

IYPT 第 16 题——气垫大炮

郑壹夫

1 问题回顾

Place an object on a large air cushion and drop several other objects in such a way that the first object is catapulted away. Investigate how the **exit velocity** depends on **relevant parameters**.

将一个物体放在一个大的气垫上，然后以同样的方式把其他几个物体扔下去，这样第一个物体就会弹射出去。研究出射速度与相关参数的关系。

2 参数与符号

| 符号 | 意义 | 单位 |
|-----------|--|--------------|
| h | 下落高度 h | m |
| l | 两物体的间距 l | m |
| m_1 | 下落物体质量 m_1 | kg |
| v_1 | 撞击速度 v_1 | m/s |
| m_2 | 弹射物体质量 m_2 | kg |
| v_2 | 出射速度 v_2 | m/s |
| m_{air} | 气体质量 m_{air} | kg |
| v_{air} | 撞击后的气体速度 v_{air} | m/s |
| E_{air} | 撞击后气体包含的重力势能与动能 E_{air} | J |
| E_d | 撞击过程中耗散的能量 $E_{dissipate}$ | J |
| E_{re} | 空气传播过程中耗散的能量 $E_{resistence}$ | J |
| U | 流体的速度场 U | m/s |
| p | 流体的气压场 p | Pa |
| μ | 动力学粘性系数 μ | $Pa \cdot s$ |
| u | x 轴方向上的速度 u | m/s |
| v | y 轴方向上的速度 v | m/s |
| ∇ | 梯度操作 $\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$ | — |
| ∇ | 散度操作 $\nabla \cdot \mathbf{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y}$ | — |
| Δ | 拉普拉斯算子（梯度的散度） $\Delta f = \nabla \cdot \nabla f$ | — |

表 1: 使用的参数与符号

3 主体思路

根据能量守恒定律，我们可以得到

$$m_1gh = \frac{1}{2}m_2v_2^2 + E_{air} + E_d + E_{re} \quad (1)$$

根据动量守恒定律

$$m_1v_1 = m_{air}v_{air} \quad (2)$$

$$m'_{air}v'_{air} = m_2v_2 \quad (3)$$

其中对于 m_{air} 与 v_{air} 的计算需要考虑到空气对动量与能量的复杂传递过程与气体运动过程。对此我们尝试对气垫中的空气进行流体力学分析。

4 长方形二维方腔区域的流体力学模型

考虑到气垫受气体的流动影响会产生无法描述的形变，进而难以对气垫内空气的流动过程进行系统化与公式化描述，我们尝试简化气垫的物理特征，将其视作刚体结构，并对气垫的二维剖面进行计算流体力学计算。

