第八次作业

尹朝阳 物理学系

1 题目 1

1.1 题目描述

Consider the Poisson equation

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

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) $\rho(x,y) = 0, \phi(0,y) = \phi(L_x,y) = \phi(x,0) = 0, \phi(x,L_y) = 1V$, $L_x = 1m$, and $L_y = 1.5m$;

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

1.2 程序描述

本题目要求用 relaxation method 进行 Poisson 方程的求解。这里选取 Gauss-Seidel Algorithm 进行求解。

首先，对于二维的直角坐标系的情况，Poisson 方程形式为

$$\nabla^2\phi(x,y) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -\frac{\rho(x,y)}{\epsilon_0}$$

可以采用有限差分的方法将其离散化：

$$u_{xx} = \frac{u(x_i+\Delta x, y_j) - 2u(x_i,y_j) + u(x_i-\Delta x, y_j)}{\Delta x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}$$

$$u_{yy} = \frac{u(x_i, y_j+\Delta y) - 2u(x_i,y_j) + u(x_i, y_j-\Delta y)}{\Delta y^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2}$$

若以正方形网格进行划分，记 $\Delta x = \Delta y = h$，则原式变为

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

其中已经将 ϵ 吸收到 ρ 中，即此处的 ρ 表示 $\frac{\rho}{\epsilon}$ 的值。

假设网格划分为 $n \times m$ 个格点，则除去边界上的点受边界条件的约束，还一共有 $(n-2)(m-2)$ 个未知数，可以列出 $(n-2)(m-2)$ 个方程，则问题转化为解线性方程组

$$Ax = b$$

其中 A 为系数矩阵，b 为边界条件组成的已知向量。

Gauss-Seidel 算法的根本思路是先设置估计的初值，然后用相邻的格点对每个个点进行更新。其迭代方式为

$$y_j^{(i+1)}=\frac{1}{A_{jj}}(b_j-\sum_{k=1}^{j-1}A_{jk}y_k^{(i)}-\sum_{k=j+1}^{N}A_{jk}y_k^{(i)})$$

第一种方法是用矩阵形式每一轮对所有格点整体进行更新。记

$$A=D+L+U$$

其中 D 为对角矩阵，L 为严格下三角矩阵，U 为严格下三角矩阵。则迭代解为

$$y^{(i+1)}=D^{-1}b-D^{-1}(L+U)y^{(i)}$$

可以通过初值进行迭代最终找到真正的解。

但是，系数矩阵 A 和向量 b 受划分方式的影响较大，例如，对于一个正方形区域如果采用 4×4 分划，则向量 x 对应的矩阵等式为

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix}\begin{bmatrix} u_{11} \\ u_{12} \\ u_{21} \\ u_{22} \end{bmatrix}=\begin{bmatrix} u_{01}+u_{10}+h^2\rho_{11} \\ u_{13}+u_{02}+h^2\rho_{12} \\ u_{31}+u_{20}+h^2\rho_{21} \\ u_{32}+u_{23}+h^2\rho_{22} \end{bmatrix}$$

而如果采用 5×5 分划，则矩阵等式又变成

$$\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix}\begin{bmatrix} u_{11} \\ u_{21} \\ u_{31} \\ u_{12} \\ u_{22} \\ u_{32} \\ u_{13} \\ u_{23} \\ u_{33} \end{bmatrix}=\begin{bmatrix} u_{01}+u_{10}+h^2\rho_{11} \\ u_{20}+h^2\rho_{21} \\ u_{41}+u_{30}+h^2\rho_{31} \\ u_{02}+h^2\rho_{12} \\ h^2\rho_{22} \\ u_{42}+h^2\rho_{32} \\ u_{03}+u_{14}+h^2\rho_{13} \\ u_{24}+h^2\rho_{23} \\ u_{43}+u_{34}+h^2\rho_{33} \end{bmatrix}$$

