

偏微分方程数值解上机作业 1

王崇斌 2201110455

2022 年 10 月 30 日

1 单位圆盘上的 Poisson 方程

$$\begin{cases} -\Delta u = 1, & (x, y) \in B \\ u(x, y) = 0, & (x, y) \in \partial B \\ B = \{(x, y), x^2 + y^2 < 1\} \end{cases} \quad (1)$$

1.1 极坐标中的差分格式

利用极坐标中 Laplace 算子的表达式 $\Delta = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}$, 可以将上述直角坐标中的 Poisson 方程改写为如下形式, 额外补充了圆心处的有界性条件:

$$\begin{cases} -\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial u}{\partial r} \right) - \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} = 1, & (r, \theta) \in (0, 1) \times [0, 2\pi) \\ u(1, \theta) = 1, & \theta \in [0, 2\pi) \\ \lim_{r \rightarrow 0^+} \frac{\partial u}{\partial r} = 0, & \theta \in [0, 2\pi) \end{cases} \quad (2)$$

可以看到方程与边界条件都具有 θ 方向的平移对称性 (或者说是直角坐标中的旋转对称性), 因此可以推断方程的解只是 r 的函数, 上述方程进一步可以改写为一个二阶常微分方程的边值问题:

$$\begin{cases} -\frac{1}{r} \frac{d}{dr} \left(r \frac{du}{dr} \right) = 1, & r \in (0, 1) \\ u(1) = 0 \\ \left. \frac{du}{dr} \right|_{r=0^+} = 0 \end{cases} \quad (3)$$

为了利用半整点法构造 $r = 0$ 处的边界条件, 需要仔细构造网格, 令 $\Delta r = \frac{1}{N-0.5}$, $r_j = \frac{j-0.5}{N-0.5}$, $j = 0, 1, \dots, N$, 在这样的网格下 $r_{0.5} = \frac{r_0 + r_1}{2}$ 为数轴上 $r = 0$ 的点, 采用中心差分逼近内点处的导数, 在 r_N 处采用 Dirichlet 边界条件, 在 $r = 0$ 处采用半整点法构造边界条件, 得到如下差分格式:

$$\begin{cases} -\frac{1}{\Delta r^2 \cdot r_j} [r_{j+0.5} (u_{j+1} - u_j) - r_{j-0.5} (u_j - u_{j-1})] = 1, & j = 1, \dots, N-1 \\ u_N = 0 \\ \frac{u_1 - u_0}{\Delta r} = 0 \end{cases} \quad (4)$$

利用边界条件, 上述差分方程可以转化为 $N-1$ 个未知数的三对角线性方程组:

$$\begin{bmatrix} \frac{r_{1.5}}{r_1} & -\frac{r_{1.5}}{r_1} & & & \\ & \ddots & & & \\ & & -\frac{r_{j-0.5}}{r_j} & 2 & -\frac{r_{j+0.5}}{r_j} \\ & & & \ddots & \\ & & & -\frac{r_{N-1.5}}{r_{N-1}} & 2 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ \vdots \\ u_j \\ \vdots \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} \Delta r^2 \\ \vdots \\ \Delta r^2 \\ \vdots \\ \Delta r^2 \end{bmatrix} \quad (5)$$

求解这个线性方程组, 即可得到 Poisson 方程的解.

1.2 精确解与数值解的比较

将方程 [3] 两边同时乘 r 后积分, 再除以 r 积分, 可以得到不考虑边值时的方程的通解:

$$u = -\frac{r^2}{4} + C_1 \ln r + C_2 \quad (6)$$

带入 [3] 的边值条件后可以得到方程的精确解:

$$u = -\frac{r^2}{4} + \frac{1}{4} = -\frac{x^2 + y^2}{4} + \frac{1}{4} \quad (7)$$

数值计算时采用 LU 分解求解线性方程组, 使用了之前学习某课程时编写的数值线性代数代码 (My_Matrix.cpp, My_Matrix.h), 由于矩阵很小因此没有对三对角求解做特殊优化. 计算、编译、绘图程序分别参见 1_poisson_solver.cpp, compile1.bat, 1_poisson_plot.py. 将 $N_r = 8, 16$ 的数值结果与精确结果绘图比较, 参见图 (1), (2), 精度竟然会如此之高.