4.1 模型假设

首先，我们需要进行模型假设。

4.1.1 假设 1

小球垂直下落，对气垫施加的力 F 。其具有瞬时性和局部性，即这个力只在一个特定的瞬间和位置施加。

理由：考虑到下落物体与气垫的碰撞过程较为短暂，且气体对碰撞作用的响应时间较短，我们在进行二维方腔中的计算流体力学计算时将碰撞过程简化为短暂施力过程。

4.1.2 假设 2

在低速流动环境中，空气可以视作是不可压缩流体。

理由：该假设基于物理现实。考虑到空气在低速流动下的可压缩性远小于其在高速流动下的可压缩性，且该实验中不设计高强度的压缩碰撞，我们将空气视作不可压缩流体。

4.2 模型简化

模型的简化基于假设，此时我们尝试搭建更加理想化的模型。

4.2.1 简化 1

气垫是一个封闭的刚性矩形空间。

理由：正如之前所言，考虑完全具备气垫的物理性质尤其是弹性的矩形空间将会使得计算流体力学的计算过程变得极度复杂，可能涉及到混沌系统。

4.2.2 简化 2

在此矩形空间点 1，有一已知的加速度 a_1 。

理由：我们将瞬时的力更加简化为气体的加速度，这是因为计算流体力学中极少考虑力场，而是更多考虑速度场与压力场。

4.2.3 简化 3

在点 2 处，存在一个由于空气流体的传导而产生的加速度 a_2 。

理由：同上，我们尝试将模型简化为仅仅对速度场与压力场的考量。

4.2.4 简化 4

空气仅在矩形空间的二维平面内移动。（在这里我们考虑处在中间位置的二维矩形平面）

理由：考虑到矩形的对称性，我们倾向于认为与二维平面相交的速度与压力可以被分解为互相抵消的一对平行分量与互相叠加的一对垂直分量，故我们仅仅对二维矩形平面进行分析。

4.3 计算流体力学的理论

对于流体流动行为的描述，常采用纳维斯托克斯方程 (Navier-Stokes 方程，简称 NS 方程)，并试图寻找对其有效的数值求解方法。因为 NS 方程包含了两个未知数——速度场 U 和压力场 p ——以及两个方程所构成的方程组，理论上可以模拟和解决纳维斯托克斯方程的时间演化，从而求得某一时刻流体的气压场以及与气压场耦合的速度场。故需要尝试对纳维斯托克斯方程在特定区域进行时间离散与空间离散，将时间连续域分解成一系列离散的时间步长，在每一个时间步长上对空间离散后由偏微分方程转换得到的线性的方程进行计算机求解。时间离散方法包括显式方法（例如向前欧拉方法）、隐式方法（例如后向欧拉方法与 Crank-Nicolson 方法）与半隐式方法。空间离散方法包括有限差分法、有限元法与有限体积法。其中有限差分法尝试使用差分公式在离散网格上近似导数，主要有交错网格法、中心差分格式法、上风差分格式法与其他高阶差分格式方法。

4.4 显式时间离散法与有限差分法

4.4.1 对纳维斯托克斯方程的时间离散

NS 方程可以表示为以下方程组：

$$\begin{cases} \nabla \cdot U = 0 \\ \frac{\partial U}{\partial t} + U \cdot (\nabla U) = -\frac{1}{\rho} \nabla p + \nu \nabla^2 U \end{cases} \quad (4)$$

其中 $\nabla \cdot U$ 是速度场 U 的散度； $\frac{\partial U}{\partial t}$ 是速度 U 随时间 t 的变化率； $U \cdot (\nabla U)$ 是对流项，描述了速度场 U 的自我平流效应； $-\frac{1}{\rho} \nabla p$ 是由于压力梯度产生的加速度项，其中 ρ 是流体的密度， ∇p 是压力的梯度； $\nu \nabla^2 U$ 是粘性项，其中 ν 是流体的动力学粘性系数， $\nabla^2 U$ 是速度场 U 的拉普拉斯算子。

显然流体的连续性方程（方程 14）本质上只是对速度场的约束条件，并不包含压力。若给动量守恒方程（方程 15）输入一个压力场，那么通过方程 15 求得的速度场 U 将不一定满足于连续性方程所规定的速度散度为 0 这一约束。因此在速度场的时间推进过程中，应引入对速度场的矫正量 U^* ，从而使得速度场在时间推进过程中始终满足连续性方程的约束条件。

对于方程 5 中的时间项 $\frac{\partial U}{\partial t}$ ，采用时间向前的差分格式 (Forward Time Difference Scheme)，尝试对时间偏导数进行近似表示，并考虑对速度场矫正量 U^* 的引入，可以将方程 5 写作：

$$\frac{\partial U}{\partial t} \rightarrow \frac{U^{n+1} + U^* - U^* - U^n}{\Delta t} \quad (6)$$

