

无痛苦N-S方程笔记

An Introduction of The Navier-Stokes Equation
Without Pain

Dongyue Li
李东岳



裂痕是什么

那是阳光照进来的地方

前言 CFD 糊合物理、数值方法和计算机科学于一体，用于模拟流动状态。早期工作可追溯到 1917 年，Lewis Richardson 尝试手算偏微分方程组来预测天气预报。1943 年，Los Alamos 实验室，意图通过求解偏微分方程来预测飞行器的激波位置并用于军事行动。这被认为是 CFD 的雏形。当时，CFD 被 Los Alamos 实验室当做一种机密黑科技。到了 70 年代，得益于计算机技术的高速发展，CFD 被大量地应用于航空航天的流场模拟中。在航空行业的大力支持下，大量的湍流模型顺势而生。从某种程度上说，早期 CFD 的发展和计算机的发展直接相关。多亏大型计算集群，1980 年代，已经有了计算二维势流和三维欧拉方程的能力。随后，一些加速数值算法也被提出，如多重网格技术 [22]。80 年代中后期，CFD 已经可以用于求解粘性 N-S 方程。与此同时，大量的涡粘模型被提出。随后，直接模拟 (Direct Numerical Simulation, DNS) 以及大涡模拟 (Large Eddy Simulation, LES) 成为了 CFD 中一种解析度非常高的模拟技术。今天，连智能手机都配备了 8G 内存，4 核心的处理器几乎成为最低配置，甚至有人用手机进行小型的 CFD 计算。2020 年，计算机算力大大提升。CFD 与各个领域相结合，成为了一种辅助工程设计的技术。学术界，CFD 也与各种学科相互交叉，研究成果日新月异，层出不穷。呈现一种蓬勃发展的态势。

在这里，有必要区分一下流体力学与 CFD。二者之间存在不可割舍的联系，但差异也非常明显。从 CFD 的名字上来理解，Computational 译为“计算”，Fluid Dynamics 则为“流体动力学”。因此 CFD 即为计算流体力学。流体力学着重研究如何构建偏微分方程，CFD 着重研究如何计算偏微分方程。因此，CFD 也可以理解为首先通过流体力学研究构建对应的数学模型，然后通过数值方法对模型进行求解的过程。流体力学侧重建模，CFD 侧重数值求解。没有流体力学构建的模型 CFD 无用武之地，流体力学构建模型之后没有 CFD 则不可能获得解。举例，多相流体力学的研究者可能对两个气泡之间如何破碎感兴趣，因此这些研究者的工作可能是通过实验监控气泡周围的流场数据（如湍流动能），构建一个破碎频率关于湍流动能的关系式。多相计算流体力学的研究者在拿到这个关系式之后，需要用数值方法去对其进行求解。另一方面，流体力学研究者离不开实验，CFD 研究者可以无实验进行研究。流体力学的顶级期刊是 Journal of Fluid Mechanics 等，CFD 的顶级期刊是 Journal of Computational Physics 等。流体力学与 CFD，也存在一种所属关系。CFD 属于流体力学研究，但不能认为流体力学研究就是 CFD。同时，Journal of Fluid Mechanics 经常会看到偏 CFD 的研究。但 Journal of Computational Physics 上流体力学的研究要少的多。所以，CFD 应该属于流体力学研究的一个小类。正如 Ferziger 等人所说 [34]：

“CFD 可以看做是流体力学或者数值算法的一个分支。CFD 从业者需要在这两方面具备牢固的背景知识。如果偏科，很有可能会得到错误的模拟结果。我们希望读者能重视这一点，并学习相关的知识。”

在生活中，流体的流动也吸引了科学家广泛的关注并进行了深入的研究。Bourouiba 等针对人们打喷嚏的现象，详细的研究了喷嚏液滴的流动状态，得出喷嚏飞沫可以漂浮 8

米远的结论，在 2020 年的新冠疫情中，对普通百姓防疫起了重要作用。[Gillespie and Hamilton](#)对一种烤肉架进行 CFD 模拟分析，认为经过 CFD 优化后的烤肉架炙烤后的菜品更加美味并申请了美国专利。[Centaur 公司](#)通过将 CFD 温控技术加入粮食储备系统，可以实时监控粮食仓储系统的温湿度，防止粮食变质，并拿下 3000 万人民币的融资。著名卫浴领导品牌[Roca](#)，为了提升马桶的用户体验，通过 CFD 对马桶进行了模拟，研究了水管的形状、弯度、管道直径、需水量等影响因素对流体流动的影响。此外，CFD 在汽车领域的研究已属家常便饭，[蔚来](#)、[比亚迪汉](#)、甚至[宾利](#)都通过 CFD 来对新能源汽车进行设计，力图减少风阻增加续航。[Wallevik](#) 等采用开源 CFD 软件，通过两相流模型，研究混凝土搅拌车的搅拌效果。除了地上跑的汽车，[美国宇航局](#)甚至用自编 CFD 代码研究太空飞的陨石，力图降低陨石在高空爆炸引起的危害。更为新奇的，有人用 CFD 的方法研究[精子运动与人类受孕](#)，研究[喷墨打印机](#)的喷口流动，[Rabino](#)甚至用 CFD 商业软件来分析女性平胸与巨乳的阻力系数，得到了一些令人瞠目结舌的结果。

总而言之，CFD 已经成为了一个上可飞天，下可落地的技术。各个行业的工程师，但凡涉及到流体的学问，必然都会诉求 CFD 来解决实际问题。然而，CFD 并不像 word 软件，拿来就能用，随便就能学。大量的从业者表示，CFD 学习的曲线非常陡峭。商业软件的 CFD 学习尚且相对容易。开源软件的 CFD 学习，以及 CFD 算法的学习，则是难上加难。但是，要将 CFD 学成真本领，还真得从算法出发，上升至应用，才能将 CFD 真正掌握，玩弄于股掌。同时，商业 CFD 软件与自编 CFD 软件基本上是两条路。商业软件侧重 CFD 应用，自编/开源 CFD 软件侧重算法开发。我国数值传热学泰斗、中国科学院院士陶文铨教授，曾为宇波教授等人著作的《数值传热学实训》做序，其中写道 [15]：

“如果我们培养的大学生甚至研究生，只会使用商业软件，而不会开发自己的，特别考虑到目前广为使用的商业软件均是国外的产品，那么多年后我们国家的工程技术和研究人员的开发水平会是什么样子？不堪想象！”

2019 年起，中美贸易摩擦频繁发生，除了百姓的必需品大豆之外，连芯片及工业软件（如 MATLAB），均被限制出口。华为、中兴的业务惨遭锁喉。以计算机芯片为例，国外的芯片设备商掌握了复杂的芯片制造工艺，随着工艺越来越高精尖，芯片设备商的话语权一步一步的提升。因此，在过去的长达几十年里，美国一直以各种手段，保证掌控芯片制作工艺，保证自己在设备领域的绝对领导地位。在美国贸易霸权之下，2020 年 8 月 7 日，华为余承东公开表示，因为美国的禁令，台积电不再为华为代工，华为的海思麒麟芯片将绝版。

类似于芯片，CFD 软件，作为 CAE 大系中的一个小类，一直都是中国卡脖子的技术难点。国外的开源软件 OpenFOAM，用了 40 年的时间发展到了今天。中国的 CFD 大牛并不少，甚至涌现出很多世界一流的 CFD 研究学者，在这些学者的领导下，出现了一些 in-house 的 CFD 求解器。另一方面，很多优秀的普适性 CFD 求解器甚至都开源了。但是为什么中国做不出一个类似 OpenFOAM 的强大的普适性 CFD 求解器。这与我们如何看待

待知识产权的态度、如何评价科研成果等有直接关系。要做中国自主的 CFD 软件，需要 5 年、10 年，甚至几代人的努力。任正非曾经感叹：

“我们国家修桥、修路、修房子，已经习惯了砸钱就行。但是芯片砸钱不行，得砸数学家、物理学家、化学家。”

因此，如果这份笔记，能让我国的 CFD 人才少走弯路，或激发相关人员研究 CFD 的兴趣，也算作为一个 CFD 从业者，为我国整个 CFD 行业应尽的一份责任。

联系方式 本笔记为李东岳博士讲授的[东岳流体 CFD 课程](#)笔记。笔记中算法全部来源于开源 CFD 软件 OpenFOAM。可作为 OpenFOAM 使用人员的参考资料。内容也可为自编 CFD 软件开发人员提供算法角度的参考。作者水平有限，一直对算法保持敬畏，难免有不妥和错误之处，敬请各位老师同学指正。本笔记采用 L^AT_EX 制作。勘误、增补等可前往[CFD 中文网](#)，或邮件联系：li.dy@dyfluid.com

引用 Li. D, An Introduction of The Navier-Stokes Equation Without Pain (in Chinese), DYFLUID, Beijing, 2019.

时间戳 电子版首发于 2019 年 2 月 14 日，最新修订于 2022 年 8 月 27 日



勘误:

- 190705 陈佳 增加脚注 3
 190619 谢鹏 更正若干笔误
 190610 沈学峰 更正若干笔误
 190504 Frank0514 更正若干笔误
 190610 刘威 更正 8 个问题
 190607 汪洋 更正若干笔误
 190314 灿 更正交叉引用
 190222 金国庆 (2.31)左侧第二项 $\frac{\partial}{\partial t}$ 更正为 $\frac{\partial}{\partial x}$
 191023 蔡欣 更正方程(3.79)下标
 191204 梁钢 更正方程(3.46)、(3.47)、(3.48)
 191221 陈东林 更正若干笔误
 200207 文洋 增加方程(7.110)中的 Δx
 200318 赵鹏 更正方程(7.117)的索引
 200326 周佳其 更正若干笔误
 200716 Yongbo 更正方程索引
 200826 姚卫 更正方程(3.81)的 k 项
 210102 王飞 更正若干笔误
 210219 ZY1905417 更正若干笔误
 210225 王忠琨 更正若干笔误
 210317 张仕钊 更正方程(5.2)正负号
 210319 方筑 更正若干笔误
 210325 warnerchang 更正若干笔误
 210818 袁超 更正若干笔误
 210903 Kingkong 更正若干笔误
 210921 向北 更正若干笔误
 211021 王飞 更正若干笔误
 211209 袁宝强 更正若干笔误
 220320 Tong 更正一些概念问题
 220406 管策 更正一处笔误
 220527 周星光 更正雷诺应力扩散项和湍动能耗散率扩散项中少写了一个点
 220711 郑亮 增加方程(3.79)的脚注
 220802 沙小涵 修正脚注 3 的时间项
 220802 路天庆 修正泰勒公式若干标识
 220827 J. Hu 修正泰勒公式若干标识

目录

第一章 不只是 N-S 方程	1
第二章 方程标识	3
2.1 矢量标识法	3
2.2 张量标识法	5
2.3 偏导标识法	7
2.4 运算符标识法	8
2.5 $\nabla \mathbf{U}$ 的不同写法	8
第三章 流体与 N-S 方程	11
3.1 导论	11
3.2 泰勒公式	13
3.3 流动模型	13
3.3.1 有限控制体模型	14
3.3.2 无穷小微团模型	14
3.3.3 物质导数	15
3.4 连续性方程	17
3.4.1 微分形式	18
3.4.2 积分形式	20
3.4.3 通量与速度散度的物理意义	21
3.4.4 拉格朗日观点	25
3.4.5 连续性方程小结	25
3.5 动量方程	26
3.5.1 受力分析	26
3.5.2 动量守恒	29
3.5.3 守恒/非守恒转化	30
3.5.4 封闭	31

3.5.5 积分观点	33
3.6 学习要求与思考	35
第四章 多相流模型	37
4.1 多尺度模型	39
4.1.1 微观模型	39
4.1.2 宏观模型	41
4.1.3 介尺度模型	43
4.1.4 多尺度模型的适用性	44
第五章 数值方法	47
5.1 稀疏线性系统求解器	47
5.1.1 基本迭代求解器	47
5.1.2 多重网格求解器	49
第六章 拓展内容	51
6.1 湍流	51
6.1.1 直接模拟 DNS	52
6.1.2 大涡模拟 LES	53
6.1.3 雷诺平均 RANS	57
6.1.4 二维 DNS/LES	58
6.1.5 RANS-LES 混合模型	59
6.2 结构网格与非结构网格	61
6.3 动理学方程	62
6.3.1 麦克斯韦分布与高斯分布	62
6.3.2 矩方程、五矩模型（欧拉方程）、十矩模型	64
6.3.3 欧拉方程（一个推导实例）	65
6.4 Von Neumann 稳定性分析与截断误差	67
6.4.1 变量分离法	67
6.4.2 抛物线方程稳定性分析、无条件稳定	69
6.4.3 截断误差	70
6.5 双曲系统与高阶格式	71
6.5.1 黎曼问题	72
6.5.2 双曲系统、特征变量	72
6.5.3 积分形式与守恒方程	75
6.5.4 封闭与基架点	76

目录	XI
6.5.5 中心格式	76
6.5.6 Lax-Friedrichs 格式	76
6.5.7 Kurganov-Tadmor 格式	77
6.5.8 Godunov 格式	77
6.6 为什么 $-\frac{2}{3}\mu(\nabla \cdot \mathbf{U})\mathbf{I}$ 为体膨胀系数项?	79
6.7 什么是应力的 Deviatoric 与 Hydrostatic 部分?	79
6.8 二阶张量不变量: Invariants	80
6.9 如何理解雷诺应力项中的 $\frac{2}{3}\rho k\mathbf{I}$?	81
6.10 变量有界性	82
6.11 守恒与非守恒、守恒变量与原始变量	83
6.12 分离式求解器与耦合式求解器	84
6.13 显性离散、隐性离散	85
6.14 Jacobian	85
6.15 非牛顿流体	86
6.16 算子分裂法	88
6.17 总温、驻点温度	89
第七章 OpenFOAM 模型速查	91
7.1 OpenFOAM 中的湍流模型	91
7.1.1 RANS 湍流模型	91
7.1.2 LES 湍流模型	100
7.1.3 RANS-LES 混合模型	102
7.2 OpenFOAM 中的壁面函数	103
7.3 OpenFOAM 中的非结构网格	105
7.3.1 point	105
7.3.2 line 和 edge	106
7.3.3 face, 面心, 面积	106
7.3.4 虚拟的 cell, 体心, 体积	109
7.3.5 偏斜度	113
7.3.6 非结构网格 TVD 格式	114
7.4 OpenFOAM 中的有限体积离散	117
7.4.1 显性对流项 $\nabla \cdot (\mathbf{U}T)$ 、通量 $\sum \mathbf{U}_f \cdot \mathbf{S}_f$ 计算	117
7.4.2 显性拉普拉斯项 $\nabla \cdot (\nabla T)$ 计算	119
7.4.3 显性面法向梯度项 $(\nabla T)_f \cdot \frac{\mathbf{S}_f}{ \mathbf{S}_f }$ 计算	119
7.4.4 显性梯度项 ∇T 计算	121

