计算物理第八次作业

李柏轩 20300200004

December 13, 2022

1 题目 1：Poisson Equation

1.1 题目描述

Consider the Poisson equation

$$\nabla^2\phi(x,y) = -\frac{\rho(x,y)}{\epsilon_0} \tag{1}$$

from electrostatics on a rectangular geometry with $x \in [0, L_x]$ and $y \in [0, L_y]$. Write a program that solves this equation using the relaxation method. Test your program with:

(1) $\rho(x,y) = 0, \phi(0,y) = \phi(L_x,y) = \phi(x,0) = 0, \phi(x,L_y) = 1\text{V}, L_x = 1\text{m}, L_y = 1.5\text{m}$

(2) $\frac{\rho(x,y)}{\epsilon_0} = 1\text{V/m}^2, \phi(0,y) = \phi(L_x,y) = \phi(x,L_y) = \phi(x,0) = 0, L_x = L_y = 1\text{m}$

1.2 程序描述

本程序使用 Gauss-Seidel 方法求解二维静电势分布的 Poisson 方程（Eq.1）。在空间上进行离散化，得到离散的 Poisson 方程为

$$u_{i,j} = \frac{1}{4}(u_{i,j-1} + u_{i-1,j} + u_{i,j+1} + u_{i+1,j}) - \frac{h^2}{4}\rho_{i,j} \tag{2}$$

其中 h 为 x, y 方向实空间离散化的步长。Gauss-Seidel 方法每次迭代求解每个格点上取值的公式为

$$u_{i,j} = u_{i,j} + \frac{\omega}{4}(-\rho_{i,j}\cdot h^2 + u_{i,j-1} + u_{i-1,j} - 4u_{i,j} + u_{i,j+1} + u_{i+1,j}) \tag{3}$$

等号右边 u 的两个指标大于 i, j 的为上轮迭代的值，小于 i, j 的为本轮迭代的值。对 u 矩阵在相邻两轮迭代间作差并求 $l-2$ 范数，当其小于指定值时认为算法收敛到了 Poisson 方程的解。

本程序源文件为 Poisson.py，运行依赖 Python 第三方库 Numpy 和 Matplotlib。在终端进入当前目录，使用命令 python -u Poisson.py 运行本程序。运行后绘制出如 Fig.1所示的两种电荷分布 + 边界条件下 Poisson 方程的解。

1.3 伪代码

求解 Eq.1的 Gauss-Seidel 方法的伪代码如 Alg.1所示。

Algorithm 1 Gauss-Seidel Method

Input: Charge density distribution ρ, boundary condition $\phi(x,0), \phi(0,y), \phi(x,L_y), \phi(L_x,y)$, tolerable error ϵ, maximum iteration number M, real space discretization step h, and a initial guess u_0(with boundary condition satisfied).

Output: The solution of the 2-D Poisson equation.

$\text{norm}_\text{f} \leftarrow 0$
$\text{norm}_\text{l} \leftarrow ||u_0||$
iter_num $\leftarrow 0$
while iter_num $< M$ & $|\text{norm}_\text{l} - \text{norm}_\text{f}| > \epsilon$ **do**
  $\text{norm}_\text{f} \leftarrow \text{norm}_\text{l}$
  for $i \leftarrow 1, N_x - 1$ **do**
    for $j \leftarrow 1, N_x - 1$ **do**
      $u_{i,j} \leftarrow u_{i,j} + \frac{\omega}{4}(-\rho_{i,j} \cdot h^2 + u_{i,j-1} + u_{i-1,j} - 4u_{i,j} + u_{i,j+1} + u_{i+1,j})$
    end for
  end for
  $\text{norm}_\text{l} \leftarrow ||u||$
  iter_num $\leftarrow$ iter_num $+ 1$
end while
return u

1.4 输入输出示例

在 $h = 0.005, \epsilon = 1 \times 10^{-6}$ 时，程序输出两种电荷分布 + 边界条件下 Poisson 方程的解如 Fig.1所示。可见结果符合边界条件，符合定性预期，故知程序正确。