这即是时间偏导数的时间离散形式。

对于压力梯度项 $-\frac{1}{\rho}\nabla p$ ，因为压力是一个强烈的驱动流动的因子，且压力场与速度场之间存在复杂的耦合关系，显性方法对于压力梯度项可能会导致时间步长受到严格的限制（为了数值稳定性），从而需要更小的时间步长，导致计算效率低下。故采用隐式离散法，使用未来时间步的值来表示压力的变化。压力梯度项的时间离散形式为：

$$-\frac{1}{\rho}\nabla p \rightarrow -\frac{1}{\rho}\nabla p^{n+1} \quad (7)$$

对于与速度有关的对流项 $U \cdot (\nabla U)$ 和剪切应力项 $v\nabla^2 U$ [又称粘性项，因为粘性流体的内部摩擦主要是由流体层之间的剪切产生。流体内部的这种摩擦阻力导致了剪切应力的产生。用数学语言描述，剪切应力与速度场的梯度（速度的空间变化率）成正比。在牛顿型流体中，这种比例关系由动力学粘性系数 ν 决定]，因为其通常具有非线性特性，直接采用隐式方法会导致每个时间步都需要解决复杂的非线性方程组，这在计算上可能非常昂贵。故采用显性离散法，在离散方程中采用当前时刻的值以使得每个时间步在计算上相对高效。得到：

$$U \cdot (\nabla U) \rightarrow U^n \cdot (\nabla U^n) \quad (8)$$

$$v\nabla^2 U \rightarrow v\nabla^2 U^n \quad (9)$$

整理方程 6-9，将对流项移动到等式右侧，得到动量守恒方程的时间离散方程：

$$\frac{U^{n+1} + U^* - U^* - U^n}{\Delta t} = -\frac{1}{\rho}\nabla p^{n+1} - U^n \cdot (\nabla U^n) + v\nabla^2 U^n \quad (10)$$

将方程 10 拆分为 $n+1$ 时刻的隐式部分与 n 时刻的显性部分，得到以下方程组：

$$\begin{cases} \frac{U^{n+1} - U^*}{\Delta t} = -\frac{1}{\rho}\nabla p^{n+1} \end{cases} \quad (11)$$

$$\begin{cases} \frac{U^* - U^n}{\Delta t} = -U^n \cdot (\nabla U^n) + v\nabla^2 U^n \end{cases} \quad (12)$$

对于方程 12，可以得到计算速度矫正量 U^* 的计算方法：

$$U^* = U^n + \Delta t (v\nabla^2 U^n - U^n \cdot (\nabla U^n)) \quad (13)$$

对于方程 11，通过变换并对等式左右两侧同时求散度，将梯度算符变为拉普拉斯算符，可以得到：

$$\nabla \cdot U^{n+1} - \nabla \cdot U^* = -\frac{\Delta t}{\rho}\nabla^2 p^{n+1} \quad (14)$$

在方程 14 中，对于经过矫正的速度场 U^{n+1} ，应符合 NS 方程中 $\nabla \cdot U = 0$ （方程 4）这一约束。通过对方程 14 的变换可以得到压力泊松方程：

$$\nabla^2 p^{n+1} = \frac{\rho}{\Delta t}\nabla \cdot U^* \quad (15)$$

同时也可以从方程 11 中得到下一时刻速度场 U^{n+1} 的计算方法，即：

$$U^{n+1} = U^* - \frac{\Delta t}{\rho}\nabla p^{n+1} \quad (16)$$

综合方程 13-16，求解纳维斯托克斯方程速度压力耦合场的方程经时间离散后转化为了以下方程组：

$$\begin{cases} U^* = U^n + \Delta t (v \nabla^2 U^n - U^n \cdot (\nabla U^n)) & (17) \\ \nabla^2 p^{n+1} = \frac{\rho}{\Delta t} \nabla \cdot U^* & (18) \\ U^{n+1} = U^* - \frac{\Delta t}{\rho} \nabla p^{n+1} & (19) \end{cases}$$

对于该方程组，首先需要利用方程 17 通过当前的速度场 U 求得速度场矫正量 U^* ，接着利用压力泊松方程（方程 18）通过速度场矫正量 U^* 求得下一时刻的压力场 p^{n+1} ，最后利用方程 19 通过速度场矫正量 U^* 与下一时刻的压力场 p^{n+1} 求得下一时刻的速度场，从而完成速度场从 n 时刻向 $n+1$ 时刻的推进。