7.4.5 显性时间项计算	121
7.5 OpenFOAM 中的稀疏线性系统求解器	122
7.5.1 稀疏线性系统、 <code>lduMatrix</code> , <code>fvMatrix</code> 等	122
7.5.2 隐性扩散项离散	127
7.5.3 拉普拉斯项非正交修正	129
7.5.4 隐性对流项离散	129
7.5.5 稀疏线性系统带宽减小算法	132
7.5.6 将网格单元设为固定值	133
7.5.7 <code>fvMatrix</code> 之 Gauss-Seidel 求解器	134
7.5.8 <code>fvMatrix</code> 的单位	136
7.5.9 <code>fvMatrix</code> 的 <code>H()</code> 、 <code>H1()</code> 函数	137
7.5.10 <code>fvMatrix</code> 的“减法”操作符	139
7.5.11 <code>fvMatrix</code> 的 <code>correction()</code> 函数	140
7.6 OpenFOAM 中的多重参考系	143
7.7 OpenFOAM 中的多孔介质模型	144
7.8 OpenFOAM 中的 IATE 模型	147
7.9 OpenFOAM 中的常用代码	149
7.9.1 <code>laplacian(ϕ) = div(grad(ϕ))?</code>	149
7.9.2 <code>-fvc::flux(-phir,..) = fvc::flux(phir,..)?</code>	150
7.9.3 <code>pEqn.flux()</code> 函数	151
7.9.4 <code>fvc::div(phiHbyA)</code> 函数	152
7.9.5 <code>fvc::reconstruct()</code> 函数	153
7.9.6 <code>setReference()</code> 函数	154
7.9.7 <code>bound()</code> 函数	155
7.9.8 <code>adjustPhi()</code> 函数	156
7.9.9 <code>CourantNo</code> 三维库朗数	157
7.9.10 <code>deltaCoeffs()</code> 函数	157
7.10 OpenFOAM 中的曳力模型	158
7.10.1 欧拉方法	158
7.11 OpenFOAM 中的状态方程	160
7.12 OpenFOAM 中的边界条件	161
7.12.1 边界条件计算方法	161
7.12.2 OpenFOAM 边界条件速查	162

附录 A 10 网格 1D 方程离散	169
A.1 扩散方程 1	169
A.2 扩散方程 2	172
A.3 扩散方程 3	174

第一章 不只是 N-S 方程

谈到 CFD，大部分人最开始想到的就是 Navier-Stokes 方程（N-S 方程）。N-S 方程从守恒定律推导而来：

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0, \quad (1)$$

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \nabla \cdot \tau, \quad (2)$$

其中 ρ 为密度， \mathbf{U} 为速度， p 为压力， τ 为剪切应力。N-S 方程具有以下特点：

- 方程(2)中左边第二项是关于 \mathbf{U} 的偏导数，这种未知量和未知量乘积的问题构成非线性问题，CFD 对非线性问题需要特殊处理。另一方面，非线性的双曲问题的解可能会存在间断（如激波）。激波通常存在于高超声速的欧拉问题求解中。同时，非线性项也是湍流在数学方程中的体现；
- 方程(2)的数学特征为抛物线。不同数学特征的问题需要调用不同的时间/空间离散格式，隐性时间格式更有利于求解抛物线问题。若方程(2)中省略若干项则会改变方程的数学特征，例如若将方程右侧置为 0，则变为双曲特征的欧拉方程。欧拉方程得益于其双曲特性，可采用迎风类显性算法推进，各种基于有限体积法的高分辨率格式因此而生（交错网格中心格式、中心-迎风格式等）。同时，动量方程的对流项和扩散项的相对强弱，也会影响边界条件的设置（如 DNS 的出口无反射边界条件）；
- 在马赫数较大时（如大于 0.3），方程(1)可用来求解密度，方程(2)可用来求解速度，同时附加能量方程求解温度以及状态方程求解压力，即密度基求解器。在马赫数较小时，并没有单独的压力方程，并且方程(1)缺少主要求解变量。这导致压力的求解需要特殊的策略。这也是 CFD 中压力基 SIMPLE/PISO 算法要处理的问题。在构建离散矩阵的情况下，对速度和压力整合处理还是单独处理，是分离求解与耦合算法的重要问题；
- N-S 方程之前传：N-S 方程为宏观方程，调用了宏观假定，其可从玻尔兹曼方程附加 Chapman-Enskog 展开推导而来。N-S 方程也可以看做从介尺度模型演化的宏观二阶矩模型。在无压力无粘性的条件下具备弱双曲特征。由于失去了高阶矩的统计学特征，因此 N-S 方程在某些情况下是不适用的，比如稀薄流、微流动、以及气固多相流中的颗粒轨迹交叉现象都不能用 N-S 方程来描述。其数学本源在于 N-S 方程假定流体为

一种近似均衡状态，在某一个网格点存在一个单值的分布。然而，在分子/颗粒数量非常小的情况下，这些分子/颗粒基本无碰撞，会导致多值问题的产生。因此，在这种情况下，需要对统计学分布的高阶矩进行追踪，N-S 方程作为二阶矩模型是往往不够的。

正如上面讨论的，N-S 方程中存在大量的数学问题，初学者也看的不明不白。这无关紧要。本章仅仅抛出一块砖头，让大家了解隐藏在 CFD 中的方程的美妙含义。这些方程中充满火星符号（如 ∇ , ∂ ），也让人摸不清头脑。[东岳流体 CFD 课程](#)因此分为两步，第一步是基本的 N-S 方程入门，第二步是 N-S 方程求解。第一步需要同学们通过本笔记进行预习。第二步将在课程上讲授。

第二章 方程标识

学习 CFD 理论首先遇到的就是各种各样的偏微分算符。能看懂、拆分 CFD 方程是研究算法的最基本步骤。给出一个 CFD 方程（如动量方程），可以不知道是怎么推出来的，但要能看懂。本章介绍 CFD 方程的各种写法，不涉及到任何的 CFD 算法。

2.1 矢量标识法

本笔记中主要采用矢量标识法讨论 CFD 方程，在矢量标识法中，标量全部采用斜体，如压力 p 。矢量采用正体加粗，如速度矢量 \mathbf{U} ，其具有三个分量 u_1, u_2, u_3 或 u, v, w 。二阶张量也采用正体¹，比如应力张量 $\boldsymbol{\tau}$ ，其具备 9 个分量，其可表示为：

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \end{bmatrix} \quad (2.1)$$

下面粘度 ν 为 1 的不可压缩流体动量方程为例对其进行拆分，这个方程若采用矢量标识法可以写为：

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U} \mathbf{U}) = -\nabla \frac{p}{\rho} + \nabla \cdot (\nabla \mathbf{U}) \quad (2.2)$$

其中的 \mathbf{U} 为速度矢量， p 为压力， ρ 为密度。一般来讲，CFD 文献中通常采用方程(2.2)的形式，而并不进行展开进而更加紧凑。下面介绍如何将其展开为 3 个方程。

- 方程(2.2)第一项表示 \mathbf{U} 对时间的偏导数，因为 \mathbf{U} 为矢量，故其导数的分量形式为：

$$\frac{\partial \mathbf{U}}{\partial t} = \begin{bmatrix} \frac{\partial u_1}{\partial t} \\ \frac{\partial u_2}{\partial t} \\ \frac{\partial u_3}{\partial t} \end{bmatrix} \quad (2.3)$$

其中 u_1 表示 x 方向速度， u_2 表示 y 方向速度， u_3 表示 z 方向速度。这样拆分之后的方程，即为各个方向的速度针对时间的偏导数。如果理解方程(2.3)有困难，那么有必要预习一下《高等数学》（同济大学版）第二章。

¹如果难以理解二阶张量的含义，可以这样尝试：矢量是一阶张量，具有三个分量，二阶张量则具有 9 个分量。

- 方程(2.2)第二项 $\nabla \cdot (\mathbf{U}\mathbf{U})$ 中的 $\mathbf{U}\mathbf{U}$ 是一种简写，完整形式为 $\mathbf{U} \otimes \mathbf{U}$ ， \otimes 是一个张量运算符。依据 \otimes 的定义， $\mathbf{U}\mathbf{U}$ 可以写为：

$$\mathbf{U} \otimes \mathbf{U} = \mathbf{U}\mathbf{U} = \mathbf{U} \cdot \mathbf{U}^T = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} [u_1, u_2, u_3] = \begin{bmatrix} u_1u_1 & u_1u_2 & u_1u_3 \\ u_2u_1 & u_2u_2 & u_2u_3 \\ u_3u_1 & u_3u_2 & u_3u_3 \end{bmatrix} \quad (2.4)$$

接下来看 $\nabla \cdot$ ，其为散度算符，有时用 div 来表示。对一个矢量（1 阶张量）做散度的结果为一个标量（0 阶张量），对一个 2 阶张量做散度的结果为矢量（1 阶张量）。因此，对任意 n 阶张量做散度操作之后，结果为 $n - 1$ 阶张量。举例，对一个矢量 \mathbf{U} 做散度有： $\nabla \cdot \mathbf{U} = \frac{\partial u_1}{\partial x} + \frac{\partial u_2}{\partial y} + \frac{\partial u_3}{\partial z}$ 。因此，方程(2.2)中的第二项 $\nabla \cdot (\mathbf{U}\mathbf{U})$ 即为对一个 2 阶张量做散度：

$$\nabla \cdot (\mathbf{U}\mathbf{U}) = \nabla \cdot \begin{bmatrix} u_1u_1 & u_1u_2 & u_1u_3 \\ u_2u_1 & u_2u_2 & u_2u_3 \\ u_3u_1 & u_3u_2 & u_3u_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial u_1u_1}{\partial x} + \frac{\partial u_2u_1}{\partial y} + \frac{\partial u_3u_1}{\partial z} \\ \frac{\partial u_1u_2}{\partial x} + \frac{\partial u_2u_2}{\partial y} + \frac{\partial u_3u_2}{\partial z} \\ \frac{\partial u_1u_3}{\partial x} + \frac{\partial u_2u_3}{\partial y} + \frac{\partial u_3u_3}{\partial z} \end{bmatrix} \quad (2.5)$$

- 方程(2.2)第三项也存在 ∇ ，这一项中没有了 \cdot 符号。单独的 ∇ 表示梯度运算，有时也被写为 grad 。对一个标量（0 阶张量）做梯度的结果为一个矢量（1 阶张量），对一个矢量做梯度的结果为 2 阶张量。因此，对任意 n 阶张量做梯度之后的结果为 $n + 1$ 阶张量。举例，对一个标量 p 做梯度有²：

$$\nabla p = \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \\ \frac{\partial p}{\partial z} \end{bmatrix} \quad (2.6)$$

方程(2.2)第四项 $\nabla \cdot (\nabla \mathbf{U})$ 为对速度 \mathbf{U} 先做梯度再做散度。 $\nabla \cdot \nabla$ 通常也写为 ∇^2 ，并称为拉普拉斯算子，有时也被写为 laplacian。即 $\nabla \cdot (\nabla \mathbf{U}) = \nabla^2 \mathbf{U}$ 。现尝试对其进行展开：

$$\nabla \cdot (\nabla \mathbf{U}) = \nabla \cdot \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_2}{\partial x} & \frac{\partial u_3}{\partial x} \\ \frac{\partial u_1}{\partial y} & \frac{\partial u_2}{\partial y} & \frac{\partial u_3}{\partial y} \\ \frac{\partial u_1}{\partial z} & \frac{\partial u_2}{\partial z} & \frac{\partial u_3}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} \left(\frac{\partial u_1}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial u_1}{\partial y} \right) + \frac{\partial}{\partial z} \left(\frac{\partial u_1}{\partial z} \right) \\ \frac{\partial}{\partial x} \left(\frac{\partial u_2}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial u_2}{\partial y} \right) + \frac{\partial}{\partial z} \left(\frac{\partial u_2}{\partial z} \right) \\ \frac{\partial}{\partial x} \left(\frac{\partial u_3}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial u_3}{\partial y} \right) + \frac{\partial}{\partial z} \left(\frac{\partial u_3}{\partial z} \right) \end{bmatrix} \quad (2.7)$$

结合方程(2.3)、(2.5)、(2.6)、(2.7)这四项，有三个方程。下面仅列出 x 方向（取各个项展开形式中的第一行）：

$$\frac{\partial u_1}{\partial t} + \frac{\partial u_1u_1}{\partial x} + \frac{\partial u_2u_1}{\partial y} + \frac{\partial u_3u_1}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \frac{\partial}{\partial x} \left(\frac{\partial u_1}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial u_1}{\partial y} \right) + \frac{\partial}{\partial z} \left(\frac{\partial u_1}{\partial z} \right) \quad (2.8)$$

²一般情况下压力 p 为标量，即为各向同性的。在某些模型中（如假定高斯分布的十矩模型），压力为二阶各向异性张量。

方程(2.8)即为关于 u_1 的方程。仔细观察方程(2.8)，其只不过是各种导数的加和。除了 u_1 之外，如果所有的变量均为已知，那么就可以求出 u_1 关于时间的步进（看第一项速度关于时间的导数）。方程(2.8)的特点可参考第一章论述的内容。

2.2 张量标识法

CFD 中的偏微分方程除了用矢量标识法表示外，还可以用张量标识法表示，这种方法在 SCI 论文以及书籍中也非常常见 [96]。在张量标识法中，符号的下标表示张量的阶数。如标量压力 p 可以记为（二者无区别）：

$$p \equiv p \quad (2.9)$$

矢量 \mathbf{U} 可以写为 u_i ：

$$u_i \equiv \mathbf{U} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (2.10)$$

二阶张量 $\boldsymbol{\tau}$ 定义为 τ_{ij} ：

$$\tau_{ij} \equiv \boldsymbol{\tau} = \begin{bmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \end{bmatrix} \quad (2.11)$$

一定需要注意，在使用张量标识法的时候， u_i 表示矢量而不是某个具体的分量。举例说明：如果某文章采用张量标识法，则 u_i 表示速度矢量， u_1 表示 x 方向速度分量。这和上一节的写法是不同的。在本笔记中， \mathbf{U} 表示速度矢量， u_i 表示各个方向的分量而不是速度矢量，其中 $i = 1, 2, 3$ 。因此，遇到 u_i ，采用不同的数学表示方法，其含义可能不同。张量标识法通常结合爱因斯坦操作符使用变成非常强大的武器。举例说明：当在做乘积操作的时候，如果某一项中下标重复，那么则需要进行加和，如：

$$a_i b_i = \sum_{i=1}^3 a_i b_i = a_1 b_1 + a_2 b_2 + a_3 b_3 = \mathbf{a} \cdot \mathbf{b} \quad (2.12)$$

$$\begin{aligned} \tau_{ij} \tau_{ij} &= \sum_{i=1}^3 \sum_{j=1}^3 \tau_{ij} \tau_{ij} = \\ \tau_{11}\tau_{11} + \tau_{12}\tau_{12} + \tau_{13}\tau_{13} + \tau_{21}\tau_{21} + \tau_{22}\tau_{22} + \tau_{23}\tau_{23} + \tau_{31}\tau_{31} + \tau_{32}\tau_{32} + \tau_{33}\tau_{33} &= \boldsymbol{\tau} : \boldsymbol{\tau} \quad (2.13) \end{aligned}$$

$$\tau_{ij} u_j = \sum_{j=1}^3 \tau_{ij} u_j = \begin{bmatrix} \tau_{11}u_1 + \tau_{12}u_2 + \tau_{13}u_3 \\ \tau_{21}u_1 + \tau_{22}u_2 + \tau_{23}u_3 \\ \tau_{31}u_1 + \tau_{32}u_2 + \tau_{33}u_3 \end{bmatrix} = \boldsymbol{\tau} \cdot \mathbf{U} \quad (2.14)$$