可以发现，由于分划方式的不同，系数矩阵 A 和向量 b 变化很大，程序没有较好的可移植性。但是也可以发现，不同的分划方式对应的系数矩阵 A 都是一个主元较大的稀疏矩阵，满足 $|a_{ii}|>\sum\limits_{j=1,j\neq i}^{n}|a_{ij}|$，则 Poisson 方程采用 relaxation 方法不会导致不收敛的问题。

故这里采用第二种方法，即每一轮逐点进行迭代的方法。即对每个点有如下迭代公式：

$$u_{ij}=\frac{1}{4}(u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1})$$

迭代轮次由 $maxIter$ 和 $diff=\sum|u_{ij}-uTemp_{ij}|/(n*m)$ 共同决定。其中 $uTemp$ 为没有开始迭代这一轮时的 u 值。

与 Jacobi 算法相比，在 $n+1$ 轮迭代中，部分用到了之前在 $n+1$ 轮已经完成迭代的值，而非只在右边代入第 n 轮的相邻值。

本题目的源程序位于 PoissonEquation 文件夹中，question1.py 和 question2.py 分别是两问的求解，导入的第三方库有 numpy、matplotlib 和 mpl_toolkits.mplot3d。

1.3 伪代码

由于第一问和第二问只是边界条件不同以及右边的源不同（第一问虽然 L_x、L_y 不相等，但是在逐点迭代的方式下没有影响，仍然可以用正方形网格进行划分），这里只给出通用的求解 Poisson 方程的伪代码。

Algorithm 1 Gauss-Seidel iteration method

set ρ_{ij}, u_{ij} and the boundary condition $u[:,0], u[:,-1], u[0,:], u[-1,:]$ ▷ Initialization
set $maxIter, \epsilon$ ▷ the maximum number of iteration and the error of final solution
$iter \leftarrow 0$; $diff \leftarrow 1$
while $iter < maxIter$ and $diff \geq \epsilon$ **do**
  $diff \leftarrow 0$
  $uTemp \leftarrow u$
  for $i \leftarrow 2$ to $m-1$ **do** ▷ m as the row
    for $j \leftarrow 2$ to $n-1$ **do** ▷ n as the column
      $u_{ij} \leftarrow 0.25 * (u_{i-1,j}, u_{i+1,j}, ui, j-1, u_{i,j+1} + h^2\rho_{ij})$ ▷ h as the step
      $diff \leftarrow diff + |uTemp_{ij} - u_{ij}|$
    end for
  end for
  $iter \leftarrow iter + 1$; $diff \leftarrow diff/(n*m)$
end while
return u

1.4 测试用例

两问的结果如下图 1、2 所示。

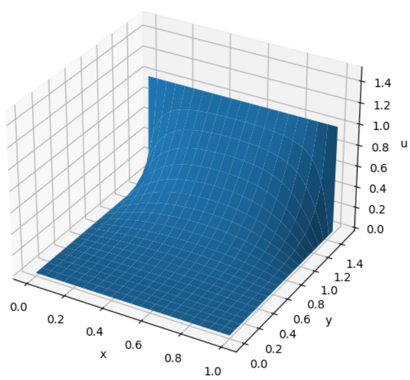

图 1: laplace_equation

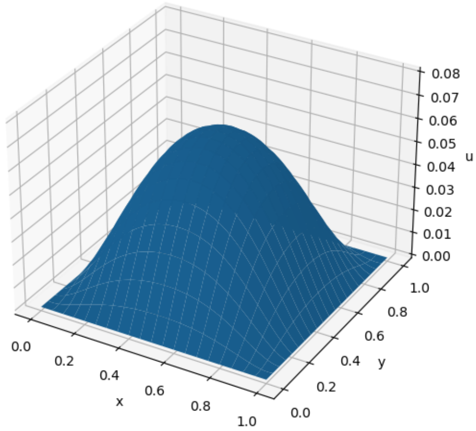

图 2: poisson_equation

2 题目 2

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.