4.4.2 对纳维斯托克斯方程的空间离散

我们尝试通过某种差分格式表达 U^* ， p^{n+1} 与 U^{n+1} 在二维方腔内的一系列空间节点上的数值计算方法。要对这些空间节点上的速度压力耦合场进行编程数值求解，首先需要沿长方形的底边与左侧边建立坐标系，并尝试将矢量形式的方程表示成坐标轴方向上的分量形式。

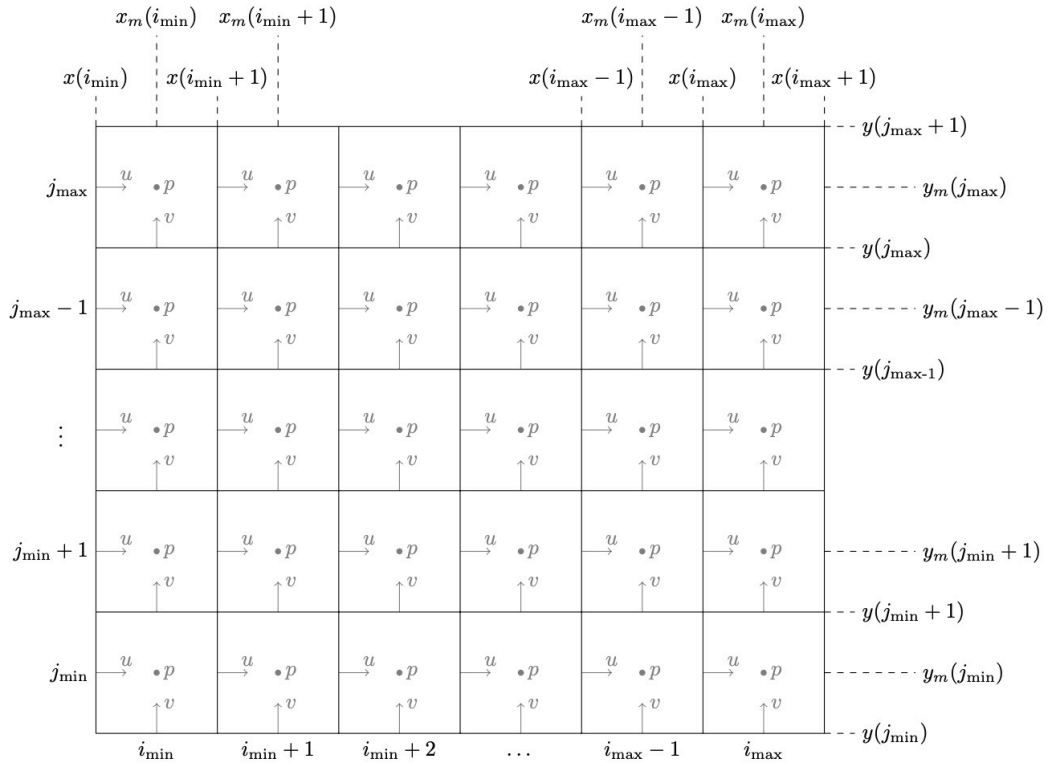


Figure. 1. 交错网格

对速度场矫正量方程的空间离散

首先将速度场矫正量方程（方程 17）由矢量形式转化为分量形式。将方程中的剪切应力项 $v \nabla^2 U^n$ 改写成先对速度求梯度，再求散度的形式，即 $v \nabla \cdot (\nabla U^n)$ 。故速度场矫正量方程变为：

$$U^* = U^n + \Delta t (v \nabla \cdot (\nabla U^n) - U^n \cdot (\nabla U^n)) \quad (20)$$

对于牵扯到矢量的剪切应力项 $v \nabla \cdot (\nabla U^n)$ ，计算括号内梯度算符与速度的外积计算并接着进行散度算符与该方阵的内积计算，得到：

$$\nabla \cdot (\nabla U^n) = \frac{\partial}{\partial x_i} \left(\frac{\partial}{\partial x_i} u_j^n \right)_{ij} \quad (21)$$