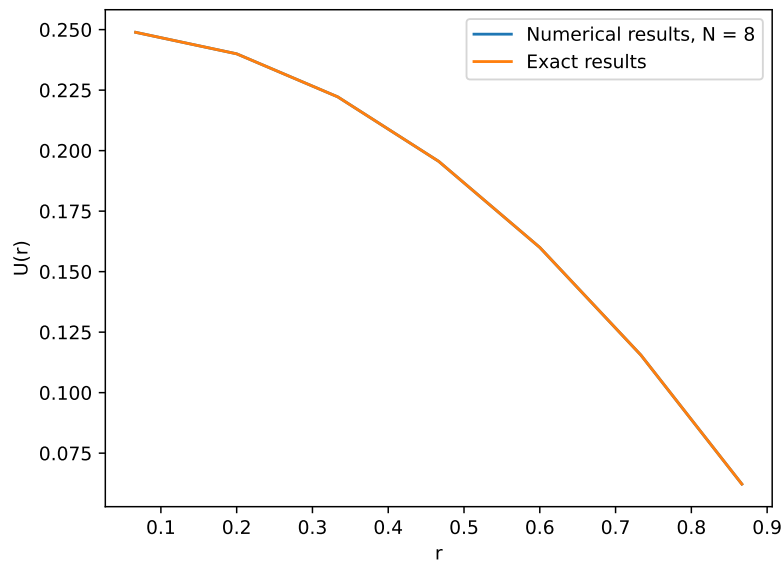
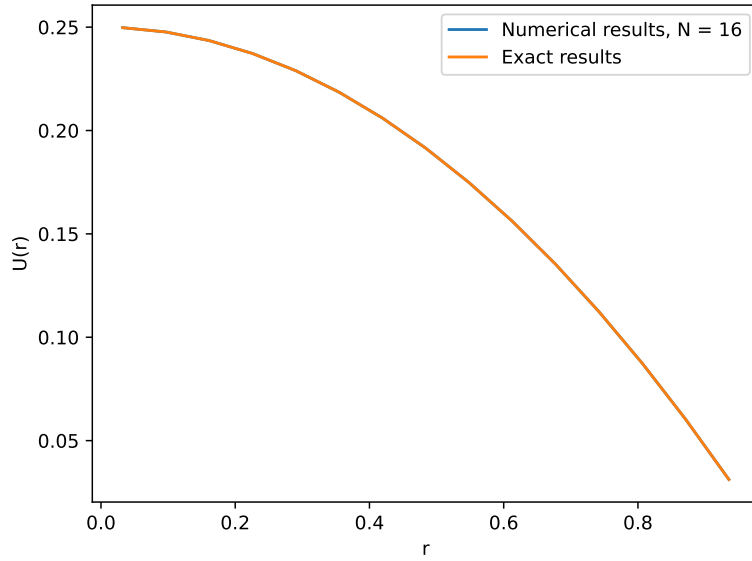


图 1: $N_r = 8$ 时数值解与精确解的结果比较

图 2: $N_r = 16$ 时数值解与精确解的结果比较

2 扩散方程的初边值问题

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, & x \in (0, 1), t > 0 \\ u(x, 0) = \frac{x^2}{2}, & x \in [0, 1] \\ u(0, t) = 0, \frac{\partial u}{\partial x}(1, t) = 1, & t > 0 \end{cases} \quad (8)$$

2.1 显格式的后验误差估计

这一题由于给定了 Δx , 使用半整点法无法解决, 被迫尝试使用影子节点法构造边界条件, 在 $[0, 1]$ 之间按照题目给定间距 Δx 插入 N 个点, 使得 $x_0 = 0, x_N = 1$, 在 x_N 右侧 Δx 处插入影子节点 x_{N+1} , 差分方程可以写为:

$$\begin{cases} \frac{u_j^{m+1} - u_j^m}{\Delta t} = \frac{1}{\Delta x^2} (u_{j+1}^m - 2u_j^m + u_{j-1}^m), & j = 1, 2, \dots, N \\ u_j^0 = \frac{x_j^2}{2}, & j = 0, 1, \dots, N \\ u_0^m = 0, & m > 0 \\ \frac{u_{N+1}^m - u_{N-1}^m}{2\Delta x} = 1, & m > 0 \end{cases} \quad (9)$$

