参数为 α , 取 $\alpha = 0.5$, 如果

$$\eta_i > \alpha \max_{1 \leq j \leq n} (\eta_j)$$

将单元 $[x_i, x_{i+1}]$ 二分加密, 计算有限元解 (转第二步)

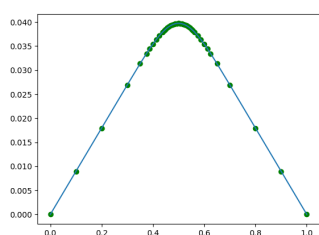
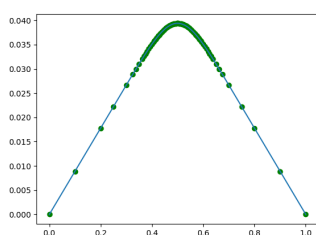
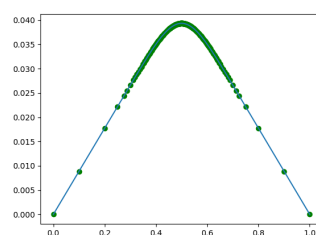
解. 初始网格取第一步中的 $n = 10$ 时的均匀剖分, 对加密参数 $\alpha = 0.5$ 改进网格, 限制网格数不超过 20, 由此算出的有限元解如下:

图 3: $n < 20, \alpha = 0.5$ 的自适应有限元解

在此图像中可以看到在函数最弯曲, 也就是二阶导数最大的地方网格越密集, 这与 η 的取法相关, 因为 η_i 与区间长度和其上函数 f 取值正相关, 函数 f 值越大的地方, 区间分划便会越细, 最终的网格便会越密集, 而 f 恰好对应解的二阶导数, 于是最终呈现的就是在解二阶导数大的地方网格越密集.

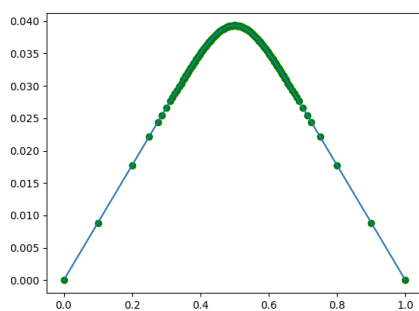
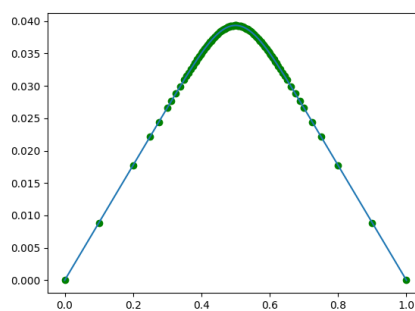
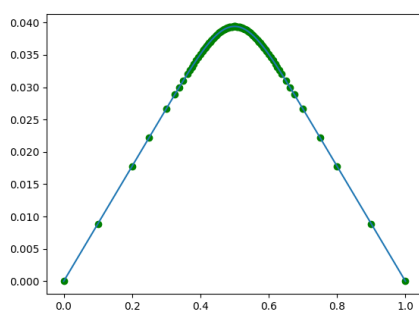
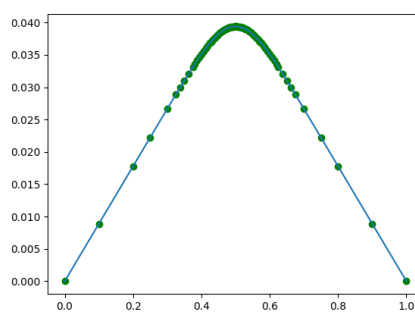
由于二阶导数为 0 的函数就是线性函数, 所以在二阶导数小的地方, 也就是二阶导近似于 0 的地方, 函数可以由线性函数良好逼近, 所以此自适应有限元算法将更多的网格集中在了解的“解的非线性部分”, 是相比于均匀剖分更加高效的有限元算法.