对于对流项 $U^n \cdot (\nabla U^n)$ ，其过程与对剪切应力项的计算过程类似，进行速度 U^n 与该二阶张量的内积运算，得到：

$$U^n \cdot (\nabla U^n) = u_i^n \left(\frac{\partial}{\partial x_i} u_j^n \right)_{ij} \quad (22)$$

最后利用爱因斯坦求和约定，对相同指标 i 进行加和处理，得到速度量矫正方程的指标形式：

$$u_i^* = u_i^n + \Delta t \left(\sum_{i=1}^2 v \frac{\partial}{\partial x_i} \left(\frac{\partial}{\partial x_i} u_j^n \right)_{ij} - \sum_{i=1}^2 u_i^n \left(\frac{\partial}{\partial x_i} u_j^n \right)_{ij} \right) (j = 1, 2) \quad (23)$$

写作分量形式，将 x_1 替换为 x ，将 x_2 替换为 y ，将 u_1 替换为 u ，将 u_2 替换为 v ，得：

$$\begin{cases} u^* = u^n + \Delta t \left[v \left(\frac{\partial^2 u^n}{\partial x^2} + \frac{\partial^2 u^n}{\partial y^2} \right) - \left(u^n \frac{\partial u^n}{\partial x} + v^n \frac{\partial u^n}{\partial y} \right) \right] \end{cases} \quad (24)$$

$$\begin{cases} v^* = v^n + \Delta t \left[v \left(\frac{\partial^2 v^n}{\partial x^2} + \frac{\partial^2 v^n}{\partial y^2} \right) - \left(u^n \frac{\partial v^n}{\partial x} + v^n \frac{\partial v^n}{\partial y} \right) \right] \end{cases} \quad (25)$$

在交错网格上使用中心差分格式。

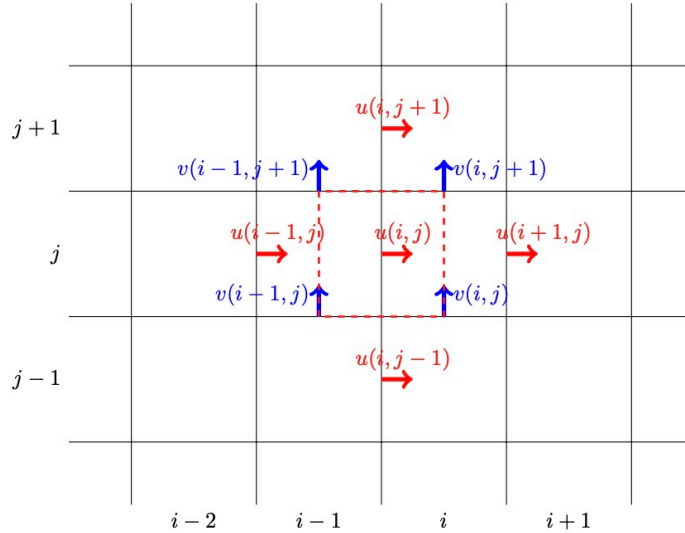


Figure. 2. 速度 u 在交错网格上的近似

在方程 24 中，采用二阶中心差分格式对速度 u^n 对于 x 和 y 的二次偏导数进行近似，得到：

$$\left(\frac{\partial^2 u^n}{\partial x^2} \right)_{j,i} \sim \frac{u_{j,i+1}^n - 2u_{j,i}^n + u_{j,i-1}^n}{\Delta x^2} \quad (26)$$

$$\left(\frac{\partial^2 u^n}{\partial y^2} \right)_{j,i} \sim \frac{u_{j+1,i}^n - 2u_{j,i}^n + u_{j-1,i}^n}{\Delta y^2} \quad (27)$$

在方程 24 中，采用一阶中心差分格式对速度 u^n 对于 x 和 y 的一次偏导数进行近似，得到：

$$\left(\frac{\partial u^n}{\partial x}\right)_{j,i} \sim \frac{u_{j,i+1}^n - u_{j,i-1}^n}{\Delta 2x} \quad (28)$$

$$\left(\frac{\partial u^n}{\partial y}\right)_{j,i} \sim \frac{u_{j+1,i}^n - u_{j-1,i}^n}{\Delta 2y} \quad (29)$$