方程(2.14)中 $\tau_{ij}u_j =$ 的 j 出现两次, 需要加和。 i 出现一次, 表示矢量。

对于二阶张量, 将在一项中最少出现两个下标如 i, j 。例如下文方程中 $D_{ik}E_{kj}$ 中的 i, j 表示二阶张量, k 表示加和:

$$\tau_{ij} = D_{ik}E_{kj} = \sum_{k=1}^3 D_{ik}E_{kj} = D_{i1}E_{1j} + D_{i2}E_{2j} + D_{i3}E_{3j} = \mathbf{D} \cdot \mathbf{E} \quad (2.15)$$

$$\tau_{ij} = D_{ki}E_{kj} = \sum_{k=1}^3 D_{ki}E_{kj} = D_{1i}E_{1j} + D_{2i}E_{2j} + D_{3i}E_{3j} = \mathbf{D}^T \cdot \mathbf{E} \quad (2.16)$$

张量标识法中还存在若干基本函数, 如克罗内克函数 δ_{ij} :

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (2.17)$$

很明显:

$$\delta_{11} = \delta_{22} = \delta_{33} = 1 \quad (2.18)$$

$$\delta_{12} = \delta_{21} = \delta_{13} = \delta_{31} = \delta_{23} = \delta_{32} = 0 \quad (2.19)$$

同样的, 对于偏微分方程组也可以进行类似的表示。例如对于下面的方程 (即连续性方程):

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_1}{\partial x} + \frac{\partial \rho u_2}{\partial y} + \frac{\partial \rho u_3}{\partial z} \equiv \frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \quad (2.20)$$

其中 $\frac{\partial \rho u_i}{\partial x_i}$ 中出现两次 i , 则需要加和。对于下面的方程 (即动量方程), 其可以表示为:

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\nu \frac{\partial u_i}{\partial x_j} \right) \quad (2.21)$$

其中 $\frac{\partial u_i u_j}{\partial x_j}$ 中出现两次 j , 则需要加和。出现一次 i , 则表示分量。因此其为一个矢量的方程。现在随意从文献中抽取一个方程, 如文献 [96] 中 $k - \varepsilon$ 模型的湍流动能方程, 即方程(4.9) 为³:

$$\rho \frac{\partial k}{\partial t} + \rho u_j \frac{\partial k}{\partial x_j} = \tau_{ij} \frac{\partial u_i}{\partial x_j} - \rho \varepsilon + \frac{\partial}{\partial x_j} \left[(\mu + \mu_t/\sigma_k) \frac{\partial k}{\partial x_j} \right] \quad (2.22)$$

对其分析可知首先其为一个标量方程而非矢量方程组, 且其展开形式为:

$$\rho \frac{\partial k}{\partial t} + \sum_{j=1}^3 \rho u_j \frac{\partial k}{\partial x_j} = \sum_{i=1}^3 \sum_{j=1}^3 \tau_{ij} \frac{\partial u_i}{\partial x_j} - \rho \varepsilon + \sum_{j=1}^3 \frac{\partial}{\partial x_j} \left[(\mu + \mu_t/\sigma_k) \frac{\partial k}{\partial x_j} \right] \quad (2.23)$$

³读者在这里应该尝试将其写为矢量标识法。例如, 对 $\tau_{ij} \frac{\partial u_i}{\partial x_j}$ 展开有 ($\boldsymbol{\tau}$ 为对称二阶张量):

$$\tau_{ij} \frac{\partial u_i}{\partial x_j} = \tau_{11} \frac{\partial u_1}{\partial x_1} + \tau_{12} \frac{\partial u_1}{\partial x_2} + \tau_{13} \frac{\partial u_1}{\partial x_3} + \tau_{21} \frac{\partial u_2}{\partial x_1} + \tau_{22} \frac{\partial u_2}{\partial x_2} + \tau_{23} \frac{\partial u_2}{\partial x_3} + \tau_{31} \frac{\partial u_3}{\partial x_1} + \tau_{32} \frac{\partial u_3}{\partial x_2} + \tau_{33} \frac{\partial u_3}{\partial x_3} = \nabla \mathbf{U} : \boldsymbol{\tau}^T = \nabla \mathbf{U} : \boldsymbol{\tau}$$

因此, $k - \varepsilon$ 的矢量标识法形式为:

$$\rho \frac{\partial k}{\partial t} + \rho \mathbf{U} \cdot \nabla k - \nabla \cdot ((\mu + \mu_t/\sigma_k) \nabla k) = \nabla \mathbf{U} : \boldsymbol{\tau} - \rho \varepsilon$$

方程左侧分别为 k 的时间项、对流项以及扩散项, 右侧分别为产生项与耗散项。

2.3 偏导标识法

CFD 一个重要分支是高超音速动力学。在高超音速研究领域，大量的文章和书籍中均采用偏导标识法进行方程分析 [53]。在偏导标识法中，方程(2.3)可以表示为：

$$\mathbf{U}_t \equiv \frac{\partial \mathbf{U}}{\partial t} \quad (2.24)$$

同理，压力 p 对 x 的导数可以写为：

$$p_x \equiv \frac{\partial p}{\partial x} \quad (2.25)$$

一维连续性方程的偏导标识法可以写为：

$$\rho_t + (\rho u_1)_x = 0 \quad (2.26)$$

三维连续性方程的偏导标识法可以写为：

$$\rho_t + (\rho u_1)_x + (\rho u_2)_y + (\rho u_3)_z = 0 \quad (2.27)$$

方程(2.27)在矢量标识法中可以表示为：

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (2.28)$$

另外，在偏导标识法中，偏微分系统通常用下面的方程表示：

$$q_t + A q_x = 0 \quad (2.29)$$

其中 q 可以为标量，也可以为矢量。如果 q 为矢量，且具备 m 个分量，那么 A 则表示 $m \times m$ 阶矩阵。例如下面的气动声学方程

$$\begin{cases} p_t + \rho c^2 u_x = 0 \\ u_t + (1/\rho)p_x = 0 \end{cases} \quad (2.30)$$

其中未知量为 p 和 u ， ρ 和 c 为密度和声速，其可以写为下面更好理解的形式：

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \frac{\partial}{\partial t} \begin{pmatrix} p \\ u \end{pmatrix} + \begin{pmatrix} 0 & c^2 \rho \\ 1/\rho & 0 \end{pmatrix} \frac{\partial}{\partial x} \begin{pmatrix} p \\ u \end{pmatrix} = 0 \quad (2.31)$$

在这种情况下有：

$$q = \begin{bmatrix} p \\ u \end{bmatrix}, A = \begin{bmatrix} 0 & c^2 \rho \\ 1/\rho & 0 \end{bmatrix} \quad (2.32)$$

偏导标识法的这种写法可以更好的用于分析偏微分方程系统的数学特征。

2.4 运算符标识法

运算符标识法在文章中比较少见，其只不过是将数学符号翻译为相应的英文名字。例如，在笛卡尔坐标系下有

$$\operatorname{div} \mathbf{U} = \nabla \cdot \mathbf{U} \quad (2.33)$$

$$\operatorname{grad} p = \nabla p \quad (2.34)$$

$$\operatorname{laplacian} p = \operatorname{div} \operatorname{grad} p = \nabla \cdot (\nabla p) \quad (2.35)$$

$$\operatorname{curl} \mathbf{U} = \nabla \times \mathbf{U} \quad (2.36)$$

在这里需要注意的是，前文使用的 ∇ 算符在不同的坐标系下需要转换。但运算符标识法不需要转换。

2.5 $\nabla \mathbf{U}$ 的不同写法

现在回头看速度梯度的数学形式，其可以进一步的进行拆分。在之前的章节中，速度的梯度被写为 $\nabla \mathbf{U}$ ，其也可以写成导数的形式：

$$\nabla \mathbf{U} = \frac{d\mathbf{U}}{dx} \quad (2.37)$$

进一步的：

$$d\mathbf{U} = (\nabla \mathbf{U}) \cdot dx \quad (2.38)$$

其中

$$\begin{aligned} du_1 &= \frac{\partial u_1}{\partial x} dx + \frac{\partial u_1}{\partial y} dy + \frac{\partial u_1}{\partial z} dz \\ du_2 &= \frac{\partial u_2}{\partial x} dx + \frac{\partial u_2}{\partial y} dy + \frac{\partial u_2}{\partial z} dz \\ du_3 &= \frac{\partial u_3}{\partial x} dx + \frac{\partial u_3}{\partial y} dy + \frac{\partial u_3}{\partial z} dz \end{aligned} \quad (2.39)$$

其可以写为矩阵乘积的形式：

$$\begin{bmatrix} du_1 \\ du_2 \\ du_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_1}{\partial y} & \frac{\partial u_1}{\partial z} \\ \frac{\partial u_2}{\partial x} & \frac{\partial u_2}{\partial y} & \frac{\partial u_2}{\partial z} \\ \frac{\partial u_3}{\partial x} & \frac{\partial u_3}{\partial y} & \frac{\partial u_3}{\partial z} \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \quad (2.40)$$

也即：

$$\nabla \mathbf{U} = \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_1}{\partial y} & \frac{\partial u_1}{\partial z} \\ \frac{\partial u_2}{\partial x} & \frac{\partial u_2}{\partial y} & \frac{\partial u_2}{\partial z} \\ \frac{\partial u_3}{\partial x} & \frac{\partial u_3}{\partial y} & \frac{\partial u_3}{\partial z} \end{bmatrix} \quad (2.41)$$

可以看出，方程(2.41)中的速度梯度与方程(2.7)形式并不一样，二者互为转置。这是因为在这里牵涉到一个张量概念：分子布局（numerator layout）以及分母布局（denominator layout）。方程(2.41)中的速度梯度采用了分子布局。方程(2.7)采用的为分母布局。两种布局导致写法不同。维基百科表示目前的教材中确实存在这一问题 [7]。在本书以及开源软件 OpenFOAM 中，使用的是方程(2.7)的形式。

第三章 流体与 N-S 方程

3.1 导论

流体通常泛指气体和液体。不同于固体，流体在施加剪切力的时候，会发生形变。且流体没有固定的形态。流体在快速变形的时候，也会存在一种抵抗的力，这种抵抗的力一旦流动停止，也即消失。流体这种抗拒本身发生变形的力称之为粘性力。不同的流体存在不同的粘性力。有些流体的粘性很大，有些流体的粘性很小。大部分日常生活中接触的为牛顿流体，其形变率和剪切力呈现线性的关系。部分流体为非牛顿流体，其剪切率和形变率呈现非线性的关系。

标准状态下，一立方毫米气体包含大约 2.43×10^{16} 个分子。由于气体分子之间的碰撞非常频繁，有效地将气体分子合理的进行了分布。因此这些粒子群表现的并不是单一的粒子而是连续介质。标准情况下，气体分子每秒钟大约进行 10^9 次碰撞，每一次碰撞导致移动的距离非常小（约 5×10^{-8} 米）。前者即所谓的碰撞频率，后者即所谓的分子自由程。分子之间频繁的碰撞的宏观表现，就是存在压力。例如，将手竖直的放置在水中，手背和手面感受到的力就是水分子的压力。对于理想气体，宏观的压力 p 和微观量的关系可以表示为 $p = nkT$ ，其中 n 表示单个分子占据的平均体积的倒数，其单位为 m^{-3} 。这个平均体积主要是分子运动的平均范围。 k 为 Boltzmann 常数，其值为 $1.38066 \times 10^{-23} \text{ J/K}$ 。 T 表示温度。

液体不同于气体的典型特征为其对体积变化的巨大抵抗力。例如我们不能把一瓶矿泉水压缩成一瓶盖的大小。然而气体对体积变化的抵抗力是相对小的。例如我们可以轻易地用手对一个封闭的空的矿泉水瓶进行压缩。基于流体的密度是否为常数，进一步可以分为不可压缩流体和可压缩流体。液体通常考虑为不可压缩流体。虽然气体比液体的可压缩性高很多，但是是否考虑气体的可压缩性主要取决于具体情况。比如在空气动力学中，如果我们的气流速度很小，气体的可压缩性对汽车的设计影响较小，但是对于飞机，其速度有可能和声速接近或者大于声速，在这种情况下气体的可压缩性对飞机的设计非常重要。有些高压容器中，即使是液体，也应该考虑可压缩性。因为只有考虑了可压缩性，才能解释某些奇怪的物理现象，比如水锤。

特殊的情况下存在一些特殊的现象。比如宇宙中的空气接近真空，这种通常被称之为

为稀薄气体。稀薄气体由于气体的分子数量太少，因此气体的碰撞并不均衡。另一种为基本无碰撞的颗粒流，在颗粒不相互碰撞的情况下，会发生颗粒轨迹交叉，也就是一个粒子从左向右移动，另一个粒子从右向左移动，在某个位置，两个粒子彼此穿过并不发生碰撞。以上两种情况为一些极端的情况，这些情况并不能用 Navier-Stokes (N-S) 方程来描述。在这里需要表明的是 N-S 方程最重要的前提是连续介质假定。其可以理解为一个网格单元内存在大量的流体分子或固体颗粒。分子/颗粒之间碰撞均衡。这种情况下才能使用 N-S 方程来描述。在 N-S 方程不能描述的情况下，只能调用玻尔兹曼方程。本笔记仅仅考虑碰撞均衡的 N-S 方程¹。

CFD 是采用计算机求解控制流体流动的偏微分方程组的学科。以日常生活中的电风扇吹风为例。电风扇吹出来的空气的流动看不见、摸不着，但是能明显的感觉吹出来的风不一样。风扇转的越快，风越大越凉快，相对应的能耗也越多。工业上的设计目标就是用最小的能耗，吹出最大的风让一群人都很凉快。在这物理现象的背后，风扇吹出来的风的速度、压力、温度等均遵循相关的物理定律，这个物理定律可以写成数学中偏微分方程的形式，其主要有两个方程：一个是连续性方程（表明流动着的空气质量是守恒的），另一个是动量方程（表明空气的流动满足牛顿第二定律）。在考虑传热的情况下，还应该附加能量方程（表明能量守恒定律）。暂且可以这样理解：求解这两个关于时间的方程，就可以求出风扇吹来的风在某一时刻的流速。有了流速之后，同时通过风扇叶片表面的受力计算一下风扇的功率，就可以算出耗电量进行相关的设计。如果通过实验，工程师可以设计 100 个叶片的形状，检测风力的大小和能耗并做分析，取最优的叶片形状。通过 CFD，工程师可以生成 100 个叶片的几何模型，用计算机求解这 100 个几何模型，就可以计算出风力的大小和能耗进行分析。在工业应用中，一个机翼的实验工作可能要花费上百万元，一个化工反应器的中试实验工作可能需要上亿。但是通过 CFD 计算模拟，可能需要几万元就可以了。可见，CFD 的目标是让工程师从实验中解脱，并节省实验资金，进而进行工业设计。在有些情况下，可能会听到一些新词汇，如数值风洞、数值水池等，这实际上就是用 CFD 来模拟风洞内的流场、用 CFD 来模拟水池内水的流动。