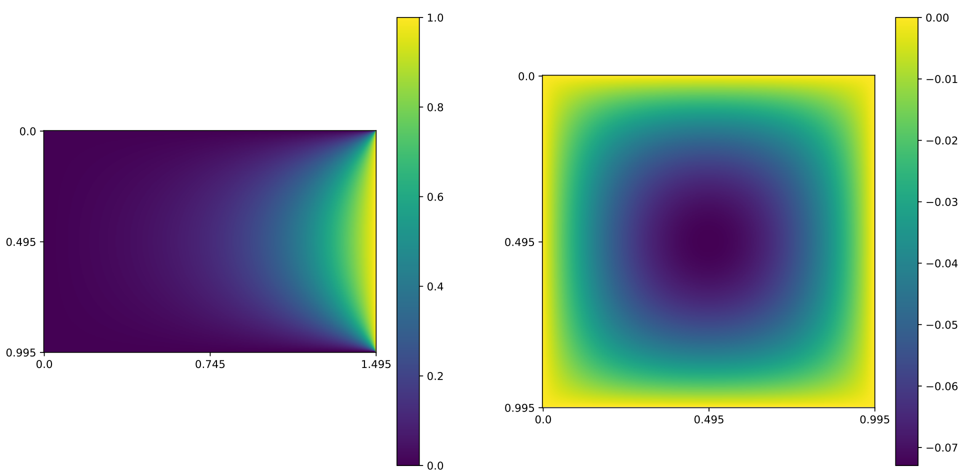

Figure 1: 两种电荷分布 + 边界条件下 Poisson 方程的解

2 题目 2：求解含时 Schrodinger 方程

2.1 题目描述

Solve the time-dependent Schrodinger equation using both the Crank–Nicolson scheme and stable explicit scheme. Consider the one-dimensional case and test it by applying it to the problem of a square well with a Gaussian initial state coming in from the left.

$$\begin{gathered}\psi(x,0)=\sqrt{\frac{1}{\pi}}e^{ik_0x-\frac{(x-\xi_0)^2}{2}}\\ V(x)=\begin{cases}+\infty, x<0 \text{ or } x>a_3\\ 0, x<a_1 \text{ or } x>a_2\\ -V_0, a_2<x<a_2\end{cases}\end{gathered}\tag{4}$$

2.2 程序描述

本程序分别通过 Crank Nicolson 方法和 Stable Explicit 方法求解含时 Schrodinger 方程。

$$i\hbar\frac{\partial\psi}{\partial t}=-\frac{\hbar^2}{2m}\frac{\partial^2\psi}{\partial x^2}+V\psi$$

Crank Nicolson 方法

$$\begin{gathered}\begin{pmatrix}1+\frac{i\Delta t\hbar}{2m\Delta x^2}+\frac{i\Delta tV_j}{2\hbar} & -\frac{i\Delta t\hbar}{4m\Delta x^2} & & & \\ -\frac{i\Delta t\hbar}{4m\Delta x^2} & 1+\frac{i\Delta t\hbar}{2m\Delta x^2}+\frac{i\Delta tV_j}{2\hbar} & -\frac{i\Delta t\hbar}{4m\Delta x^2} & & \\ & -\frac{i\Delta t\hbar}{4m\Delta x^2} & \ddots & \ddots & \\ & & \ddots & & -\frac{i\Delta t\hbar}{4m\Delta x^2}\\ & & & -\frac{i\Delta t\hbar}{4m\Delta x^2} & 1+\frac{i\Delta t\hbar}{2m\Delta x^2}+\frac{i\Delta tV_j}{2\hbar}\end{pmatrix}\begin{pmatrix}\psi_0^{n+1}\\ \psi_1^{n+1}\\ \vdots\\ \psi_{N_x-1}^{n+1}\\ \psi_{N_x}^{n+1}\end{pmatrix}\\ =\begin{pmatrix}1-\frac{i\Delta t\hbar}{2m\Delta x^2}-\frac{i\Delta tV_j}{2\hbar} & \frac{i\Delta t\hbar}{4m\Delta x^2} & & & \\ \frac{i\Delta t\hbar}{4m\Delta x^2} & 1-\frac{i\Delta t\hbar}{2m\Delta x^2}-\frac{i\Delta tV_j}{2\hbar} & \frac{i\Delta t\hbar}{4m\Delta x^2} & & \\ & \frac{i\Delta t\hbar}{4m\Delta x^2} & \ddots & \ddots & \\ & & \ddots & & \frac{i\Delta t\hbar}{4m\Delta x^2}\\ & & & \frac{i\Delta t\hbar}{4m\Delta x^2} & 1-\frac{i\Delta t\hbar}{2m\Delta x^2}-\frac{i\Delta tV_j}{2\hbar}\end{pmatrix}\begin{pmatrix}\psi_0^{n}\\ \psi_1^{n}\\ \vdots\\ \psi_{N_x-1}^{n}\\ \psi_{N_x}^{n}\end{pmatrix}\end{gathered}\tag{5}$$

Stable Explicit 方法：

$$\begin{pmatrix}\psi_0^{n+1}\\ \psi_1^{n+1}\\ \vdots\\ \psi_{N_x-1}^{n+1}\\ \psi_{N_x}^{n+1}\end{pmatrix}=\begin{pmatrix}\psi_0^{n-1}\\ \psi_1^{n-1}\\ \vdots\\ \psi_{N_x-1}^{n-1}\\ \psi_{N_x}^{n-1}\end{pmatrix}+\begin{pmatrix}-\frac{2i\Delta t\hbar}{m\Delta x^2}-\frac{2i\Delta tV_j}{\hbar} & \frac{i\Delta t\hbar}{m\Delta x^2} & & & \\ \frac{i\Delta t\hbar}{m\Delta x^2} & -\frac{2i\Delta t\hbar}{m\Delta x^2}-\frac{2i\Delta tV_j}{\hbar} & \frac{i\Delta t\hbar}{m\Delta x^2} & & \\ & \frac{i\Delta t\hbar}{m\Delta x^2} & \ddots & \ddots & \\ & & \ddots & & \frac{i\Delta t\hbar}{m\Delta x^2}\\ & & & \frac{i\Delta t\hbar}{m\Delta x^2} & -\frac{2i\Delta t\hbar}{m\Delta x^2}-\frac{2i\Delta tV_j}{\hbar}\end{pmatrix}\begin{pmatrix}\psi_0^{n}\\ \psi_1^{n}\\ \vdots\\ \psi_{N_x-1}^{n}\\ \psi_{N_x}^{n}\end{pmatrix}\tag{6}$$