将边界条件带入可以得到如下的显式递推关系, 其中网格比 $\mu = \frac{\Delta t}{\Delta x^2}$:

$$\begin{cases} u_j^{m+1} = \mu(u_{j+1}^m + u_{j-1}^m) + (1 - 2\mu)u_j^m, & j = 1, 2, \dots, N-1 \\ u_{N-1}^{m+1} = 2\mu(\Delta x + u_{N-1}^m) + (1 - 2\mu)u_N^m \end{cases} \quad (10)$$

编写程序 2_diffusion_explicit.cpp 计算不同网格参数时 $T = 1$ 的解 (编译使用命令 `g++ 2_diffusion_explicit.cpp -o 2_diffusion_explicit.exe -O3`), 使用程序 2_1_error.py 计算收敛阶, 计算时使用公式:

$$\ln ||U_h - U_{h/2}|| \approx \ln[(1 - 2^{-\alpha})C] - \alpha \ln h^{-1} \quad (11)$$

可以得到如图 (3) 所示的收敛阶示意图, 选取后两个数据点计算收敛阶, 可以得到两种范数下的收敛阶为

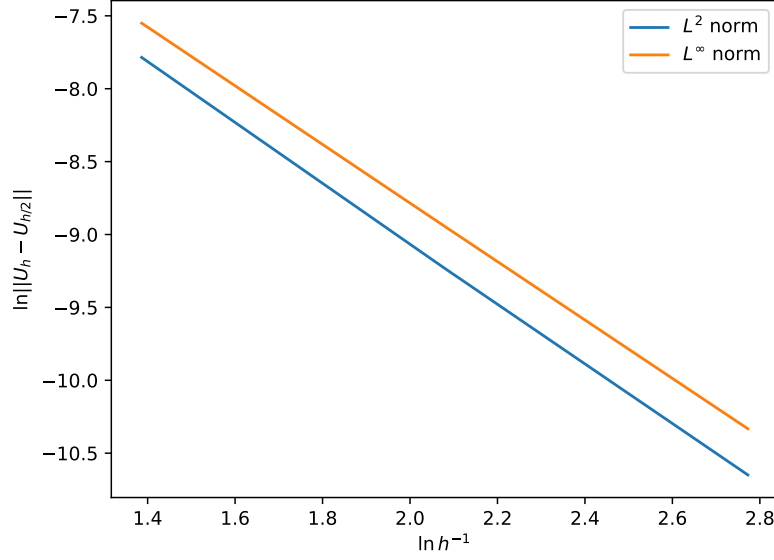


图 3: 扩散方程显式格式后验误差分析示意图

$$\alpha_{L^2} = 2.044, \alpha_{L^\infty} = 2.003$$

2.2 Crank-Nicolson 格式

同样使用影子节点法处理右端边界条件, 将 Crank-Nicolson 格式应用于此问题, 可以给出如下差分格式:

$$\begin{cases} \frac{u_j^{m+1} - u_j^m}{\Delta t} = \frac{1}{2} \left(\frac{u_{j+1}^m - 2u_j^m + u_{j-1}^m}{\Delta x^2} + \frac{u_{j+1}^{m+1} - 2u_j^{m+1} + u_{j-1}^{m+1}}{\Delta x^2} \right), & j = 1, 2, \dots, N \\ u_0^m = 0, & m > 0 \\ \frac{u_{N+1}^m - u_{N-1}^m}{2\Delta x} = 1, & m > 0 \end{cases} \quad (12)$$