对于方程 24 中的 v^n ，对其进行平均操作，得到：

$$v_{u,j,i}^n = \frac{1}{4} (v_{j,i}^n + v_{j+1,i}^n + v_{j+1,i-1}^n + v_{j,i-1}^n) \quad (30)$$

因此，方程 45 经由空间离散操作后可以得到：

$$\begin{cases} u_{j,i}^* = u_{j,i}^n + \Delta t \left\{ \nu \left[\frac{u_{j,i+1}^n - 2u_{j,i}^n + u_{j,i-1}^n}{\Delta x^2} + \frac{u_{j+1,i}^n - 2u_{j,i}^n + u_{j-1,i}^n}{\Delta y^2} \right] - \left[u_{j,i}^n \left(\frac{u_{j,i+1}^n - u_{j,i-1}^n}{2\Delta x} \right) + v_{u,j,i}^n \left(\frac{u_{j+1,i}^n - u_{j-1,i}^n}{2\Delta y} \right) \right] \right\} \end{cases} \quad (31)$$

$$\begin{cases} v_{u,j,i}^n = \frac{1}{4} (v_{j,i}^n + v_{j+1,i}^n + v_{j+1,i-1}^n + v_{j,i-1}^n) \end{cases} \quad (32)$$

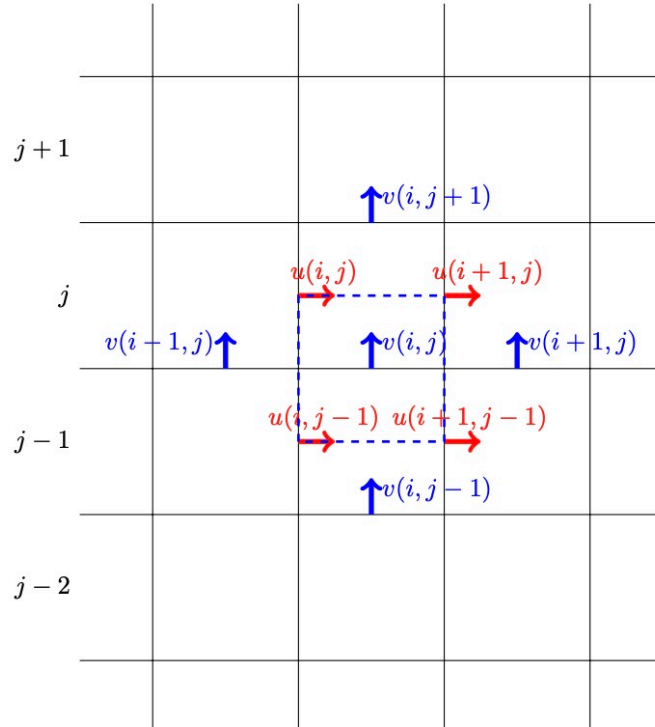


Figure. 3. 速度 v 在交错网格上的近似

同理可得在 y 方向的速度修正量 v 在空间离散后的速度修正量方程：

$$\begin{cases} v_{j,i}^* = v_{j,i}^n + \Delta t \left\{ \nu \left[\frac{v_{j,i-1}^n - 2v_{j,i}^n + v_{j,i+1}^n}{\Delta x^2} + \frac{v_{j-1,i}^n - 2v_{j,i}^n + v_{j+1,i}^n}{\Delta y^2} \right] - \left[v_{j,i}^n \left(\frac{v_{j,i+1}^n - v_{j,i-1}^n}{2\Delta x} \right) + v_{v,j,i}^n \left(\frac{v_{j+1,i}^n - v_{j-1,i}^n}{2\Delta y} \right) \right] \right\} \end{cases} \quad (33)$$

$$\begin{cases} v_{v,j,i}^n = \frac{1}{4} (v_{j,i}^n + v_{j,i+1}^n + v_{j+1,i+1}^n + v_{j+1,i}^n) \end{cases} \quad (34)$$