第四步: 考虑第二步中, 保持参数 $\alpha = 0.5$ 不变, 取 $n < 40, 80, 160$, 画出相应的有限元解及其网格的图像.

图 4: $n < 40, \alpha = 0.5$ 图 5: $n < 80, \alpha = 0.5$ 图 6: $n < 160, \alpha = 0.5$

解. 上图中可以看到随着网格点数的增多, 自适应算法会自动将网格集中在“解的非线性部分”, 对于解两侧的“几乎线性部分”, 自适应算法并没有分配新的网格点.

第五步: 考虑第三步中, 保持 $n = 80$ 不变, 取 $\alpha = 0.1, 0.25, 0.5, 0.9$, 画出相应的有限元解及其网格的图像.

图 7: $n < 80, \alpha = 0.1$ 图 8: $n < 80, \alpha = 0.25$ 图 9: $n < 80, \alpha = 0.5$ 图 10: $n < 80, \alpha = 0.9$

解. 从上图中, 固定相同的最大网格点数, 改变加密系数 α , 可以看到随着 α 的增大, 网格取法会更加集中于“解的非线性部分”, 也就是解的二阶导绝对值最大的部分. 这是因为在每次二分加密时, 更大的 α 会使得二分加密的区间更少, 只会让二阶导绝对值充分大的区间进行加密, 于是更大的 α 会导致网格点在“解的非线性部分”更为集中.

3 二维热传导方程

已知如下二维热传导方程:

$$\frac{\partial u(t, x, y)}{\partial t} = \Delta u(t, x, y), \quad (x, y) \in \Omega = (0, 1)^2, 0 < t \leq 1$$

$$u(t, x, y)|_{\partial\Omega} = 0, \quad 0 < t \leq 1$$

$$u(0, x, y) = \sin \pi x \sin \pi y, \quad (x, y) \in \Omega = (0, 1)^2$$

空间方向离散用连续分片线性有限元方法, 时间方向离散用显式, 隐式和 Crank-Nicolson 格式, 计算出有限元解 $u_h(1, x, y)$, 及误差 $\|\nabla(u(1, x, y) - u_h(1, x, y))\|_0$ 和 $\|u(1, x, y) - u_h(1, x, y)\|_0$, 其中空间步长为 $h = 1/4, 1/8, \dots, 1/128$, 时间步长自定.

解. 原方程的真解为 $u(t, x, y) = e^{-2\pi^2 t} \sin \pi x \sin \pi y$

考虑原方程解的弱形式: 乘以测试函数 $v \in C_0^1(\Omega)$ 并在整个区域 Ω 上积分得到

$$\int_{\Omega} u_t v d\Omega = \int_{\Omega} \Delta u v d\Omega = \int_{\partial\Omega} \nabla u v ds - \int_{\Omega} \nabla u \cdot \nabla v d\Omega$$

于是得到

$$\int_{\Omega} u_t v + \nabla u \cdot \nabla v d\Omega = 0$$

我们将 Ω 均匀剖分后将每个正方形再用对角线剖分得到 Ω 上的网格划分如下图

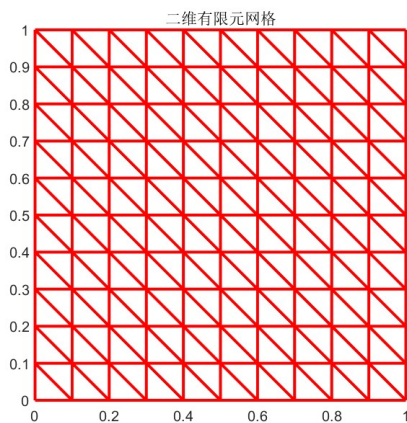


图 11: 剖分网格

其中基函数为一个顶点上取值为 1, 其余顶点取值为 0 的分片线性函数, 如下图.