上经过一些运算, 上式所示的单步演化可以转化为一个三对角线性方程组的求解, 其中已经带入了 $x = 0, x = 1$ 处边界条件:

$$\begin{bmatrix} \mu + 1 & -\frac{\mu}{2} & & & \\ & \ddots & & & \\ & & -\frac{\mu}{2} & \mu + 1 & -\frac{\mu}{2} \\ & & & \ddots & \\ & & & & -\mu & \mu + 1 \end{bmatrix} \cdot \begin{bmatrix} u_1^{m+1} \\ \vdots \\ u_j^{m+1} \\ \vdots \\ u_N^{m+1} \end{bmatrix} = \begin{bmatrix} \frac{\mu}{2}u_2^m + (1 - \mu)u_1^m \\ \vdots \\ \frac{\mu}{2}u_{j+1}^m + (1 - \mu)u_j^m + \frac{\mu}{2}u_{j-1}^m \\ \vdots \\ 2\mu\Delta x + \mu u_{N-1}^m + (1 - \mu)u_N^m \end{bmatrix} \quad (13)$$

编写程序 2_diffusion_CN.cpp 计算不同时刻的 u (编译使用 compile2.bat), 线性方程组的求解采用 LU 分解方法, 使用程序 2_2_CN_plot.py 绘制图像. 将以 $u(x, 0) = \frac{x^2}{2}$ 为初值的解作图, 参见图 (4). 在程序 2_diffusion_CN.cpp 中将初始条件替换为 $u(x, 0) = x^2 - x$, 重复编译计算并绘图, 参见图 (5). 观察这两种不同初值条件下扩散方程解的行为, 发现两者在足够长的时间后趋于同样的解 $u(x) = x$, 这说明在适当的边界条件下 (比如一端流入一端流出, 总不能是两端都流入), 扩散方程存在稳态解 $u(x, +\infty)$, 如果稳态解存在, 对应的就是同样边界条件下 Laplace(Poisson) 方程的解.