对压力泊松方程的空间离散

对于压力泊松方程

$$\nabla^2 p^{n+1} = \frac{\rho}{\Delta t} \nabla \cdot U^* \quad (35)$$

首先将拉普拉斯算子拆分为梯度的散度操作。计算哈密顿算符在与标量 p 作用之后得到的向量再与哈密顿算符进行内积运算，右侧同样进行内积运算，得到压力泊松方程的指标形式：

$$\sum_{i=1}^2 \left[\frac{\partial}{\partial x_i} \left(\frac{\partial p^{n+1}}{\partial x_i} \right) \right] = \sum_{i=1}^2 \frac{\rho}{\Delta t} \frac{\partial u_i^*}{\partial x_i} \quad (36)$$

将 x_1 替换为 x ，将 x_2 替换为 y ，将 u_1 替换为 u ，将 u_2 替换为 v ，得：

$$\frac{\partial^2 p^{n+1}}{\partial x^2} + \frac{\partial^2 p^{n+1}}{\partial y^2} = \frac{\rho}{\Delta t} \left(\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \right) \quad (37)$$

对其中的一阶导数与二阶导数在 p 网格上进行空间离散，得到压力泊松方程的空间差分格式：

$$\frac{p_{j,i-1}^{n+1} - 2p_{j,i}^{n+1} + p_{j,i+1}^{n+1}}{\Delta x^2} + \frac{p_{j+1,i}^{n+1} - 2p_{j,i}^{n+1} + p_{j-1,i}^{n+1}}{\Delta y^2} = \frac{\rho}{\Delta t} \left(\frac{u_{j,i+1}^* - u_{j,i}^*}{\Delta x} + \frac{v_{j+1,i}^* - v_{j,i}^*}{\Delta y} \right) \quad (38)$$

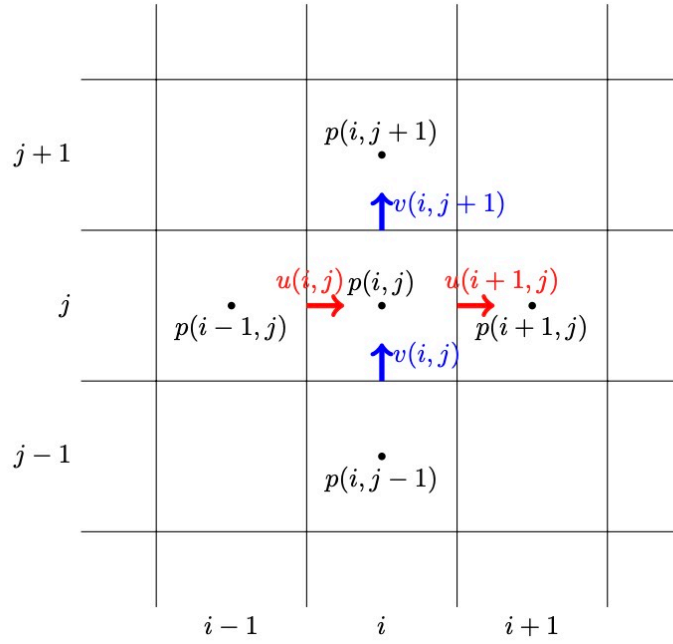


Figure. 4. 压力 p 在交错网格上的近似

对速度场时间步进方程的空间离散

对于速度场时间步进方程

$$U^{n+1} = U^* - \frac{\Delta t}{\rho} \nabla p^{n+1} \quad (39)$$

同理可得其指标形式：

$$\sum_{i=1}^2 u_i^{n+1} = \sum_{i=1}^2 u_i^* - \frac{\Delta t}{\rho} \left(\sum_{i=1}^2 \frac{\partial p^{n+1}}{\partial x_i} \right) \quad (40)$$