2.2 程序描述

本题目要求用 Crank-Kicolson 方法和 stable explicit 方法求解一维的含时薛定谔方程。

首先，为了求解简便，认为方势阱较深，高斯行波只能在势阱内运动，而无法进行隧穿，跑到势阱外。则可以设置边界条件 $\psi(0,t)=0$ 和 $\psi(L,t)=0$。

1. Crank-Nicolson Scheme

对于一维含时薛定谔方程

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

对 x 进行有限差分：$x_j=j\Delta x$，其中 $j=1,2,...,L$。

记算符 H_D 满足

$$H_D\psi_j=-\frac{\hbar^2}{2m}\frac{1}{\Delta x^2}(\psi_{j-1}+\psi_{j+1}-2\psi_j)+V_j\psi_j$$

显然，H_D 写为矩阵的形式即为

$$H_D = -\frac{\hbar^2}{2m}\frac{1}{\Delta x^2} * \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{bmatrix} + \begin{bmatrix} V_1 & & & & \\ & V_2 & & & \\ & & V_3 & & \\ & & & \ddots & \\ & & & & V_n \end{bmatrix}$$

对时间 t 进行有限差分：$t_n = n\Delta t$。

则薛定谔方程可化为

$$i\hbar\frac{1}{\Delta t}(\psi_j^{n+1} - \psi_j^n) = H_D\frac{\psi_j^n + \psi_j^{n+1}}{2}$$

整理得

$$\psi_j^{n+1} = (1 + \frac{i\Delta t H_D}{2\hbar})^{-1}(1 - \frac{i\Delta t H_D}{2\hbar})\psi_j^n$$

即得到迭代公式。

2. Stable Explicit scheme

为简便，不妨设 $\hbar = 1, m = 1$，则薛定谔方程变为

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

分离变量得时间方程为

$$U(t) = e^{-iHt}$$

则有

$$\psi^{n+1} = U(\Delta t)\psi^n, \qquad \psi^{n-1} = U(-\Delta t)\psi^n$$

将 $U(\Delta t)$ 展开到一阶，$U(\Delta t) = e^{-iH\Delta t} = 1 - iH\Delta t$，将 $H\psi_j$ 用 $H\psi_j = -\frac{1}{2}\frac{1}{\Delta x^2}(\psi_{j+1} + \psi_{j-1} - 2\psi_j) + V_j\psi_j$ 代替，最终可以整理成以下的迭代公式：

$$\psi_j^{n+1} = \psi_j^{n-1} + \frac{i\Delta t}{(\Delta x)^2}(\psi_{j+1}^n + \psi_{j-1}^n - 2\psi_j^n) - 2i\Delta t V_j\psi_j^n$$

由于该递推公式为二阶，还需要人为设定从初始 0 时刻增加 Δt 时间后的波包，并赋值给 Δt 时刻的波函数。由于初始时设置高斯波包位于方势阱内，且没有与势阱壁产生作用，故设置高斯波包的形状、大小不变，仅改变它的位置。

本题目的源程序位于 SchrodingerEquation 文件夹中，CrankNicolson_scheme.py 和 stable_explicit_scheme.py 分别为两种方法求解。导入的第三方库有 numpy、scipy 和 matplotlib。

Algorithm 2.1 Crank-Nicolson scheme

set the sphere of x, the span of t, the step Δx and Δt
for $j \leftarrow 1$ to l **do**
    $\psi_{1j} \leftarrow \Psi(x_j, 0)$ ▷ Initialization
end for
set the potential V
for $m \leftarrow 1$ to l **do**
    for $k \leftarrow 1$ to l **do**
        if $m = k$ **then**
            $H_{D_{mk}} \leftarrow \frac{1}{\Delta x^2} + V_m$
        else if $m = k + 1$ or $m = k - 1$ **then**
            $H_{D_{mk}} \leftarrow -\frac{1}{2\Delta x^2}$
        else
            $H_{D_{mk}} \leftarrow 0$
        end if
    end for
end for
for $i \leftarrow 1$ to $n - 1$ **do**
    for $j \leftarrow 1$ to l **do**
        $\psi_{i+1,j} = (1 + \frac{I\Delta t H_D}{2\hbar})^{-1}(1 - \frac{I\Delta t H_D}{2\hbar})\psi_{ij}$ ▷ I as the imaginary unit
    end for
end for
return ψ