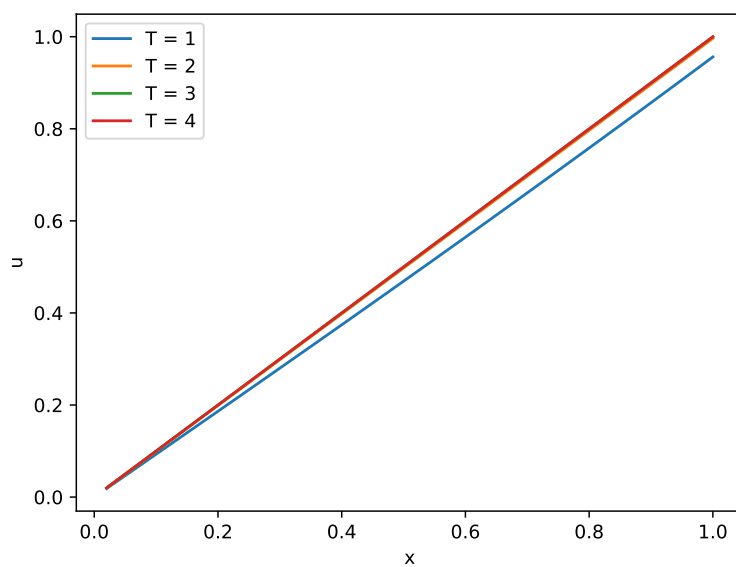


图 4: Crank-Nicolson 格式求解以 $u(x, 0) = \frac{x^2}{2}$ 为初值的扩散方程的解

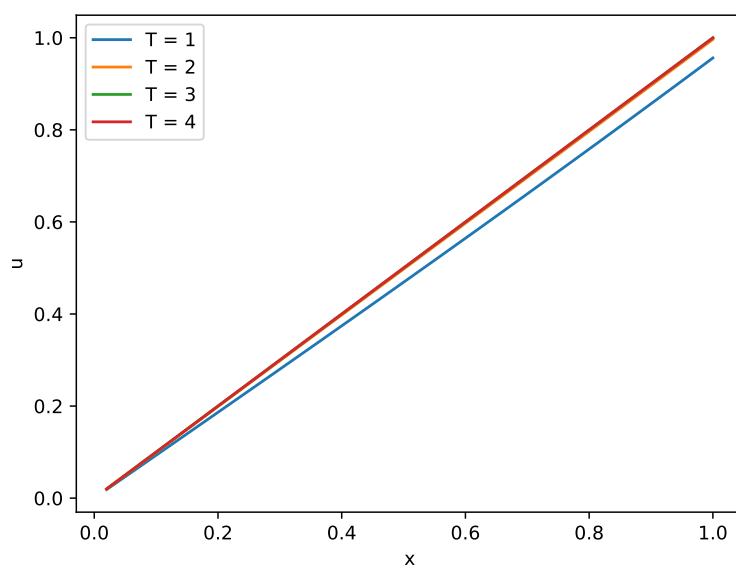


图 5: Crank-Nicolson 格式求解以 $u(x, 0) = x^2 - x$ 为初值的扩散方程的解

3 一阶波方程的初边值问题

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0, & x \in (0, 5), t > 0 \\ u(x, 0) = e^{-5(x-1)^2}, & x \in [0, 5] \\ u(-1, t) = 0, & t > 0 \end{cases} \quad (14)$$

3.1 一阶 upwind 方法的后验误差估计

由于不涉及复杂边界条件的处理, 可以采用 $[-1, 5]$ 之间间距为 Δx 的均匀网格, 由于流速大于 0, 迎风格式可以写为:

$$\begin{cases} u_j^{m+1} = (1 - \nu)u_j^m + \nu u_{j-1}^m, & m > 0, j = 1, 2, \dots, N \\ u_0^m = 0, & m > 0 \\ u_j^0 = e^{-5(x_j-1)^2}, & j = 0, 1, \dots, N \end{cases} \quad (15)$$

其中网格比选为 $\nu = 0.5$, $\Delta t = \nu \cdot \Delta x$, 编写程序 3_1_upwind.cpp 计算不同 Δx 时的数值结果, 使用程序 3_1_error.py(后验误差估计的逻辑与第二题类似), 可以得到如图 (6) 所示的收敛阶示意图, 可以根据后两个

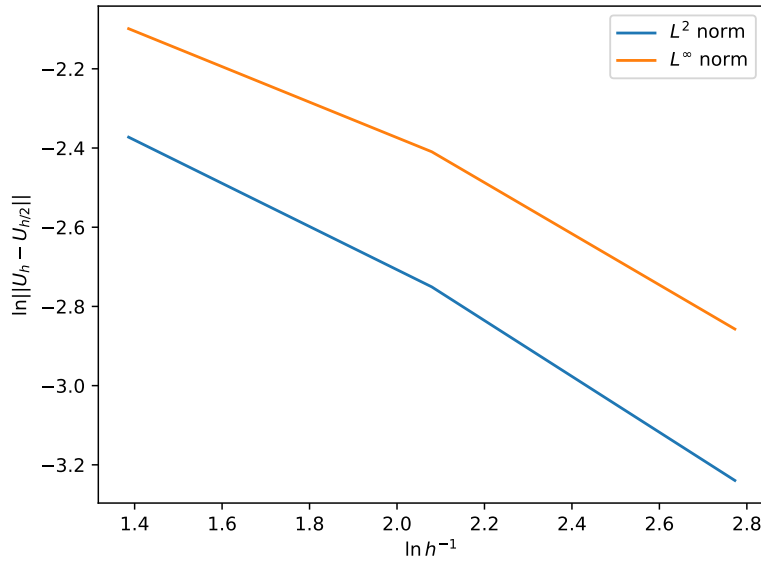


图 6: upwind 格式的后验误差分析示意图

数据点计算出两种范数下的收敛阶为 $\alpha_{L^2} = 0.705$, $\alpha_{L^\infty} = 0.646$, 能够看出收敛阶并不是非常接近理论值 1, 观察示意图, 可以看出收敛阶还没有对 Δx 收敛, 这可能是因为 upwind 格式精度较低导致收敛阶收敛变慢, 预计进一步减小 Δx 可能会得到更准确的收敛阶. 令 $\Delta x = \frac{1}{64}, \frac{1}{128}, \frac{1}{256}$, 重新绘制收敛阶的示意图 (7), 此时计算出两种范数下的收敛阶为 $\alpha_{L^2} = 0.950$, $\alpha_{L^\infty} = 0.940$, 这就体现出了理论上的收敛阶.