问题在于，用于描述 CFD 的方程非常难解，甚至目前解析解（方程的准确解，比如 $x = -1$ 是 $x + 1 = 0$ 的解析解）的存在与否都无从考证。目前 CFD 在学术上的研究，即如何获得这些方程的解，这些解通常被称之为数值解。在理想的情况下，数值解无限趋近于解析解。然而，理想很丰满，现实很骨感。用户获得的数值解，总是和解析解存在或多或少的差距。

各种算法犹如各路神仙过河一样，总会通过各种奇葩的方法来获得 CFD 方程的数值解。比如 CFD 中有一种方法为光滑粒子法，其将流体看做是粒子，然后对这些粒子进行跟踪进行求解并获得数值解。最近，还有研究学者通过深度学习来拟合 CFD 的方程并求

¹若流体的特征长度 L 远远大于平均分子自由程 λ ，若流体的分子碰撞次数在标准状态下分子的碰撞频率左右（不远低于），则可用 N-S 方程进行描述。 λ 和 L 的比值为粒子克努森数。N-S 方程只在克努森数远远小于 1 的情况下适用。

解。目前 CFD 求解方法中最广泛的，莫过于有限体积法。目前主流的 CFD 软件，均使用的有限体积法进行求解。

有限体积法的思想也很简单。举例，如果要分析整个河流的流动状态、整个房间的热流密度、血管内的血流状态，有限体积法是通过从局部到整体的方法进行分析。例如我们把整个房间分成一个个的小格子，如果可以获得每个小格子内的流动速度，那么这些格子构成的整体就是整个房间的流动速度。格子的数量越多，格子的速度场越能代表整个房间的速度场。本文主要介绍有限体积法。

3.2 泰勒公式

在推导 N-S 方程之前，有必要介绍一下泰勒公式。假设有函数 $f(x)$ ，其用来表示整个 x 的型线。泰勒公式可以理解为函数 $f(x)$ 可以表示为若干个函数的加和，这里 $f(x)$ 为仅仅关于 x 的函数：

$$f(x) = f(x_0) + \frac{df(x)}{dx} \Big|_{x=x_0} (x - x_0) + \dots \quad (3.1)$$

当 $x = x_1$ 的时候，泰勒公式可用于求 $f(x_1)$ 的值：

$$f(x_1) = f(x_0) + \frac{df(x)}{dx} \Big|_{x=x_0} (x_1 - x_0) + \dots \quad (3.2)$$

现以 $f(x) = x^3$ 举例，其导数为 $\frac{df(x)}{dx} = 3x^2$ 。取 $x_0 = 2$ ，有 $f(x_0) = 8$ ，则有：

$$f(x_1) = 8 + 3 \times 2^2 (x_1 - 2) + \dots \quad (3.3)$$

如果 $x_1 = 3$ ，有 $f(x_1) = 20$ 。如果 $x_1 = 2.1$ ，有 $f(x_1) = 9.2$ 。可见 x_1 越趋向于 x_0 ，通过泰勒公式计算的值越精准。这在 CFD 中也反映了网格单元越小，结果越精准的特性。

泰勒公式也可用于多元函数，如有函数 $f(x, y, z, t)$ ，其位于 (x_1, y_1, z_1, t_1) 点的泰勒展开可表示为²：

$$\begin{aligned} f(x, y, z, t) &= f(x_1, y_1, z_1, t_1) + \frac{\partial f(x, y, z, t)}{\partial x} \Big|_{x_1, y_1, z_1, t_1} (x - x_1) \\ &\quad + \frac{\partial f(x, y, z, t)}{\partial y} \Big|_{x_1, y_1, z_1, t_1} (y - y_1) + \frac{\partial f(x, y, z, t)}{\partial z} \Big|_{x_1, y_1, z_1, t_1} (z - z_1) \\ &\quad + \frac{\partial f(x, y, z, t)}{\partial t} \Big|_{x_1, y_1, z_1, t_1} (t - t_1) + \dots \end{aligned} \quad (3.4)$$

3.3 流动模型

在谈具体的方程之前，首先需要介绍流动模型的概念。有限体积法是通过对每一个具体的流动模型进行分析，进而获得整体解的过程。在这里来理解流动模型可能会觉得比较

²注意这里 f 为多元函数，因此使用偏导数符号 ∂ 。更详细的关于泰勒公式的内容请参考文献 [12]。

抽象。但是在阅读完本文之后，回过头再来看流动模型，会有柳暗花明的感觉。

3.3.1 有限控制体模型

如图3.1所示，有限控制体其进一步可以分为空间位置固定的有限控制体，和随流线运动的有限控制体。空间位置固定的有限控制体可以理解为一个空旷的房间（当然控制体的体积要小的多），其会流入、流出某些物理量（如质量、能量等），这些量的产生和消失符合物理学基本定律。举例，这个房间内部充满了人（不能再容纳更多的人），如果挤进来 10 个人，那么必然会挤出去 10 个人。即为质量守恒。随流线运动的有限控制体可以理解为随风飘扬的一个气球，气球内部存在大量的空气，且空气和外界不发生交换（质量不变）。这个气球的运动当然满足物理的基本定律。

在后续的讨论中，可以看出从有限控制体推导的方程为积分形式，这是因为控制体相对下文讨论的微团模型是更大的。且若采用空间位置固定的控制体，方程为守恒型方程。若采用随流线运动的控制体，方程为非守恒型。要透彻的理解 CFD 中守恒和非守恒的概念，目前来讲并不容易。在此只需要知道有不同的方程的概念就可以。

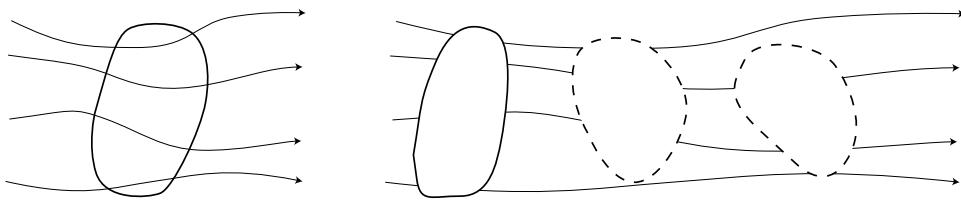


图 3.1: 左图：空间位置固定的有限控制体，流体流入控制体后继续流出。右图：空间位置移动的有限控制体，其沿着流体的流线进行移动。

3.3.2 无穷小微团模型

如图3.2所示³，类似有限控制体，无穷小微团也可以进一步分为空间位置固定的无穷小微团，和随流线运动的无穷小微团。无穷小微团和有限控制体的重要区别在于无穷小微团的体积、质量等变量可以用数学中的 dV 、 dm 来表示。有限控制体的体积要比无穷小微团要大，也即有限控制体中包含了大量的 dV ，需要用积分来表示。从数学的角度出发， dV 表示这个无穷小微团的体积足够小。需要注意的是，为了使得 CFD 方程适用于连续介质，这个 dV 还要包含一定量的流体分子⁴。正因为无穷小微团的各个属性可以用数学中的 $d...$ 来表示，同时有限控制体包含了大量的流体微团，这也就体现了从有限控制体推导出来的

³ 图中仅仅画出一个无穷小微团，若存在多个无穷小微团，每个无穷小微团应彼此相连并填充整个计算空间。也就是说整个计算域内应充满网格单元。

⁴ N-S 方程并不是普适性的方程，更为底层的为玻尔兹曼方程。只有满足连续性假定，才能使得 BGK 假定成立，进而将玻尔兹曼方程简化为 N-S 方程。例如对于稀薄气体，由于并不满足连续性假定，因此 N-S 方程并不适用，只能通过数值方法求解玻尔兹曼方程（如矩方法 [84]、动理学格式方法 [97]、蒙特卡洛法 [17] 等）。对于多相流领域同样具有类似的情况，比如颗粒轨迹交叉问题就不满足连续介质假定 [65]。

方程为积分形式，从无穷小微团推导出来的方程为微分形式。现在以空间位置固定的有限

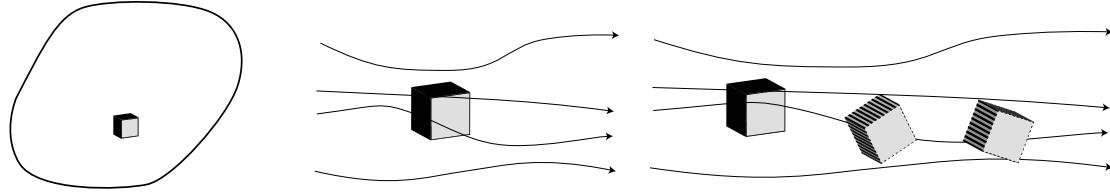


图 3.2: 左图: 有限控制体 (大圈) 和无穷小微团 (小正方体)。中图: 空间位置固定的无穷小微团, 流体流入微团后继续流出。右图: 空间位置移动的无穷小微团, 其沿着流体的流线进行移动。

控制体或流体微团为例。我们要了解一个房间内流体的流动状况。如果房间内每个有限控制体或流体微团的流动均已知, 那么我们就可以获得这个房间完整的流体流动图像。更形象的, 我们把房间分为 9 个位置并用 9 个有限控制体或流体微团来表示, 如果采用实验的方法对这 9 个有限控制体或流体微团检测速度, 那么我们就可以认为这 9 个速度的场即为整个房间的速度场。如果我们把这个房间分为 N 块, 并用实验的方法检测这 N 个位置的速度, 我们会得到一个更精确的速度场。实际上, 房间的 N 块, 即可以近似的理解为 CFD 中网格的概念。CFD 要做的, 就是通过计算机, 求解这些划分的小块上的物理量。

3.3.3 物质导数

物质导数就是数学中的全导数。理解物质导数需要借助上文中提及的流动模型。现选取随流线运动的无穷小微团上的温度 $T(x, y, z, t)$ 进行分析。需要注意的是, 在上文中指出无穷小微团上的变量可以微分算符 d 来表示, 这里的温度并没有表示为 dT , 其原因在于温度并不是一个随物质的量变化而变化的量。不管存在多少的分子, 温度均为 T 。类似的这种量, 在 CFD 被称之为强度量。反之, 质量、体积等为广度量。

在 t_1 时间点, 流体微团的位置在 (x_1, y_1, z_1) , 温度为 $T_1(x_1, y_1, z_1, t_1)$ 。随着微团的移动, 在 t_2 时刻, 位置变为 (x_2, y_2, z_2) , 同时温度为 $T_2(x_2, y_2, z_2, t_2)$ 。既然温度为时间和位置的函数, 则可参考方程(3.4)对温度进行泰勒展开:

$$T_2 = T_1 + \frac{\partial T}{\partial x} \Big|_{x=x_1} (x_2 - x_1) + \frac{\partial T}{\partial y} \Big|_{y=y_1} (y_2 - y_1) + \frac{\partial T}{\partial z} \Big|_{z=z_1} (z_2 - z_1) + \frac{\partial T}{\partial t} \Big|_{t=t_1} (t_2 - t_1) \quad (3.5)$$

对方程 (3.5) 左右两边除以 $t_2 - t_1$ 有:

$$\frac{T_2 - T_1}{t_2 - t_1} = \frac{\partial T}{\partial x} \Big|_{x=x_1} \frac{x_2 - x_1}{t_2 - t_1} + \frac{\partial T}{\partial y} \Big|_{y=y_1} \frac{y_2 - y_1}{t_2 - t_1} + \frac{\partial T}{\partial z} \Big|_{z=z_1} \frac{z_2 - z_1}{t_2 - t_1} + \frac{\partial T}{\partial t} \Big|_{t=t_1} \quad (3.6)$$

接下来定义温度 T 在 (x_1, y_1, z_1) 点的物质导数为移动的无穷小微团通过 (x_1, y_1, z_1) 点的时候, 无穷小微团针对时间的温度瞬时变化率, 即:

$$\left. \frac{DT}{Dt} \right|_{x=x_1, y=y_1, z=z_1} = \lim_{t_2 \rightarrow t_1} \frac{T_2 - T_1}{t_2 - t_1} \quad (3.7)$$

依据速度的定义，有 $\lim_{t_2 \rightarrow t_1} \frac{x_2 - x_1}{t_2 - t_1} = u$, $\lim_{t_2 \rightarrow t_1} \frac{y_2 - y_1}{t_2 - t_1} = v$, $\lim_{t_2 \rightarrow t_1} \frac{z_2 - z_1}{t_2 - t_1} = w$, 代入到方程 (3.6) 有 (x_1, y_1, z_1) 点的物质导数:

$$\frac{DT}{Dt} \Big|_{x=x_1, y=y_1, z=z_1} = \frac{\partial T}{\partial x} \Big|_{x=x_1} u + \frac{\partial T}{\partial y} \Big|_{y=y_1} v + \frac{\partial T}{\partial z} \Big|_{z=z_1} w + \frac{\partial T}{\partial t} \Big|_{t=t_1} \quad (3.8)$$

对于任意一点，有

$$\frac{DT}{Dt} = \frac{\partial T}{\partial x} u + \frac{\partial T}{\partial y} v + \frac{\partial T}{\partial z} w + \frac{\partial T}{\partial t} \quad (3.9)$$

方程 (3.9) 即为温度 T 在笛卡尔坐标系下的物质导数的定义。可见，物质导数即为数学上物理量对时间的全导数。

参考内积的数学定义，方程 (3.9) 可写为⁵:

$$\frac{DT}{Dt} = \frac{\partial T}{\partial t} + \mathbf{U} \cdot \nabla T \quad (3.10)$$

上述以温度 T 举例说明物质导数的由来。类似的，有速度 \mathbf{U} 的物质导数:

$$\frac{D\mathbf{U}}{Dt} = \frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} \quad (3.11)$$

速度的物质导数表示流体微团的加速度。除此之外，任何的流场数据都可以用物质导数来表示，如:

$$\frac{D(\cdot)}{Dt} = \frac{\partial(\cdot)}{\partial t} + u \frac{\partial(\cdot)}{\partial x} + v \frac{\partial(\cdot)}{\partial y} + w \frac{\partial(\cdot)}{\partial z} = \frac{\partial(\cdot)}{\partial t} + \mathbf{U} \cdot \nabla(\cdot) \quad (3.12)$$

方程 (3.12) 中的 $\frac{\partial(\cdot)}{\partial t}$ 又称为局部导数，第二项 $\mathbf{U} \cdot \nabla(\cdot)$ 又称之为对流导数。需要注意的是，如果求解器不调用拉格朗日粒子，方程(3.10)左边的物质导数项并不能够直接求出，其只能通过方程右边计算而来。

方程(3.10)可以理解为:

$$\underbrace{\frac{D}{Dt}}_{\text{Lagrangian}} = \underbrace{\frac{\partial}{\partial t} + \mathbf{U} \cdot \nabla}_{\text{Euler}} \quad (3.13)$$

其中 $\frac{\partial}{\partial t}$ 表示对时间的变化率， $\mathbf{U} \cdot \nabla$ 表示对流的变化率。在这里增加一个对欧拉以及拉格朗日的解释，如图3.3所示，欧拉方法描述一个固定点的变量的时间变化值，如图中方框内的变量。拉格朗日方法描述一个移动的粒子的变量的时间变化（如图中红色粒子的移动）⁶。

⁵在这里要注意， ∇ 算子一般也可以写为:

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)^T$$

因此可以定义操作符 $\mathbf{U} \cdot \nabla$ 为 \mathbf{U} 与 ∇ 算子的内积:

$$\mathbf{U} \cdot \nabla = u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} + w \frac{\partial}{\partial z}$$

这样， $\mathbf{U} \cdot \nabla T$ 也可以写成 $(\mathbf{U} \cdot \nabla)T$ 。同理:

$$(\mathbf{U} \cdot \nabla)\mathbf{U} = \begin{pmatrix} u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \\ u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \\ u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \end{pmatrix}$$

⁶在守恒法则研究领域，物质导数也表示沿着特征线的变量时间变化率。

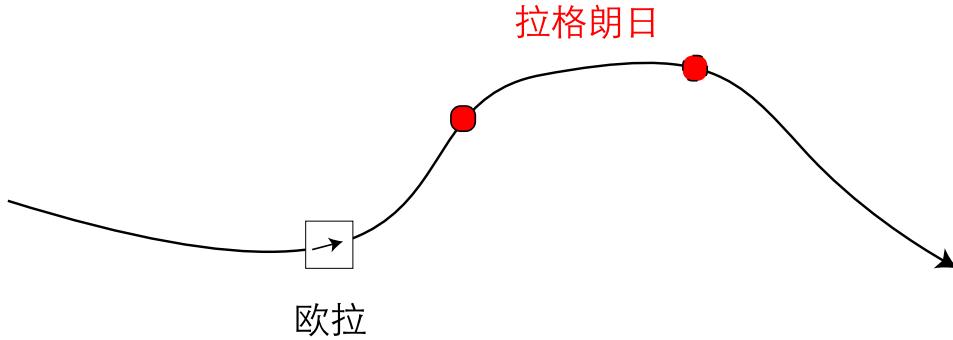


图 3.3: 欧拉与拉格朗日的描述。

以温度举例，欧拉方法描述方框内温度随着时间的变化，拉格朗日方法描述移动的粒子的温度在不同的位置处随时间产生的变化。另外，拉格朗日方法的自变量通常为 t ，欧拉方法的自变量通常为 (x, y, z, t) 。例如，有粒子随时间变化的位置矢量定义为：

$$\mathbf{S} = (x, y, z) = (t^2, 2t, 0) \quad (3.14)$$

则有

$$x = t^2, y = 2t, z = 0 \quad (3.15)$$

那么拉格朗日框架下的粒子的速度矢量通常表示为：

$$\mathbf{U}(t) = (2t, 2, 0) \quad (3.16)$$

欧拉框架下的速度矢量通常表示为：

$$\mathbf{U}(x, y, z, t) = (2t, 2, 0) = (2\sqrt{x}, 2, 0) \quad (3.17)$$

另外从方程(3.17)可以看出，针对本示例在固定的网格位置处的速度不变。从物理意义来讲，即粒子均以同样的速度通过某固定网格点。

3.4 连续性方程

CFD 的控制方程无论具有什么形式，都是建立在流体力学基本控制方程：连续性方程、动量方程、能量方程的基础之上⁷。他们主要来源于质量守恒定律、牛顿第二定律、能量守恒定律。N-S 方程最基本的假设为：把流体看做是连续介质。当从宏观尺度来分析的时候（比如一微米），流体的随机分子运动可以忽略。这样，就可以描述流体的一些宏观物理量如速度、压力、密度和温度等。因为这些物理量可以看做大量分子的平均。本节一步一步地对连续性方程进行推导，旨在让同学们了解 CFD 中基本控制方程的来源，做到在 CFD

⁷再一次强调，这里假定流动为平衡流，即分子碰撞是频繁的。

模拟的时候有理有据。复杂的方程离散、对流格式、双曲特征以及激波等将在 GCFD 课程中进行讲述。

3.4.1 微分形式

连续性方程可以在上文中的 4 种流动模型基础上进行推导而来，本节从空间位置固定的无穷小微团角度出发，对连续性方程进行推导。这种推导形式是最容易理解的，只涉及到高等数学中的泰勒方程。从空间位置固定的角度来分析的方法，通常被称之为欧拉观点。欧拉观点下的质量守恒意味着位置固定无穷小微团质量的变化 = 流入无穷小微团质量 - 流出无穷小微团质量。也即为：位置固定无穷小微团质量的变化率 = (流入无穷小微团质量 - 流出无穷小微团质量) 的变化率。图3.4为一个质量守恒的实例。

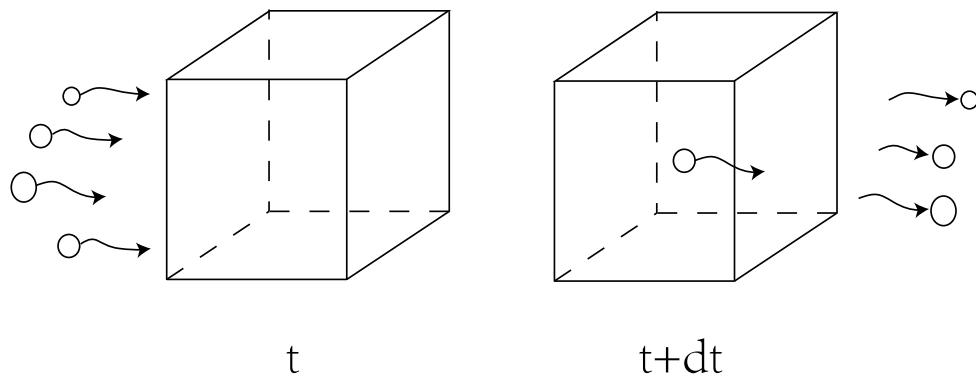


图 3.4: 质量守恒实例。在 t 时刻，有 4 单位的流体要流入无穷小微团（左图）。在 $t + dt$ 时刻，有 3 单位的流体流出了无穷小微团（右图）。那么这个无穷小微团的变化，即为增加了 1 单位的流体（右图无穷小微团内增加 1 单位流体）。

下面首先看无穷小微团的质量的变化率。如上文所述，流体微团的体积可以表示为 $dV = dx dy dz$ 。同时，有密度 ρ ，则对应的无穷小微团质量为

$$dm = \rho dx dy dz \quad (3.18)$$

依据质量变化率的定义（质量的变化除以时间的变化），其可以表示为

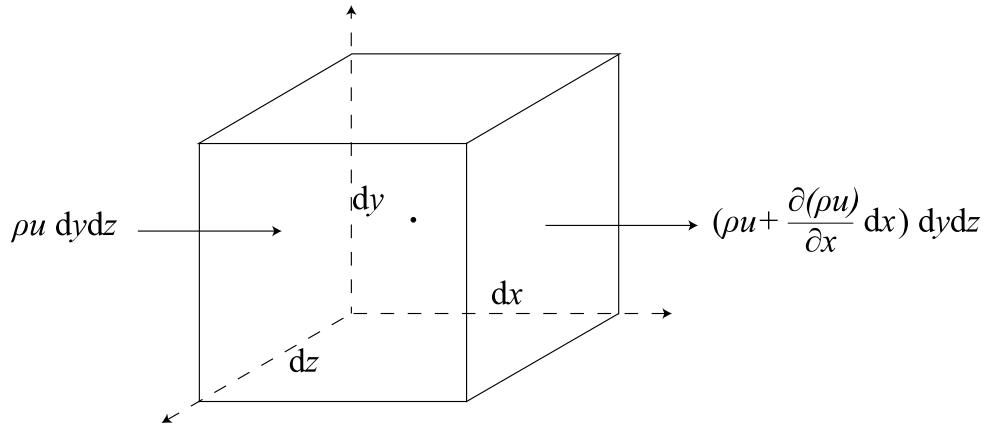
$$\frac{\partial dm}{\partial t} = \frac{\partial \rho dx dy dz}{\partial t} = \frac{\partial \rho}{\partial t} dx dy dz \quad (3.19)$$

下面考虑无穷小微团的流入和流出。在这里介绍一下泰勒方程的另一个形式：如果对于一个函数， x_0 点的值为 $f(x_0)$ ，那么 $x_0 + h$ 的值可以通过下述方程求得⁸：

$$f(x_0 + h) = f(x_0) + f'(x_0)h + \mathcal{O}(h) \approx f(x_0) + f'(x_0)h \quad (3.20)$$

现在看图3.5中的无穷小微团，定义立方体左侧单位面积的质量流量为 ρu ，其表示单位时

⁸ 在这里假定函数是连续可微的。然而对于某些特殊问题这种假定并不成立（例如存在激波的问题）。这种情况下需要从积分的角度推导连续性方程（参见下节）。

图 3.5: 无穷小微团 x 方向的通量。

间内流入单位面积的质量（注意其中的单位时间和单位面积）。同时立方体左侧的面积为 $dydz$ ，因此有单位时间内流入的质量为

$$\rho u dydz \quad (3.21)$$

对于立方体右侧的面，在已知立方体左侧面 ρu 的定义的时候，右侧可以通过泰勒方程求出，即单位时间内流出立方体右侧的单位面积的质量为 $\rho u + \frac{\partial \rho u}{\partial x} dx$ 。同样的，立方体右侧的面的面积为 $dydz$ ，因此有单位时间内流出的质量约为⁹

$$\left(\rho u + \frac{\partial \rho u}{\partial x} dx \right) dydz \quad (3.22)$$

结合无穷小微团单位时间的流出质量，即方程(3.22)，和无穷小微团单位时间的流入质量，即方程(3.21)，有单位时间的 x 方向的净质量变化率为

$$\rho u dydz - \left(\rho u + \frac{\partial \rho u}{\partial x} dx \right) dydz = - \frac{\partial \rho u}{\partial x} dx dydz \quad (3.23)$$

同理，有 y 方向的净质量变化率为

$$-\frac{\partial \rho v}{\partial y} dx dy dz \quad (3.24)$$

z 方向的净质量变化率为

$$-\frac{\partial \rho w}{\partial z} dx dy dz \quad (3.25)$$

将各个方向的质量变化率加和有

$$-\frac{\partial \rho u}{\partial x} dx dy dz - \frac{\partial \rho v}{\partial y} dx dy dz - \frac{\partial \rho w}{\partial z} dx dy dz = - \left(\frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} \right) dx dy dz \quad (3.26)$$

⁹ 其中忽略了无穷小量 $\mathcal{O}(dx)$ 。

同时，结合方程(3.19)，有

$$\frac{\partial \rho}{\partial t} dx dy dz = - \left(\frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} \right) dx dy dz \quad (3.27)$$

也即为连续性方程：

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} = 0 \quad (3.28)$$

方程(3.28)在 CFD 中至关重要，其直接关系到压力泊松方程的导出。在这里，同学们应该尝试去多次的理解连续性方程推导的本源，即连续性方程表示的是质量守恒。

3.4.2 积分形式

上文中从空间位置固定的无穷小微团推导的连续性方程是最容易理解的。下面考虑从空间位置固定有限控制体推导的连续性方程。本节涉及到高等数学的散度定律。若难以理解可跳过本节且不失去连续性。本节只不过是另一种推导连续性方程的途径。要提前强调

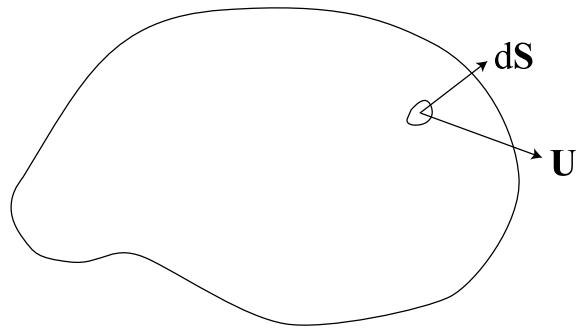


图 3.6: 有限控制体上的速度和面矢量。

的是，有限控制体和无穷小微团的关系可以理解为有限控制体包含着很多的无穷小微团（如图3.2左图所示）。因此，有限控制体中的量都需要用积分来进行计算。对于空间位置固定的有限控制体，其质量可通过下面的方程进行计算：

$$\int \rho dV \quad (3.29)$$

实际上就是高等数学中，质量的计算方程。其对应的质量变化率为

$$\frac{\partial}{\partial t} \int \rho dV \quad (3.30)$$

下面考虑空间位置固定的有限控制体由于流入流出引起的质量变化。图3.6中的通量可以定义为

$$\rho \mathbf{U} \cdot d\mathbf{S} \quad (3.31)$$

那么整个有限控制体上的净通量可以表示为

$$-\int_S \rho \mathbf{U} \cdot d\mathbf{S} \quad (3.32)$$

其中负号的引入是因为考虑的为流入减去流出的通量（而不是流出减去流入）¹⁰。结合方程(3.32)和方程(3.30)，有

$$\frac{\partial}{\partial t} \int_V \rho dV + \int_S \rho \mathbf{U} \cdot d\mathbf{S} = 0 \quad (3.33)$$

也即¹¹

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{U} = 0 \quad (3.34)$$

方程(3.34)即为通过空间位置固定的有限控制体推导的连续性方程，其和(3.28)是等价的。在这里需要注意的是，从方程(3.33)推导至(3.34)的时候，需要假定函数是光滑的连续的。因此积分形式的方程(3.33)是比微分形式的方程(3.34)更基础的控制方程。

方程(3.33)是一种形式的积分形式连续性方程。对其对时间做积分，有另一种形式的积分形式的连续性方程¹²：

$$\int_t^{t+\Delta t} \frac{\partial}{\partial t} \int_V \rho dV dt + \int_t^{t+\Delta t} \int_S \rho \mathbf{U} \cdot d\mathbf{S} dt' = 0 \quad (3.35)$$

$$\int_V \rho(t + \Delta t) dV - \int_V \rho(t) dV = - \int_t^{t+\Delta t} \left(\int_S \rho(t') \mathbf{U}(t') \cdot d\mathbf{S} \right) dt' \quad (3.36)$$

方程(3.36)的左边表示在体积 V 上，时间间隔 Δt 上质量的变化，右侧则考虑通量的变化形式，同样表示体积 V 上，时间间隔 Δt 上质量的变化。方程(3.36)与(3.33)是相同的，且是精准的。如果知道方程(3.36)右侧 $\rho(t') \mathbf{U}(t')$ 的准确值，那么结果就是精准的。方程(3.36)主要用于双曲系统来推导高阶通量格式（如 HLL 格式，其通过 V 将黎曼扇区域进行包围并假定 $\rho(t') \mathbf{U}(t')$ 的近似值即可获得网格面上的通量）。

3.4.3 通量与速度散度的物理意义

在这里有必要介绍通量的概念。因为初学者经常将对流项引申的量理解为通量。然而 CFD 控制方程中离散后很多项都可以称之为通量，比如对流项 $\nabla \cdot (\mathbf{U} \mathbf{U})$ 在做线性化后会出现对流通量。扩散项 $\nabla \cdot (\mu \nabla \mathbf{U})$ 离散后同样会出现扩散通量。进一步的，通量可区分为体积通量和质量通量。体积通量的物理意义为单位时间内流经某个网格单元面流体的体积，单位为 m^3/s 。质量通量的物理意义为单位时间内流经某个网格单元面流体的质量，单位为 kg/s 。不同的场合通常选取不同的通量。对于不可压缩流体，通常选取体积通量，对于可