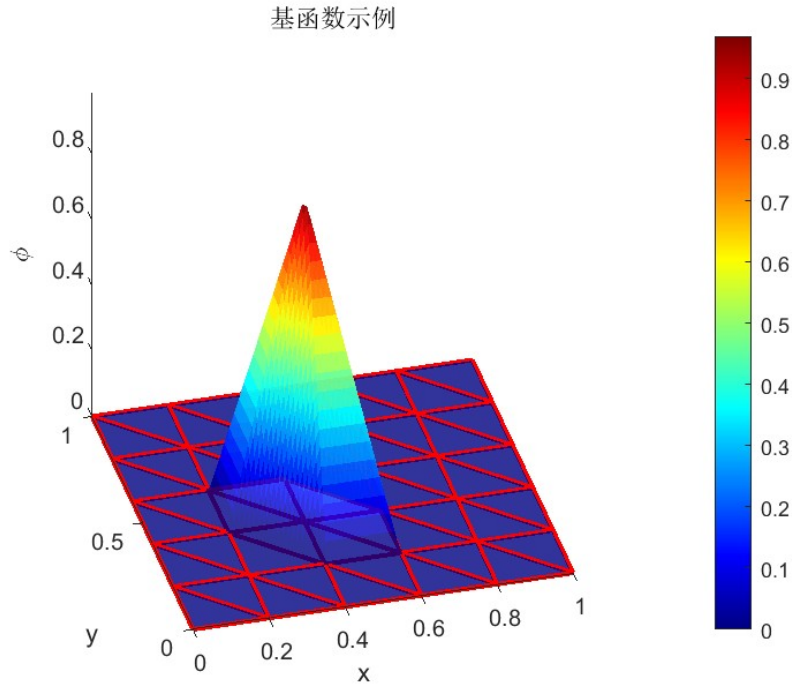


图 12: 基函数示例

设 $\phi_i, i = 1, \dots, N$ 为有限元的基函数, 其构成的空间为 V_h , 设 $\phi = (\phi_1, \dots, \phi_N)^\top$ 为基函数构成的向量, $u_h(t, x, y) = \phi^\top \cdot U(t)$ 为有限元解, 其中 $U(t)$ 为基函数的系数向量.

为使得弱形式的要求在 V_h 上成立, 分别带入 $v = \phi_i$ 得到

$$\int_{\Omega} \phi^\top \phi d\Omega \cdot \frac{dU}{dt} + \int_{\Omega} \nabla \phi^\top \nabla \phi d\Omega \cdot U = 0$$

将矩阵 $\int_{\Omega} \phi^\top \phi d\Omega$ 记为 Φ_1 , 矩阵 $\int_{\Omega} \nabla \phi^\top \nabla \phi d\Omega$ 记为 Φ_2 . 在空间步长确定之后, Φ_1, Φ_2 可精确计算.

将 $0 \leq t \leq 1$ 均等分为 n 份, 记 $U_t = U(\frac{t}{n})$, 于是在时间方向离散用显式, 隐式和 Crank-Nicolson 格式为:

$$\Phi_1 n(U_{i+1} - U_i) + \Phi_2 U_i = 0$$

$$\Phi_1 n(U_{i+1} - U_i) + \Phi_2 U_{i+1} = 0$$

$$\Phi_1 n(U_{i+1} - U_i) + \Phi_2 \frac{U_i + U_{i+1}}{2} = 0$$

而 $u(0, x, y)$ 已知, 故 U_0 已知, 可取 U_0 在每个节点上的值与 $u(0, x, y)$ 相同, 由上面变分形式得到 U_i 到 U_{i+1} 的递推, 由此可解得 U_n 故 $u_h = \phi^\top \cdot U_n$ 为 $u(1, x, y)$ 的近似解.