3.2 不同网格比时 upwind 格式的行为

继续使用上一题的程序, 修改其中 $\nu = 0.5, 1, 2$, 维持 $\Delta x = \frac{1}{32}$, 将得到的解绘制在图 (8) 中, 观察网格比 μ 不同的解, 可以观察到 $\nu = 1$ 时的解与精确解严格一致, 而 $\nu = 0.5$ 数值解有明显的耗散现象 (波峰降低). $\nu = 2$ 的解不满足 CFL 条件, 可以看出高斯波形的边缘开始出现小于 0 的部分 (震荡), 而在波峰处开始出现增长 (发散), 这表明不满足 CFL 条件的数值解不能正确描述方程的演化.

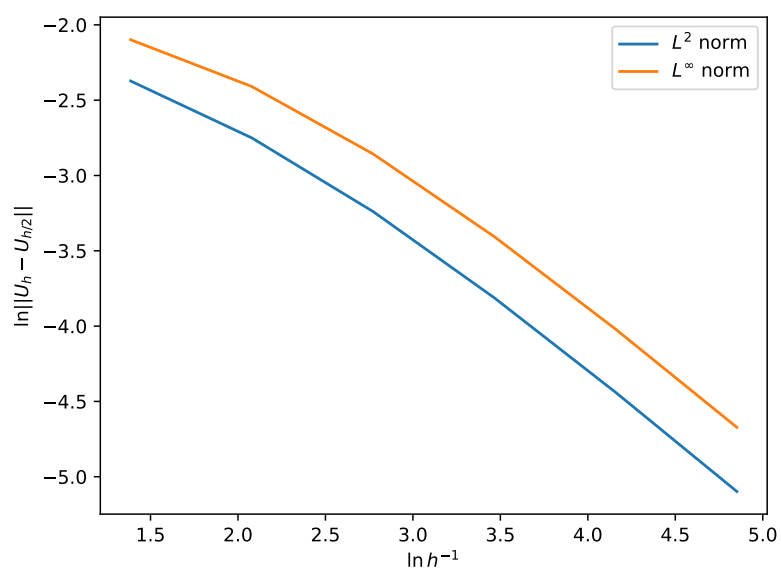
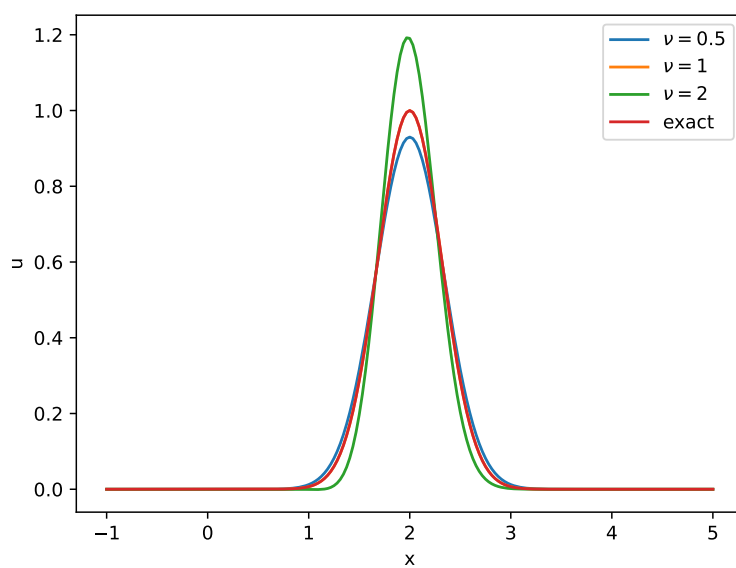


图 7: upwind 格式的后验误差分析示意图

图 8: 使用不同网格比 $\nu = 0.5, 1, 2$ 时波方程的解

利用 Von-Neumann 分析, 能够得到增长因子和相位的表达式:

$$\begin{cases} |\lambda_k|^2 = 1 - 4\nu(1 - \nu) \sin^2\left(\frac{k\Delta x}{2}\right) \\ \arg \lambda_k = -\arctan\left[\frac{\nu \sin k\Delta x}{(1 - \nu) + \nu \cos k\Delta x}\right] \end{cases} \quad (16)$$

可以看到当 $\nu = 1$ 时, 对于任意波数 k 的 Fourier 波形, 均不会产生振幅和相位的误差, 因此从理论上讲此种情形的数值解是精确的.