¹⁰ 流入的通量为负。

¹¹ 其中要调用高斯定律 $-\int \rho \mathbf{U} \cdot d\mathbf{S} = -\int \nabla \cdot \rho \mathbf{U} dV$

¹² 右侧的时间 t 以及 $t + \Delta t$ 表示积分限，为固定值， t' 才是真正的积分变量，即时间

压缩流体，通常选取质量通量。从数学的角度来理解更为简单：速度 \mathbf{U} 乘以面积 \mathbf{S} 就是体积通量，即每单位时间流经某个网格单元面多少立方米的流体：

$$\phi_f = \mathbf{U}_f \cdot \mathbf{S}_f \quad (3.37)$$

其中 ϕ_f 表示网格单元面 \mathbf{S}_f 上的通量， \mathbf{S}_f 也被称之为面矢量， \mathbf{U}_f 为定义在网格单元面上的速度。图3.7中左中右三个网格单元的最上方的网格单元面对应的三个面矢量，方向相同，大小不同。面矢量 \mathbf{S}_f 的大小可通过下式计算：

$$|\mathbf{S}_f| = \sqrt{S_x^2 + S_y^2 + S_z^2} \quad (3.38)$$

其中 S_x, S_y, S_z 表示面矢量 \mathbf{S}_f 的分量。

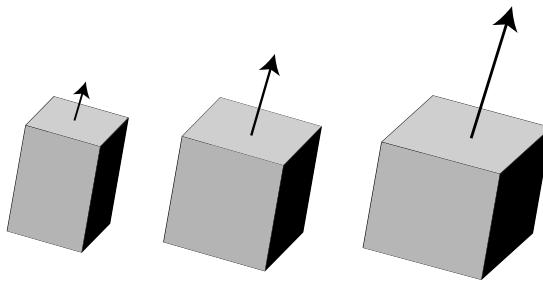


图 3.7: 上图中表示了相同方向，不同大小（箭头长度不同）的面矢量。

CFD 中另外一个重要的概念是面法向矢量 \mathbf{n}_f ，其用于定义面的法向，面法向矢量的特点是其模均为 1，计算公式为：

$$\mathbf{n}_f = \frac{\mathbf{S}_f}{|\mathbf{S}_f|} \quad (3.39)$$

如图3.8所示，在一个网格的边界区域，通常由若干个网格单元面构成，每个网格单元面则分别对应一个面矢量和面法向矢量。

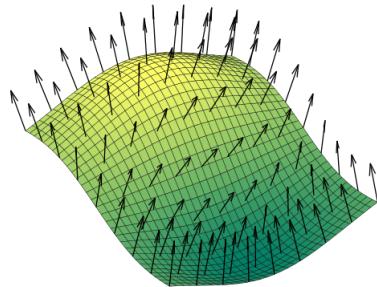


图 3.8: 任意的弯曲几何的面法向矢量。

除了边界面，网格的内部面也存在面矢量。图3.9表示为网格内部面的面矢量定义。图中的 \mathbf{S} 为两个网格连接面的矢量，其垂直于网格面，大小等于面的面积。由宿主网格（图中 P 点）指向相邻网格（图中 N 点）。同时可以看出，连接 P 网格体心和 N 网格体心的矢量方向和 \mathbf{S}_f 的方向并不相同。这会导致梯度的计算差异，也即 CFD 中的非正交修正算法，在此不做介绍。回到通量的计算公式，有：

$$\phi_f = \mathbf{U}_f \cdot \mathbf{S}_f = |\mathbf{U}_f| |\mathbf{S}_f| \cos\theta \quad (3.40)$$

其中 θ 为面矢量和面速度矢量的夹角。很明显通量的大小取决于夹角的大小。图3.10形象的表示了这一特性。左图中速度矢量和面矢量方向平行，则其中的虚线表示单位时间内通过面的体积。中图中的速度矢量和面矢量存在一定的夹角，因此体积有所减小。右图中速度矢量和面矢量垂直，因此没有任何流体通过这个面，因此通量为零。

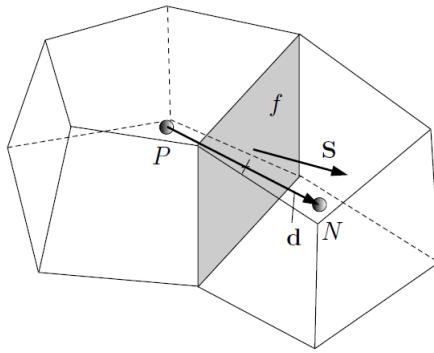


图 3.9: 网格单元内部面面矢量与网格体心点连线示意图。

附加几点有关通量的总结和评价：

- 通量的计算不仅和面的大小以及速度的大小有关，并且和面的方向以及速度的方向有很大关系；
- 有限体积法中的守恒表示为通量的守恒，然而通量上的面速度和网格单元体心处的速度是有区别的；
- 本文中面速度均为给定，但是在 CFD 计算中，一些通量需要从体心速度插值而来，一些通量需要通过压力速度耦合算法计算而来；
- 对于图3.9中的非矩形网格以及高偏斜网格，需要对应的非正交修正算法，或者采用最小二乘法计算梯度；

下面介绍速度散度的物理意义，需要一些高等数学中积分的内容，若难以理解可不失去连续性的略过后续本节内容¹³。在考虑体积通量 ϕ_f （单位时间内流经某个面的流体体积）

¹³ 但3.4.4节以此为基础展开。

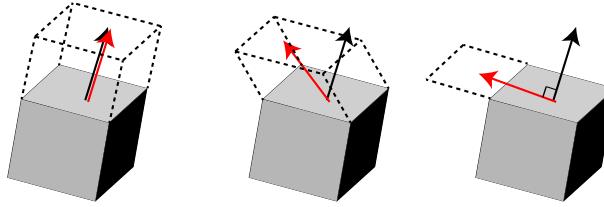


图 3.10: 不同速度矢量对应的通量大小, 其中黑色箭头表示面矢量, 红色箭头表示速度矢量。

的物理意义的基础上。参考图3.10, 如果在体积通量的基础上乘以时间 Δt , 则表示体积为 V 的流动控制体, 由于其中的面 $d\mathbf{S}$ 以 \mathbf{U} 的速度移动, 在 Δt 时间内体积的变化量:

$$dV_f = \phi_f \Delta t = \mathbf{U}_f \cdot d\mathbf{S}_f \Delta t \quad (3.41)$$

因此, 整个流动控制体的体积变量需要在方程(3.41)的基础上做积分:

$$\Delta V = \int_V dV = \int_S \mathbf{U}_f \cdot d\mathbf{S}_f \Delta t \quad (3.42)$$

体积 V 的物质导数定义为

$$\frac{DV}{Dt} = \frac{\Delta V}{\Delta t} = \int_S \mathbf{U}_f \cdot d\mathbf{S}_f \quad (3.43)$$

依据高等数学的散度定律有

$$\int_S \mathbf{U}_f \cdot d\mathbf{S}_f = \int_V \nabla \cdot \mathbf{U} dV \quad (3.44)$$

即

$$\frac{DV}{Dt} = \int_V \nabla \cdot \mathbf{U} dV \quad (3.45)$$

方程(3.45)对体积 V 做积分。如果考虑体积 V 足够小, 即 $V \rightarrow \delta V$, 方程(3.45)在此 δV 可以表示为:

$$\frac{D(\delta V)}{Dt} = \int_{\delta V} (\nabla \cdot \mathbf{U}) dV \quad (3.46)$$

由于体积 δV 足够小, $\nabla \cdot \mathbf{U}$ 在此 δV 上都相等可以提出, 即

$$\int_{\delta V} (\nabla \cdot \mathbf{U}) dV = \nabla \cdot \mathbf{U} \delta V \quad (3.47)$$

结合方程(3.46)和(3.47)有:

$$\nabla \cdot \mathbf{U} = \frac{1}{\delta V} \frac{D\delta V}{Dt} \quad (3.48)$$

即速度散度的物理意义: 每单位体积流动着的控制体体积随时间的变化率。

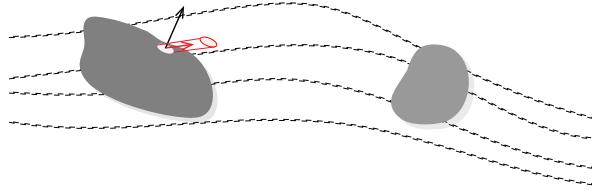


图 3.11: 流动的控制体由于速度的变化导致形状体积的改变, 左图中面 $d\mathbf{S}$ 以 \mathbf{U} 的速度移动, 在 Δt 时间间隔内引起的体积变化为圆柱体体积。右图为最终的控制体形状。

3.4.4 拉格朗日观点

上文是从空间位置固定的有限控制体以及无穷小微团下推导的连续性方程的。下面讨论另一种推导形式: 从空间位置移动的无穷小微团推导的连续性方程。这种方式偏向于拉格朗日思想。若难以理解可跳过本节且不失连续性。本节只不过是另一种推导连续性方程的途径。

在拉格朗日框架下, 随流线运动无穷小微团质量的变化 = 0。参考前文物质导数的定义, 即

$$\frac{D}{Dt} dm = 0 \quad (3.49)$$

其中采用 dm 是因为跟踪的为无穷小微团。依据物质导数的定义, 方程(3.49)可以化简为

$$\frac{D}{Dt} dm = \frac{D(\rho dV)}{Dt} = \rho \frac{D(dV)}{Dt} + dV \frac{D\rho}{Dt} = 0 \quad (3.50)$$

即

$$\frac{\rho}{dV} \frac{D(dV)}{Dt} + \frac{D\rho}{Dt} = 0 \quad (3.51)$$

依据上一节介绍的物质导数和速度散度的物理意义, 有

$$\frac{1}{dV} \frac{D(dV)}{Dt} = \nabla \cdot \mathbf{U} \quad (3.52)$$

代入到方程(3.50), 有

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{U} = 0 \quad (3.53)$$

3.4.5 连续性方程小结

方程(3.28)、(3.34)以及(3.53)均表示连续性方程, 但各自的推导采用了不同的流动模型。有的模型出发点和推导比较容易理解, 有的方法比较复杂。这些方程可以相互的转换。且各种教材中书写并不统一。例如, 下面的方程均为采用不同书写方式书写的连续性方程:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (3.54)$$

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \mathbf{U}) = 0 \quad (3.55)$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \quad (3.56)$$

$\nabla \cdot$ 和 div 在笛卡尔网格下等价，均表示散度操作。方程(3.56)则是另一种书写方式。初学 CFD 可能对方程中的这些符号感到很生疏，在此记住下面最根本的的连续性方程形式即可：

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} = 0 \quad (3.57)$$

3.5 动量方程

本节讨论动量方程。动量方程依然可以通过上面四种不同的流动模型进行推导。由于篇幅所限本节不再一一推导，仅介绍一种较为容易理解的推导方法，即从空间位置移动的无穷小微团进行分析。在推导之前，需要介绍一些基础的知识。

3.5.1 受力分析

对动量方程进行推导，就需要对无穷小微团进行受力分析。正是因为流体的受力，才会引致流体的流动。无穷小微团的受力可区分为体积力和表面力。表面力为作用在无穷小微团面上的力，如压力、表面张力，海洋表面上的风也可以认为是表面力。体积力作用在无穷小微团全部的体积上（不仅表面有，体积内也有），例如重力、电磁力，或者一切引起旋转的力（科氏力）。在这些力中，最重要的表面力为压力和应力（后续会对应力进行介绍），最重要的体积力为重力（若不考虑其他源项力）¹⁴。

首先看压力，如图3.12所示，CFD 中的压力用 p 来表示（其实际为压强），且这个压力表示静压。其作用主要导致流体的压缩和膨胀，而不是切应力引致的变形。另外，压力为一种正应力（不同于下文中要介绍的切应力），其永远作用于无穷小微团的面的垂直的方向。同时，如图3.13所示，无穷小微团还受到剪切应力的作用。其主要体现为流体的粘

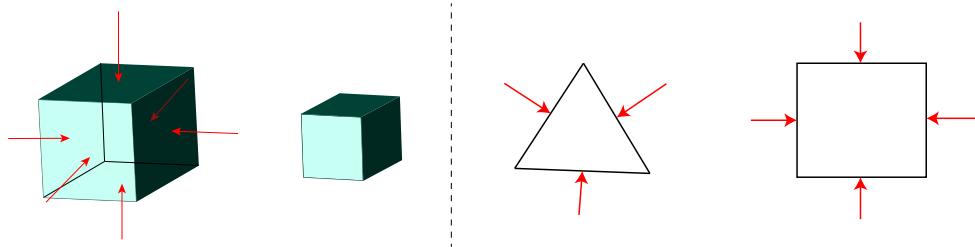


图 3.12: 左图：无穷小微团由于压力的作用导致体积的压缩。右图：三角形网格和四边形网格受到的垂直的压力。

性。可以很明显的注意到，蜂蜜要比水要粘。这就是因为要获得同样大的形变率，施加给蜂蜜的剪切应力需要更大。剪切应力的作用导致流体产生形变。这个应力不是和形变成正比，而是和形变的速率成正比。也就是说，应力越大，形变的速度越快。

¹⁴CFD 中的一些体积力与表面力需要特殊处理，否则会引起数值震荡，感兴趣的读者可以阅读相关文献了解相关信息 [100, 101]。

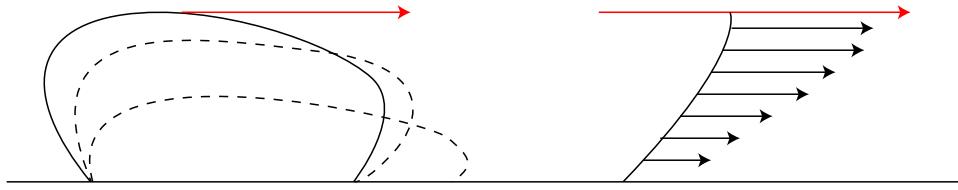


图 3.13: 左图: 一块蜂蜜由于剪切应力的作用导致的变形。右图: 流体由于上方剪切力作用导致的流速区别, 壁面附近速度较小。红色的箭头表示剪切应力, 黑色的箭头表示速度大小和方向。

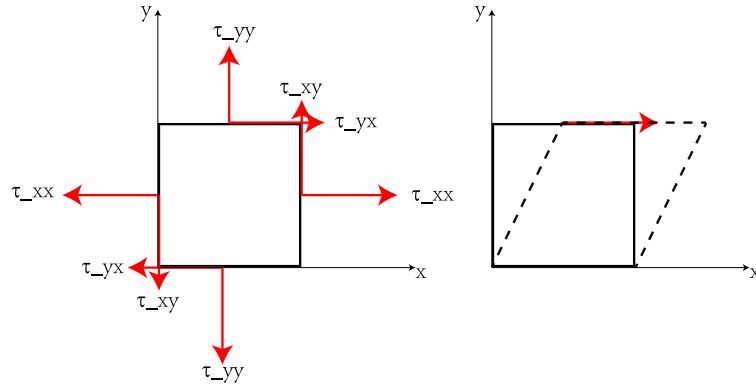


图 3.14: 左图: 一个四边形无穷小微团所受的剪切应力。右图: 一个四边形有限控制体受到剪切应力导致形变。

如图3.14左图所示, 一个二维四边形无穷小微团所受的剪切应力分量分别为

$$\tau_{yy}, \tau_{yx}, \tau_{xx}, \tau_{xy} \quad (3.58)$$

其中的 τ 表示剪切应力, 第一个下标表示作用于与某方向垂直的平面, 第二个下标表示力的方向。比如 τ_{yy} 表示作用于和 y 垂直的平面的剪切应力分量, 其方向为 y 方向。 τ_{yx} 表示作用于和 y 垂直的平面的剪切应力分量, 其方向为 x 方向。图3.14右图表示了流体由于剪切应力作用发生的形变。图3.15则是三维的无穷小微团受到的剪切应力。在三维的情况下, 剪切应力的分量可以表示为

$$\tau_{xx}, \tau_{xy}, \tau_{xz}, \tau_{yx}, \tau_{yy}, \tau_{yz}, \tau_{zx}, \tau_{zy}, \tau_{zz} \quad (3.59)$$