将 x_1 替换为 x ，将 x_2 替换为 y ，将 u_1 替换为 u ，将 u_2 替换为 v ，得：

$$\begin{cases} u^{n+1} = u^* - \frac{\Delta t}{\rho} \left(\frac{\partial p^{n+1}}{\partial x} \right) \end{cases} \quad (41)$$

$$\begin{cases} v^{n+1} = v^* - \frac{\Delta t}{\rho} \left(\frac{\partial p^{n+1}}{\partial y} \right) \end{cases} \quad (42)$$

最后得到速度场时间步进方程的空间差分格式：

$$\begin{cases} u_{j,i}^{n+1} = u_{j,i}^* - \frac{\Delta t}{\rho} \left(\frac{p_{j,i}^{n+1} - p_{j,i-1}^{n+1}}{\Delta x} \right) \end{cases} \quad (43)$$

$$\begin{cases} v_{j,i}^{n+1} = v_{j,i}^* - \frac{\Delta t}{\rho} \left(\frac{p_{j,i}^{n+1} - p_{j-1,i}^{n+1}}{\Delta y} \right) \end{cases} \quad (44)$$

至此，我们已经完成了纳维斯托克斯方程的时间离散与空间离散，下面我们尝试通过计算机编程解决计算问题。

4.5 求解压力泊松方程的 Jacobi 迭代算法

对于压力泊松方程

$$\nabla^2 p^{n+1} = \frac{\rho}{\Delta t} \nabla \cdot U^* \quad (45)$$

我们将右侧的 $\frac{\rho}{\Delta t} \nabla \cdot U^*$ 视作已知项并写作 R ，则对于求解域内的任意网格 (j,i) 处，压力泊松方程可以写作：

$$\nabla^2 p_{j,i}^{n+1} = R_{j,i} \quad (46)$$

我们写出所有网格处的压力泊松方程，将 \mathbf{p}^{n+1} 作为所有网格处压力 p 的列向量，将 \mathbf{R} 作为所有网格处 R 的值的列向量。故所有网格处的关系可以写作：

| 网格坐标 | \mathbf{Lp} | \mathbf{R} |
|-------|--------------------------|--------------|
| (1,1) | $\nabla^2 p_{1,1}^{n+1}$ | $R_{1,1}$ |
| (1,2) | $\nabla^2 p_{1,2}^{n+1}$ | $R_{1,2}$ |
| (1,3) | $\nabla^2 p_{1,3}^{n+1}$ | $R_{1,3}$ |
| (1,4) | $\nabla^2 p_{1,4}^{n+1}$ | $R_{1,4}$ |
| (1,5) | $\nabla^2 p_{1,5}^{n+1}$ | $R_{1,5}$ |
| (2,1) | $\nabla^2 p_{2,1}^{n+1}$ | $R_{2,1}$ |
| ... | ... | ... |

$$\mathbf{Lp} = \mathbf{R} \quad (47)$$

对于这一线性方程组，采用 Jacobi 迭代法，Jacobi 迭代的矩阵形式为：

$$\mathbf{x}^{k+1} = -\mathbf{D}^{-1} \mathbf{R} \mathbf{x}^k + \mathbf{D}^{-1} \mathbf{b} \quad (48)$$

将其转化为指标形式：

$$x_i^{k+1} = -D_{ii}^{-1} R_{ij} x_j^k + D_{ii}^{-1} b_i \quad (49)$$

随后进行编程：

Fortran Code: Jacobi Method

```
module linear
  implicit none

  contains

  subroutine jacobi(A, x, b, n, k)
    implicit none
    integer, intent(in) :: n, k
    real, intent(in) :: A(n, n), b(n)
    real, intent(out) :: x(n)
    integer :: iter, i, j
    real :: s

    x = 0.0

    do iter = 1, k
      do i = 1, n
        s = 0.0
        do j = 1, n
          if (j /= i) then
            s = s + A(i, j) * x(j)
          end if
        end do
        x(i) = (b(i) - s) / A(i, i)
      end do
    end do
  end subroutine jacobi
end module linear
```