3.3 Lax-Wendroff 格式

如果利用 Lax-Wendroff 格式, 需要在右边边界上补充一个边界条件, 为了更好观察到高斯波形的传播 (也是为了便于数值计算), 我打算采用无反射边界条件, 补充影子节点 $u_{N+1}^m = u_N^m$, 可以构造如下的差分格式:

$$\begin{cases} u_j^{m+1} = -0.5\nu(1 - \nu)u_{j+1}^m + (1 - \nu^2)u_j^m + 0.5\nu(1 + \nu)u_{j-1}^m, & j = 1, 2, \dots, N \\ u_N^m = u_{N+1}^m \end{cases} \quad (17)$$

将边界条件带入可以得到显式递推关系:

$$\begin{cases} u_0^m = 0 \\ u_j^{m+1} = -0.5\nu(1 - \nu)u_{j+1}^m + (1 - \nu^2)u_j^m + 0.5\nu(1 + \nu)u_{j-1}^m, & j = 1, 2, \dots, N-1 \\ u_N^{m+1} = (-0.5\nu + 1 - 0.5\nu^2)u_N^m + 0.5\nu(1 + \nu)u_{N-1}^m \end{cases} \quad (18)$$

在计算中采用 $\nu = 0.5$, $\Delta x = 1/100$, 编写程序 3_3_Lax_Wendroff.cpp 计算数值结果, 编写程序 3_3_LW_plot.py 绘制图像, 如图 (9)(10)(11)(12) 所示.

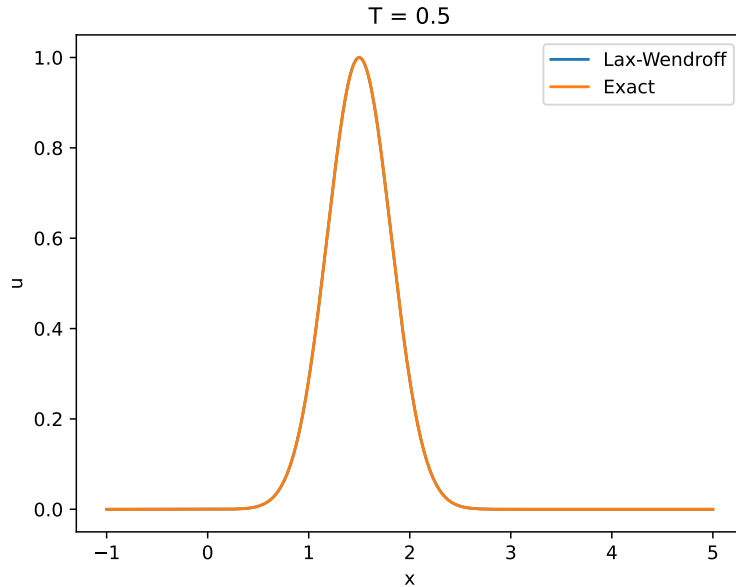


图 9: $T = 0.5$ 时使用 Lax-Wendroff 格式求解与精确解的对比

从图中可以看出, Lax-Wendroff 作为一种高阶格式, 其精度相较于 upwind 有着本质的提升, 仔细比较可以看出, 数值解在振幅和最值的角度很好地逼近了精确解, 而在相位上略慢于真实解, 这与理论上对 Lax-Wendroff 格式的增长因子的分析是一致的 ($O(\Delta x^3)$ 的振幅误差, $O(\Delta x^2)$ 的相对相位误差).