其通常写成下面的形式:

$$\begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} \quad (3.60)$$

同时要注意, 剪切应力张量为一个对称的量。只有其为对称的(净扭矩为零), 才能防止产生旋转。

在这里, 读者应该知道方程(3.60)中的剪切应力分量的定义是什么? 分别作用在哪个面? 哪个方向? 其会引起什么样的物理过程。下面来分析一个具体的无穷小微团受到的力。

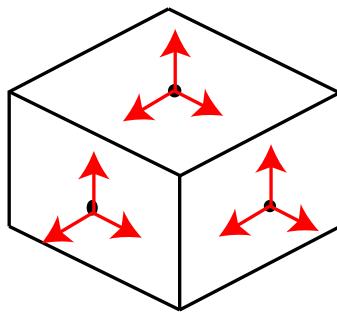
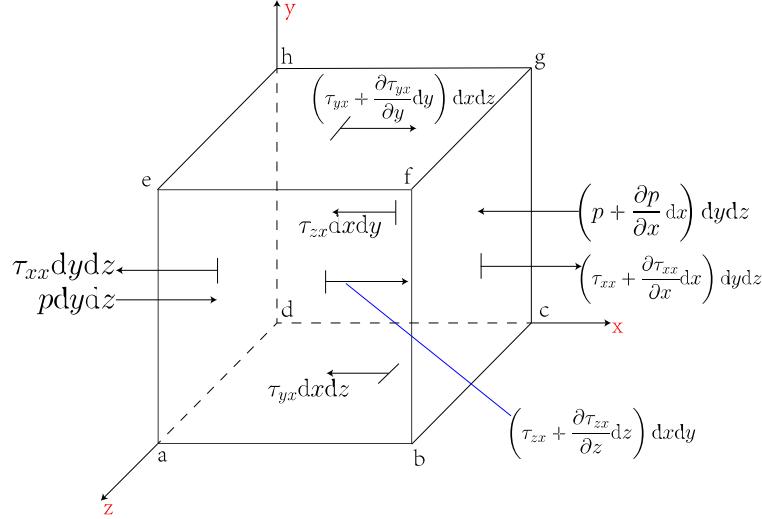


图 3.15: 三维剪切应力。

图3.16表示施加在无穷小微团 x 方向上的全部表面力¹⁵。在面 abcd 上, 仅仅存在由切应力引起的 x 方向的分量 $\tau_{yx}dxdz$ 。面 efg 和面 abcd 的距离为 dy , 所以 efg 面上的 x 方向的切应力为 $(\tau_{yx} + (\partial\tau_{yx}/\partial y)dy)dxdz$ 。对于面 abcd 和面 efg 上的切应力, 要注意他们的方向。在底面, τ_{yx} 是向左的 (与 x 轴方向相反), 在顶面, $\tau_{yx} + (\partial\tau_{yx}/\partial y)dy$ 是向右的 (与 x 轴方向相同)。这与下面的约定是一致的: 即速度的三个分量的正的增量与坐标轴的正向一致。例如, 图3.16中的平面 efg, 因为 u 沿着 y 轴正向是增加的, 所以在稍微高于平面 efg 的地方, 速度 u 要比平面 efg 上的 u 大。于是就形成了“拉”的动作, 试图将流体微团向右拉向 x 轴的正向。相反, 若考虑平面 abcd, 则在稍稍低于平面 abcd 的地方, 速度 u 要比平面 abcd 上的 u 小。于是对流体形成了一个阻碍的作用, 作用在 x 轴的负向。图3.16中其他剪切应力的方向, 都可以用相同的方式进行判断。特别是在面 dcgh 上, τ_{zx} 指向 x 轴负向, 而在面 abfe 上, $\tau_{zx} + (\partial\tau_{zx}/\partial z)dz$ 指向 x 轴正向。在垂直于 x 轴的面 adhe 上, x 方向的力有压力 $pdydz$, 指向流体微团的内部; 还有沿 x 轴负向的应力 $\tau_{xx}dydz$ 。依据前面提到的速度增量方向的约定, 我们可以解释为什么在图3.16中, 面 adhe 上 τ_{xx} 是指向左边的。依据规定, 速度 u 的正增量和 x 轴的正向一致, 所以稍微离开面 adhe 左面一点点, u 的值比面 adhe 上的 u 要小。因此, 正应力的粘性作用在面 adhe 上就好像是一个吸力, 产生一个向左拉的作用, 想要阻止流体微团的流动。与此相反, 在面 bcfg 上, 压力 $(p + (\partial p/\partial x)dx)dydz$ 指向流体微团的内部 (沿着 x 轴负向)。而由于在稍微离开面 bcfg 右面一点点的地方, u 的值比面 bcfg 上的 u 要大, 就会产生一个由正应力引起的吸力, 将流体微团向右拉, 这个力的大小为 $(\tau_{xx} + (\partial\tau_{xx}/\partial x))dydz$, 方向指向 x 轴正向。如果上面的描述同学们感觉难以理解, 下面的结论要知悉。通过对流体微团 x 方向的受力做分析, 有 x 方向所受到的表面力 (压力和剪切应力贡献) 为:

¹⁵ 图3.16看起来是非常复杂的。如果感觉理解有困难, 可以进一步参考文献 [13] 第 47 页的描述。

图 3.16: x 方向各个面受到的力。

$$F_x = \left(p - \left(p + \frac{\partial p}{\partial x} dx \right) \right) dy dz + \left(\left(\tau_{xx} + \frac{\partial \tau_{xx}}{\partial x} dx \right) - \tau_{xx} \right) dy dz + \left(\left(\tau_{yx} + \frac{\partial \tau_{yx}}{\partial y} dy \right) - \tau_{yx} \right) dx dz + \left(\left(\tau_{zx} + \frac{\partial \tau_{zx}}{\partial z} dz \right) - \tau_{zx} \right) dx dy \quad (3.61)$$

即

$$F_x = \left(-\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} \right) dx dy dz \quad (3.62)$$

同理，有 y 方向和 z 方向：

$$F_y = \left(-\frac{\partial p}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} \right) dx dy dz \quad (3.63)$$

$$F_z = \left(-\frac{\partial p}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} \right) dx dy dz \quad (3.64)$$

方程(3.62)、(3.63)和(3.64)即为流体微团在三个方向上的受力（为一个矢量）：

$$\mathbf{F} = \begin{bmatrix} \left(-\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} \right) dx dy dz \\ \left(-\frac{\partial p}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} \right) dx dy dz \\ \left(-\frac{\partial p}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} \right) dx dy dz \end{bmatrix} \quad (3.65)$$

3.5.2 动量守恒

在分析完受力之后，回到牛顿第二定律，即力等于质量乘加速度。力已经在方程(3.65)中给出，现在来看质量和加速度。在上文中，已经知道流体微团的质量可以表示为 $dm =$

$\rho dxdydz$, 同时, 流体微团的加速度为 $\frac{D\mathbf{U}}{Dt}$ (参考物质导数一节), 其分量形式为

$$\left[\frac{Du}{Dt}, \frac{Dv}{Dt}, \frac{Dw}{Dt} \right]^T \quad (3.66)$$

那么有 x 方向的动量方程为

$$\rho dxdydz \frac{Du}{Dt} = \left(-\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} \right) dxdydz \quad (3.67)$$

同理, y 方向和 z 方向可以类似导出。即

$$\rho \frac{Du}{Dt} = -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} \quad (3.68)$$

$$\rho \frac{Dv}{Dt} = -\frac{\partial p}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} \quad (3.69)$$

$$\rho \frac{Dw}{Dt} = -\frac{\partial p}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} \quad (3.70)$$

方程(3.68)、(3.69)和(3.70)即动量方程。同学们应该结合上文介绍的相关知识, 将方程(3.68)、(3.69)和(3.70)左边项写成欧拉形式 (参考物质导数一节)。然后, 再将这三个分量形式的动量方程写成矢量形式 (结合张量一节) ¹⁶:

$$\rho \frac{D\mathbf{U}}{Dt} = -\nabla p + \nabla \cdot \boldsymbol{\tau} \quad (3.71)$$

也即为

$$\rho \left(\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} \right) = -\nabla p + \nabla \cdot \boldsymbol{\tau} \quad (3.72)$$

3.5.3 守恒/非守恒转化

方程(3.72)为动量方程的非守恒形式。这是因为上述动量方程是从空间位置移动的无穷小微团进行推导的。非守恒控制方程可通过结合连续性方程转化为守恒的控制方程。将连续性方程(3.54)左右乘以 \mathbf{U} 有

$$\mathbf{U} \frac{\partial \rho}{\partial t} + \mathbf{U} \nabla \cdot (\rho \mathbf{U}) = 0 \quad (3.73)$$

随后将方程(3.73)与(3.72)相加有

$$\mathbf{U} \frac{\partial \rho}{\partial t} + \mathbf{U} \nabla \cdot (\rho \mathbf{U}) + \rho \frac{\partial \mathbf{U}}{\partial t} + \rho \mathbf{U} \cdot \nabla \mathbf{U} = -\nabla p + \nabla \cdot \boldsymbol{\tau} \quad (3.74)$$

即守恒形式的动量方程:

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} \quad (3.75)$$

¹⁶ 另一种常见的写法为

$$\rho \frac{D\mathbf{U}}{Dt} = \nabla \cdot \boldsymbol{\sigma}$$

其中 $\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}$ 。在这种情况下 $\boldsymbol{\sigma}$ 考虑的为总应力 (剪切应力 $\boldsymbol{\tau}$ 和正应力 p 的和)。

连续性方程可以进行类似的转化。考虑从空间位置移动的无穷小微团进行推导的连续性方程(3.53)，依据物质导数的定义将其展开有

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{U} = \frac{\partial \rho}{\partial t} + \mathbf{U} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{U} = \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{U} = 0 \quad (3.76)$$

其中 $\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{U}$ 为非守恒形式， $\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{U}$ 为守恒形式。

同理，守恒形式的方程，在假定原始变量光滑的情况下，可以演变为非守恒形式。

3.5.4 封闭

方程(3.68)、(3.69)和(3.70)还不能求解，因为其中的未知量为方程左边的速度，但方程右边还存在未知量 τ ，这种待求未知量之外还存在未知量的情况，被称之为封闭问题。为了将其封闭，需要将 τ 表示为 \mathbf{U} 的函数。剪切应力 τ 和速度 \mathbf{U} 的关系，被称之为本构关系。他们的方程，被称之为本构方程。17世纪末牛顿指出，流体的剪切应力和速度的梯度成线性关系。这种流体因此也被称之为牛顿流体。还存在一些不符合牛顿流体定义的流体，被定义为非牛顿流体，会展现一些不常见的流动特性。

下面讨论一个新的 CFD 变量：形变率，其也被称为剪切速率。未知量 τ 可以表示为形变率的函数，形变率进一步可以表示为速度的函数。这样，方程(3.68)、(3.69)和(3.70)就可以封闭。形变率和剪切应力类似，是一个二阶张量，一般用 \mathbf{S} 来表示。但由于在 CFD 中，形变率本身并不会封闭方程（而是形变率和速度的关系封闭了方程），因此在 CFD 方程中往往不会出现形变率这一项¹⁷。若 \mathbf{S} 符号表示形变率，即：

$$\mathbf{S} = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \quad (3.77)$$

对于各向同性流体，形变率的 9 个分量有 6 个是独立的：

$$S_{xy} = S_{yx}, S_{xz} = S_{zx}, S_{yz} = S_{zy} \quad (3.78)$$

当然也存在一些各向异性流体，如一些聚合物。对这些流体的研究会导致问题更加复杂。当然，很多模型都是从各向同性假定出发的。形变率进一步的可以表示为速度的方程：

$$\mathbf{S} = \begin{bmatrix} \frac{1}{2} \left(\frac{\partial u_1}{\partial x} + \frac{\partial u_1}{\partial x} \right) & \frac{1}{2} \left(\frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x} \right) & \frac{1}{2} \left(\frac{\partial u_1}{\partial z} + \frac{\partial u_3}{\partial x} \right) \\ \frac{1}{2} \left(\frac{\partial u_2}{\partial x} + \frac{\partial u_1}{\partial y} \right) & \frac{1}{2} \left(\frac{\partial u_2}{\partial y} + \frac{\partial u_2}{\partial y} \right) & \frac{1}{2} \left(\frac{\partial u_2}{\partial z} + \frac{\partial u_3}{\partial y} \right) \\ \frac{1}{2} \left(\frac{\partial u_3}{\partial x} + \frac{\partial u_1}{\partial z} \right) & \frac{1}{2} \left(\frac{\partial u_3}{\partial y} + \frac{\partial u_2}{\partial z} \right) & \frac{1}{2} \left(\frac{\partial u_3}{\partial z} + \frac{\partial u_3}{\partial z} \right) \end{bmatrix} \quad (3.79)$$

若写成矢量形式，有¹⁸：

$$\mathbf{S} = \frac{1}{2} (\nabla \mathbf{U} + \nabla \mathbf{U}^T) \quad (3.80)$$

¹⁷ \mathbf{S} 这个符号在有限体积法中更倾向于被定义为面矢量。

¹⁸ 方程(3.79)若按项对应应为 $\mathbf{S} = \frac{1}{2} (\nabla \mathbf{U}^T + \nabla \mathbf{U})$ 。

对于牛顿流体, 其定义剪切应力和形变速率为线性的关系。进一步考虑不可压缩流体(忽略体膨胀率)¹⁹:

$$\tau = 2\mu \mathbf{S} = \mu (\nabla \mathbf{U} + \nabla \mathbf{U}^T) \quad (3.81)$$

其中 μ 是粘度, 其为一个物理属性。将方程(3.81)代入到方程(3.75)中, 其中的 τ 就被速度表示出来进而被封闭(其中的密度 ρ 被除掉):

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U} \mathbf{U}) = -\frac{1}{\rho} \nabla p + \nabla \cdot (\nu (\nabla \mathbf{U} + \nabla \mathbf{U}^T)) \quad (3.82)$$

其中 $\nu = \frac{\mu}{\rho}$ 。方程(3.82)中的 $\nabla \cdot (\nu (\nabla \mathbf{U} + \nabla \mathbf{U}^T))$ 在大量文献中被写为 $\nabla \cdot (\nu \nabla \mathbf{U})$, 下面我们来看一下为什么。首先有²⁰

$$\nabla \mathbf{U} = \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_2}{\partial x} & \frac{\partial u_3}{\partial x} \\ \frac{\partial u_1}{\partial y} & \frac{\partial u_2}{\partial y} & \frac{\partial u_3}{\partial y} \\ \frac{\partial u_1}{\partial z} & \frac{\partial u_2}{\partial z} & \frac{\partial u_3}{\partial z} \end{bmatrix} \quad (3.83)$$