令 n 为时间分划个数, m 为空间分划个数, 即时间步长为 $\frac{1}{n}$, 空间步长为 $\frac{1}{m}$, 特别地取 $m = 32, n = 4000$ 时隐式解作为示例与精确解对比如下:

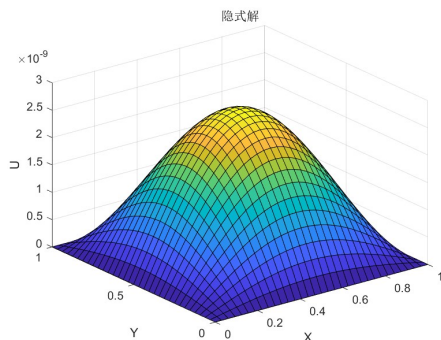


图 13: 有限元解

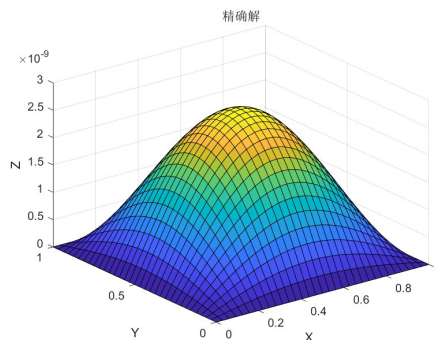


图 14: 精确解

有限元解 u_h 与精确解 $u(1, x, y)$ 的误差 $\|u(1, x, y) - u_h(1, x, y)\|_0$ 如下:

(m, n)	(4, 1000)	(8, 2000)	(16, 4000)	(32, 16000)	(64, 64000)	(128, 256000)
显式	1.29×10^{-9}	7.91×10^{-10}	2.92×10^{-10}	7.96×10^{-11}	2.03×10^{-11}	5.11×10^{-12}
(m, n)	(4, 4000)	(8, 4000)	(16, 4000)	(32, 4000)	(64, 4000)	(128, 4000)
隐式	1.27×10^{-9}	6.98×10^{-10}	1.83×10^{-10}	1.94×10^{-12}	4.94×10^{-11}	6.21×10^{-11}
(m, n)	(4, 4000)	(8, 4000)	(16, 4000)	(32, 4000)	(64, 4000)	(128, 4000)
C-N格式	1.28×10^{-9}	7.31×10^{-10}	2.39×10^{-10}	6.42×10^{-11}	1.63×10^{-11}	4.15×10^{-12}

表 1: $\|u(1, x, y) - u_h(1, x, y)\|_0$

有限元解 u_h 与精确解 $u(1, x, y)$ 梯度的误差 $\|\nabla(u(1, x, y) - u_h(1, x, y))\|_0$ 如下:

(m, n)	(4, 1000)	(8, 2000)	(16, 4000)	(32, 16000)	(64, 64000)	(128, 256000)
显式	5.76×10^{-9}	3.55×10^{-9}	1.37×10^{-9}	4.48×10^{-10}	1.69×10^{-10}	7.6×10^{-11}
(m, n)	(4, 4000)	(8, 4000)	(16, 4000)	(32, 4000)	(64, 4000)	(128, 4000)
隐式	5.69×10^{-9}	3.15×10^{-9}	9.58×10^{-10}	2.91×10^{-10}	2.66×10^{-10}	2.86×10^{-10}
(m, n)	(4, 4000)	(8, 4000)	(16, 4000)	(32, 4000)	(64, 4000)	(128, 4000)
C-N格式	5.70×10^{-9}	3.29×10^{-9}	1.16×10^{-9}	3.98×10^{-10}	1.61×10^{-10}	7.50×10^{-11}

表 2: $\|\nabla(u(1, x, y) - u_h(1, x, y))\|_0$

由于解的数量级在 10^{-9} , 而误差的数量级普遍小于 10^{-10} , 由上述误差可以看到有限元解在空间步长也就是网格边长到达 8 时就能很好地拟合偏微分方程的解.

时间方向对于隐式和 Crank-Nicolson 格式, 时间方向步长我统一选定的 $\frac{1}{4000}$, 但对于显式差分方法行不通, 迭代过程并不收敛, 对于显式差分方法, 需要随空间步长调节时间步长, 由此可以看出在该问题上显式方法稳定性不及隐式和 Crank-Nicolson 格式.