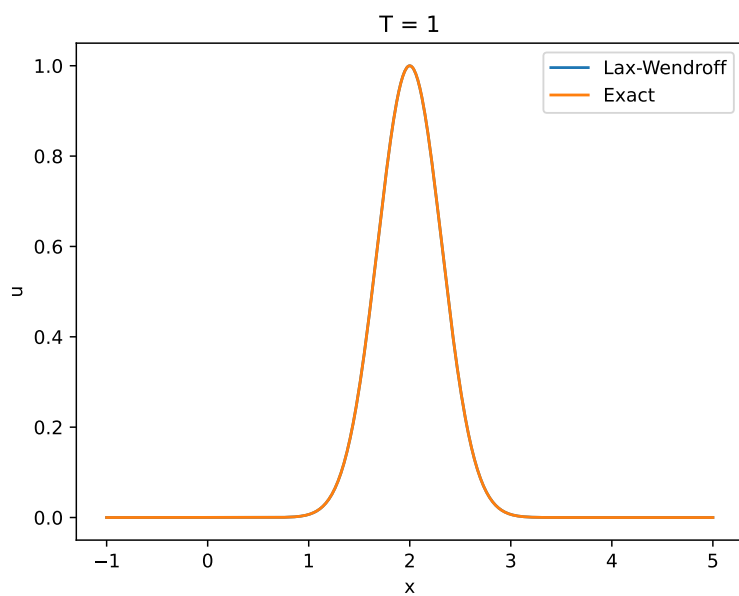


图 10: $T = 1.0$ 时使用 Lax-Wendroff 格式求解与精确解的对比

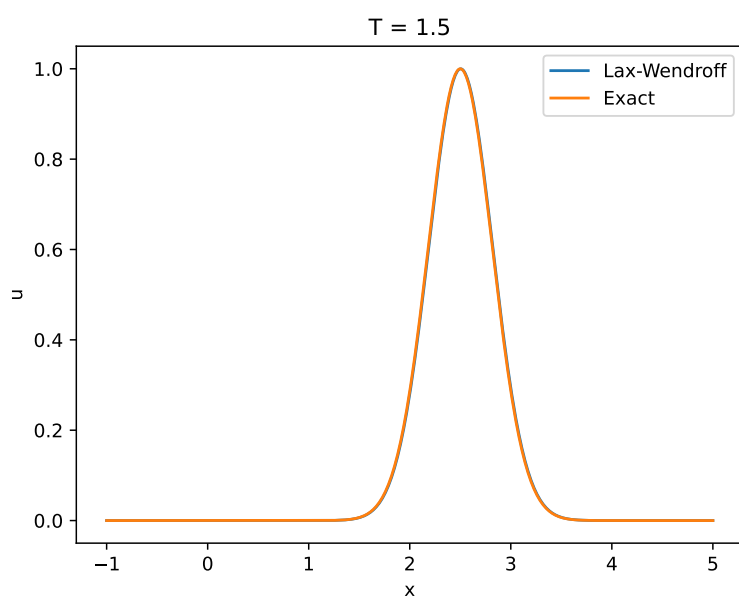


图 11: $T = 1.5$ 时使用 Lax-Wendroff 格式求解与精确解的对比

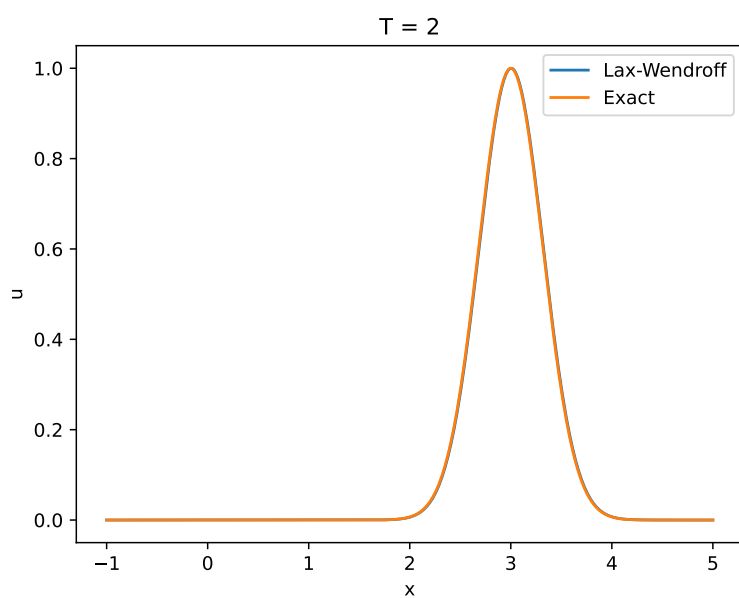


图 12: $T = 2.0$ 时使用 Lax-Wendroff 格式求解与精确解的对比