Algorithm 2.2 stable explicit scheme

set the sphere of x, the span of t, the step Δx and Δt
for $j \leftarrow 1$ to l **do** ▷ Initialization
    $\psi_{1j} \leftarrow \Psi(x_j, 0)$
    $\psi_{2j} \leftarrow \Psi(x_j, \Delta t)$
end for
set the potential V
for $i \leftarrow 2$ to n **do**
    for $j \leftarrow 2$ to $l - 1$ **do**
        $\psi_{i+1,j} \leftarrow \psi_{i-1,j} + I\Delta t[(\psi_{i,j-1} + \psi_{i,j+1} - 2\psi_{i,j})/(\Delta x)^2 - 2V_j\psi_{i,j}]$ ▷ I as the imaginary unit
    end for
end for
return ψ

2.3 伪代码

2.4 测试用例

由于采用 Crank-Nicolson scheme 和 stable explicit scheme 解得的结果基本一致，只对其中一个结果进行分析（这里选择 Crank-Nicolson scheme 进行分析）。首先，由于体系（方势阱）的偶宇称性，行波遇到左边势垒与右边势垒的情形是一致的，故这里我们截取时间段为高斯波包开始向 x 轴正方向行进，遇到势垒被反射的最初一段过程进行分析。该时间段的高斯波包的行进如下图 3 所示。其中九张图分别为 $t = 0, 4, 12, 15, 19, 22, 28, 30, 36$ 时刻的情形。