本程序源文件为 Schrodinger.py，运行依赖 Python 第三方库 Numpy、math 和 Matplotlib。在终端进入当前目录，使用命令 python -u Schrodinger.py 运行本程序。运行后输出 Fig.2所示图像

2.3 伪代码

Crank Nicolson 方法的伪代码如 Alg.2所示，Stable Explicit Scheme 的伪代码如 Alg.3所示，

Algorithm 2 Crank Nicolson method

Input: The initial wavefunction ψ_0, the time step Δt, the time step number N_t

Output: The time evolution of ψ

for $n \leftarrow 1, 2, \cdots, N_t$ **do**

  $\psi_n \leftarrow (1 + \frac{i\Delta t\hat{H}}{2\hbar})^{-1}(1 - \frac{i\Delta t\hat{H}}{2\hbar})\psi_{n-1}$

  $\psi_n(0), \psi_n(a) \leftarrow 0$

end for

Algorithm 3 Stable Explicit Scheme

Input: The initial wavefunction ψ_0, the time step Δt, the time step number N_t

Output: The time evolution of ψ

for $n \leftarrow 1, 2, \cdots, N_t$ **do**

  $\psi_n \leftarrow \psi_{n-2} - 2i\hat{H}\Delta t\psi_{n-1}$

  $\psi_n(0), \psi_n(a) \leftarrow 0$

end for

2.4 输入输出示例

程序中令 $m = \hbar = 1, V_0 = 10, a_1 = 75, a_2 = 125, a_3 = 200, k = 5, \xi_0 = 40$，两种方法的结果分别如 Fig.2所示。可以看到初始高斯波包在遇到势阱时分成反射和透射两部分，符合定性预期，两种方法结果也大致相符。

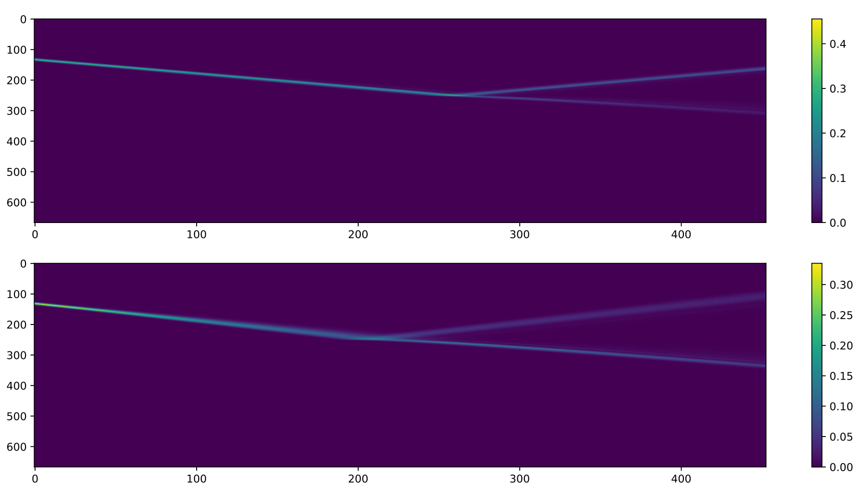

Figure 2: Crank Nicolson 方法（上）和 Stable Explicit Scheme（下）的计算结果

3 题目 3：稳定条件证明

3.1 题目描述

Prove the stability condition($\dfrac{c\Delta t}{\Delta x}$) of the explicit scheme of the 1D wave equation by performing Von Neumann stability analysis.

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \tag{7}$$

3.2 证明过程

$$\begin{aligned}
&\left.\frac{\partial^2 u}{\partial t^2}\right|_{ij} = \frac{1}{\Delta t^2}(u_{i,j+1} + u_{i,j-1} - 2u_{i,j}) \\
&\left.\frac{\partial^2 u}{\partial x^2}\right|_{ij} = \frac{1}{\Delta x^2}(u_{i+1,j} + u_{i-1,j} - 2u_{i,j}) \\
&u_{i,j+1} + u_{i,j-1} - 2u_{i,j} = \frac{c^2 \Delta t^2}{\Delta x^2}(u_{i+1,j} + u_{i-1,j} - 2u_{i,j}) = \alpha^2(u_{i+1,j} + u_{i-1,j} - 2u_{i,j}) \\
&\frac{(\xi - 1)^2}{\xi} = \alpha^2 \frac{(e^{IK\Delta x} - 1)^2}{e^{IK\Delta x}} = 4\alpha^2 \sin \frac{1}{2} K \Delta x
\end{aligned} \tag{8}$$

当 $\alpha \leq 1$ 时，$|\xi| < 1$，所以稳定。