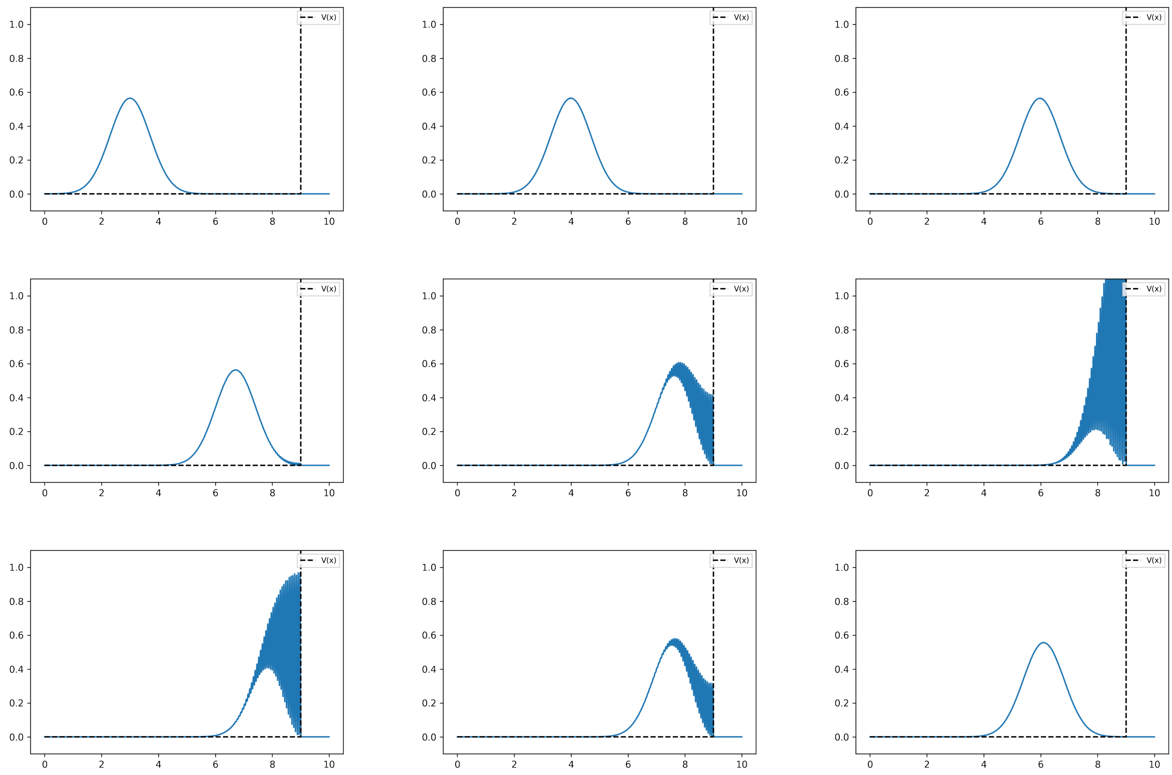

图 3: Gaussian wave in the square wall

可以看到，在遇到右壁势垒之前，由于势函数为 0 是常数，高斯波包不产生变化，以平面波的形式向右运动。当遇到势垒时，其前半部分产生反射，后半部分继续向前行进，实虚部分别发生干涉（这里由于参数设置原因干涉相长和相消在图上体现得不是很明显），直至整个波形与势垒发生作用后，被完全反射，以原来的高斯波包的形式向左行进，直到撞上左边的势垒，再次被反射。当然，这里由于选取的是无限深方势阱（真实设置时势函数取值使隧穿部分极快速衰减，可以忽略不计），故高斯波包只产生反射，而没有透射部分。若势阱高度不够甚至小于峰高，此时高斯波包会有一部分透射出去，从而反射波与入射波不完全一致。此时只需要选取边界包含两个势阱的部分（如 $-3L$ 到 $4L$），在这两端设置边界条件即可。

3 题目 3

3.1 题目描述

Prove the stability condition of the explicit scheme of the 1D wave equation by performing Von Neumann stability analysis

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

If $\frac{c\Delta t}{\Delta x} \le 1$, the explicit scheme is stable.

3.2 题目解答

首先用有限差分将变量 x 和 t 进行划分，则有 $x_i = i\Delta x, t_j = j\Delta t$，则有

$$u_{xx} = \frac{u(x_i+\Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j)}{\Delta x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}$$

$$u_{tt} = \frac{u(x_i, t_j+\Delta t) - 2u(x_i, t_j) + u(x_i, t_j - \Delta t)}{\Delta t^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta t^2}$$

代回原方程，并令 $\alpha = \frac{c\Delta t}{\Delta x}$，可得

$$u_{i,j+1} - 2u_{i,j} + u_{i,j-1} = \alpha^2(u_{i+1,j} - 2u_{i,j} + u_{i-1,j})$$

代入试解 $u_{i,j} = \xi^j e^{IKi\Delta x}$，其中 $I = \sqrt{-1}$，整理得到

$$\xi^2 - 2(1-\alpha^2)\xi + 1 - 2\alpha^2\xi\cos(K\Delta x) = 0$$

令 $\beta = 1 - 2\alpha^2\sin^2(\frac{K\Delta x}{2})$，上式化简为

$$\xi^2 - 2\beta\xi + 1 = 0$$

解得

$$\xi_{1,2} = \beta \pm \sqrt{\beta^2 - 1}$$

注意到，如果 $\beta > 1$，则 ξ_1 与 ξ_2 中至少有一个会大于 1，这会导致迭代解不收敛，即解不稳定。所以 β 必须满足 $|\beta| \le 1$，则有

$$\xi_{1,2} = \beta \pm I\sqrt{1-\beta^2}$$

此时满足 $|\xi| = \sqrt{\beta^2 + 1 - \beta^2} = 1$，此时解不发散。

$|\beta| \le 1$ 代回 α 的表达式，有

$$-1 \le 1 - 2\alpha^2\sin^2(\frac{K\Delta x}{2}) \le 1$$

即

$$0 \le \alpha^2\sin^2(\frac{K\Delta x}{2}) \le 1$$

上式对任何分划 Δx 成立，则要求

$$\alpha = \frac{c\Delta t}{\Delta x} \le 1$$

证毕。