对其转置操作有

$$\nabla \mathbf{U}^T = \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_1}{\partial y} & \frac{\partial u_1}{\partial z} \\ \frac{\partial u_2}{\partial x} & \frac{\partial u_2}{\partial y} & \frac{\partial u_2}{\partial z} \\ \frac{\partial u_3}{\partial x} & \frac{\partial u_3}{\partial y} & \frac{\partial u_3}{\partial z} \end{bmatrix} \quad (3.84)$$

然后求散度有²¹

$$\nabla \cdot (\nabla \mathbf{U}^T) = \nabla \cdot \begin{bmatrix} \frac{\partial u_1}{\partial x} & \frac{\partial u_1}{\partial y} & \frac{\partial u_1}{\partial z} \\ \frac{\partial u_2}{\partial x} & \frac{\partial u_2}{\partial y} & \frac{\partial u_2}{\partial z} \\ \frac{\partial u_3}{\partial x} & \frac{\partial u_3}{\partial y} & \frac{\partial u_3}{\partial z} \end{bmatrix} = \left[\frac{\partial}{\partial x} \left(\frac{\partial u_1}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial u_2}{\partial x} \right) + \frac{\partial}{\partial z} \left(\frac{\partial u_3}{\partial x} \right), \frac{\partial}{\partial x} \left(\frac{\partial u_1}{\partial y} \right) + \frac{\partial}{\partial y} \left(\frac{\partial u_2}{\partial y} \right) + \frac{\partial}{\partial z} \left(\frac{\partial u_3}{\partial y} \right), \frac{\partial}{\partial x} \left(\frac{\partial u_1}{\partial z} \right) + \frac{\partial}{\partial y} \left(\frac{\partial u_2}{\partial z} \right) + \frac{\partial}{\partial z} \left(\frac{\partial u_3}{\partial z} \right) \right] = 0 \quad (3.85)$$

这是因为对于任何一个标量 a , 有

$$\frac{\partial}{\partial y} \left(\frac{\partial a}{\partial x} \right) = \frac{\partial}{\partial x} \left(\frac{\partial a}{\partial y} \right) \quad (3.86)$$

因此, 动量方程(3.82)可写为

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U} \mathbf{U}) = -\frac{1}{\rho} \nabla p + \nabla \cdot (\nu \nabla \mathbf{U}) \quad (3.87)$$

¹⁹ 可压缩流体完整形式为 $\tau = \mu (\nabla \mathbf{U} + \nabla \mathbf{U}^T) - \lambda (\nabla \cdot \mathbf{U}) \mathbf{I}$, 其中 λ 为体积粘度, 也被称之为第二粘度系数等。Stokes 认为 $\lambda = \frac{2}{3}\mu$, 进一步, $\nabla \cdot \mathbf{U} = 0$, 因此可以化简。

²⁰ 值得注意的是 $\nabla \mathbf{U}$ 的定义并不唯一, 在流体力学中通常采取本文的定义。

²¹ 以 x 分量为例, 有连续性方程

$$\frac{\partial u_1}{\partial x} + \frac{\partial u_2}{\partial y} + \frac{\partial u_3}{\partial z} = 0$$

则

$$\frac{\partial}{\partial x} \left(\frac{\partial u_1}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial u_2}{\partial x} \right) + \frac{\partial}{\partial z} \left(\frac{\partial u_3}{\partial x} \right) = \frac{\partial}{\partial x} \left(\frac{\partial u_1}{\partial x} \right) + \frac{\partial}{\partial x} \left(\frac{\partial u_2}{\partial y} \right) + \frac{\partial}{\partial x} \left(\frac{\partial u_3}{\partial z} \right) = \frac{\partial}{\partial x} \left(\frac{\partial u_1}{\partial x} + \frac{\partial u_2}{\partial y} + \frac{\partial u_3}{\partial z} \right) = 0$$

这个封闭的方程若展开，为 3 个方程，且存在未知项速度 \mathbf{U} 和压力 p 共 4 个未知量，即可求解。

对于可压缩流体，结合方程(3.81)的脚注，有完整的动量方程为

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \nabla \cdot \left(\mu (\nabla \mathbf{U} + \nabla \mathbf{U}^T) - \frac{2}{3} \mu (\nabla \cdot \mathbf{U}) \mathbf{I} \right) \quad (3.88)$$

注意其中的 $\nabla \mathbf{U}^T$ 并不能化简，因为可压缩流体的 $\nabla \cdot \mathbf{U} \neq 0^{22}$ 。

读者应该仔细观察动量方程的 3 个分量方程，观察矢量形式下的散度、梯度算法如何和分量形式的方程对应。做到对动量方程不陌生、不恐惧。其只不过是各种偏导数放在了一起。具体如何通过有限体积法离散、数值格式以及算法稳定性等内容将在 GCFD 课程上介绍。

3.5.5 积分观点

动量的定义为质量和速度的乘积，若将动量除以体积，即 ρu 表示为动量密度。考虑一维的区域， (x_1, x_2) 的区间范围内的动量可以表示为：

$$\int_{x_1}^{x_2} \rho u dx \quad (3.89)$$

其变化率可表示为：

$$\frac{d}{dt} \int_{x_1}^{x_2} \rho u dx \quad (3.90)$$

(x_1, x_2) 内的动量变化率取决于流入和流出的通量。如何表示动量的通量？参考质量通量的定义，对于质量通量，其为密度和速度的乘积。那么动量的通量即为动量密度和速度的乘积，即 ρuu 。另外，压力也会引致动量的变化，因此， $\rho uu + p$ 表示动量的通量， (x_1, x_2) 内流入减去流出的通量为

$$-(\rho uu + p)|_{x_1}^{x_2} \quad (3.91)$$

结合方程(3.90)，有

$$\frac{d}{dt} \int_{x_1}^{x_2} \rho u dx = -(\rho uu + p)|_{x_1}^{x_2} \quad (3.92)$$

²² 另外一种更紧凑的动量方程写法为：

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U} + p \mathbf{I} - \rho \boldsymbol{\tau}) = 0 \quad (1)$$

如果考虑多相流体系，粒子相的方程与方程(1)具有类似的形式（方程右边需要添加动量交换项），这种方程形式更易于理解。例如，对于气固体系，有气相控制方程为：

$$\frac{\partial \alpha_g \rho_g \mathbf{U}_g}{\partial t} + \nabla \cdot (\alpha_g \rho_g \mathbf{U}_g \mathbf{U}_g + p_g \mathbf{I} - \alpha_g \rho_g \boldsymbol{\tau}_g) = \alpha_g \rho_g \mathbf{g} - \alpha_p \rho_p \mathbf{M} \quad (2)$$

其中 α_g 表示气相相分数， \mathbf{M} 表示动量交换项， \mathbf{g} 表示重力加速度矢量。同时有颗粒相控制方程（在这里考虑颗粒粒子分布为各向异性高斯分布，即十矩模型，更详细的信息请参考6.3.2节）为：

$$\frac{\partial \alpha_p \rho_p \mathbf{U}_p}{\partial t} + \nabla \cdot (\alpha_p \rho_p \mathbf{U}_p \mathbf{U}_p + \mathbf{P}_p + \mathbf{G}_p + \mathbf{Z}_p) = \alpha_p \rho_p \mathbf{g} + \alpha_p \rho_p \mathbf{M} \quad (3)$$

其中 \mathbf{P}_p 表示颗粒压力张量，其也可以分解为各向同性部分以及各向异性部分。各向同性部分为 $\Theta \mathbf{I}$ ，其中 Θ 表示颗粒温度。各向异性部分为 $\boldsymbol{\tau}_p$ ，其也可以通过 Boussinesq 近似进行假定。 \mathbf{G}_p 表示碰撞通量， \mathbf{Z}_p 表示摩擦通量。

上式即为积分观点下推导的一维动量方程。

同样的，考虑质量， (x_1, x_2) 的区间范围内的质量可以表示为：

$$\int_{x_1}^{x_2} \rho dx \quad (3.93)$$

其变化率为

$$\frac{d}{dt} \int_{x_1}^{x_2} \rho dx \quad (3.94)$$

质量的净通量可以表示为：

$$-(\rho u)|_{x_1}^{x_2} \quad (3.95)$$

因此，有连续性方程：

$$\frac{d}{dt} \int_{x_1}^{x_2} \rho dx = -(\rho u)|_{x_1}^{x_2} \quad (3.96)$$

现在尝试将积分形式转换为微分形式。进一步假定 $\rho uu + p$ 是光滑可导的，方程(3.92)可以写为：

$$\frac{d}{dt} \int_{x_1}^{x_2} \rho u dx = - \int_{x_1}^{x_2} \frac{d}{dx} (\rho uu + p) dx \quad (3.97)$$

进一步操作有

$$\int_{x_1}^{x_2} \left(\frac{d\rho u}{dt} + \frac{d}{dx} (\rho uu + p) \right) dx = 0 \quad (3.98)$$

即

$$\frac{d\rho u}{dt} + \frac{d}{dx} (\rho uu + p) = 0 \quad (3.99)$$

方程(3.99)即不考虑剪切变形 (τ 的非对角线元素为 0) 的一维的动量方程。本节呈现的是另一种动量方程的推导形式，更好的展现了从积分的角度推导动量方程的思想。

在这里需要注意的是，积分形式的方程和微分形式的方程有重要的区别。积分形式的方程(3.92)可以在控制体内出现间断。因为数学上并没有要求被积分的函数不能出现间断。但微分形式的方程(3.99)要求函数是连续的（上文中引入假定 $\rho uu + p$ 是光滑可导的），否则不可导。积分形式的 CFD 方程更加基础，更加普适性。同时，在网格变形的情况下，自动满足体积守恒方程。积分形式的 CFD 方程也适用于各种不同的网格形状。但之所以将积分形式的方程放在最后是因为积分形式的方程中会出现面积分，例如三维的连续性方程可以写为

$$\frac{d}{dt} \int_V \rho dV + \int_{S_f} \rho_f \mathbf{U}_f d\mathbf{S}_f = 0 \quad (3.100)$$

看起来对初学者并不友好。

3.6 学习要求与思考

上述内容完整详细地推导了普适性 CFD 最基本的控制方程：N-S 方程。在 GCFD 课堂中，将进一步讲解 N-S 方程的数值离散。至此，读者应该牢记连续性方程与动量方程的最终形式：

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (3.101)$$

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} \quad (3.102)$$

读者应该清楚方程(3.101)的第一项、第二项表示什么物理意义，(3.102)的第一项、第二项、第三项、第四项表示什么物理意义。同时，对各项的数学推导过程有大致的印象。N-S 方程是大量算法的根基，例如，直接模拟 DNS 就是调用高阶格式直接求解 N-S 方程，多相流 VOF 模型就是在 N-S 方程的基础上添加表面张力等体积力（体积力在哪一个方程中有体现？在方程的哪一侧？）。欧拉拉格朗日算法就是连续性方程使用 N-S 方程来描述，颗粒相使用拉格朗日算法来描述。湍流模型则是通过数值的方法增加方程的粘度使得易于收敛（粘度在方程中哪里有所体现？）。牛顿流体假定一个恒定的粘度项，非牛顿流体流动则是将粘度表示为可变的变量。相变过程则是在相分数方程和速度方程添加源项来实现等。

虽然基于 N-S 方程可以进行各种模型的演变。但在离散求解中，任何一个小小问题都可能会演变为最后的发散，例如，粘度过大可能会导致方程刚性、附加体积力的梯度项可能导致数值震荡、不依附于流线的对流项会导致较大的数值耗散，另外，如何处理温度小于 0 的情况？为什么有些情况湍流粘度过大？湍流模型的参数是否可以改动？速度压力迭代次数是否可以减少提高效率？为什么模拟过程中无法算出湍流？计算中所有的参数在数值上都有什么意义？所有这些内容，都属于 CFD 算法研究的内容。

在这里，读者可以重新翻到本笔记的前言部分，进一步了解 N-S 方程更加奇妙的数值特性。

第四章 多相流模型

多相流是自然界和工程技术中最常见的一种流动。相的概念是从热力学意义上定义的，可以理解为固态、液态和气态。自然界中存在各种各样的多相流动，如雪崩、火山云、雨夹雪等。同时，多相系统也大量的存在于大量的工业过程中，如鼓泡床、流化床、喷雾燃烧、碳烟生成、萃取塔、搅拌反应器等。在这些设备中，离散相粒子可能是自然的存在，也可能是处于某些过程的要求被人为注入。多相流是传热和传质的重要方法。

图4.1中形象的展示了一个气液多相系统和一个气固多相系统。在这些多相系统中，看起来可能并没有联系，但是通常可以用连续相（Continuous phase）和离散相（Disperse phase）对多相系统中的相进行区分。例如鼓泡床中的液体可以看作为连续相，注入的气泡（气体）可以认为是离散相。螺旋桨产生的空泡可以认为是离散相，周围的水可以认为是连续相¹。引擎中的喷雾可以认为是离散相，周围的空气可以认为是连续相。判定离散相和连续相的关系，主要是离散相往往存在某种属性的分布 [65]。例如鼓泡床中的气泡，存在粒径的分布。流化床中的粒子，存在传输速度的分布。连续相的大部分属性是均一的，比如鼓泡床中的液体以及流化床中的气体并不存在一个粒径分布²。

离散相的行为对工业过程的影响至关重要。在很多情况下，粒子要么作为一种产物产生，要么作为一种反应物为反应提供原料 [73]。工程师或许迫切想增加离散相的产率，或许想提高离散相和连续相的某种混合特性。对离散相的系统研究，有利于在更深的层次上理解整个过程的流动行为。多相流的研究方向主要有两方面：一是研究离散相的演变（如气泡的聚并和破碎），二是研究多相流体动力学。其中离散相的演变研究内容主要是研究离散相的具体单一粒子的变化过程。例如粒子如何破碎、如何碰撞、碰撞后的运动等。离散相粒子的这些行为主要由群体平衡模型（Population Balance Model, PBM）控制。多相流体力学的研究内容主要关心连续相和离散相的耦合作用，例如粒子运动产生的湍流与涡、粒子的边界层效果、气泡表面张力对气泡的形变等。这些研究需要结合流体力学和群体平衡模型来进行。简单来说，离散相的研究主要关心粒子，多相流体力学主要关心相间作用。近些

¹ CFD 中的相不同于物理化学或材料学中的相。比如材料学中的钢可以分为马氏体、奥氏体，在物理化学的铁碳相图中也存在包晶转变。但在 CFD 中，固相并不能够细化到晶型转变的程度。然而一些模拟方法，如离散单元法（Discrete Element Model, DEM），却可以识别固体的形状以及硬度等其他变量。

² 在某些情况下连续相和离散相会互相转变。比如表观气速持续增加的鼓泡床中，流型会从泡状流（Bubbly flow）转变为雾状流（Mist flow），这种情况下原本离散的气体变为连续相，原本连续的液体变为离散相。这种现象被称之为相转置。

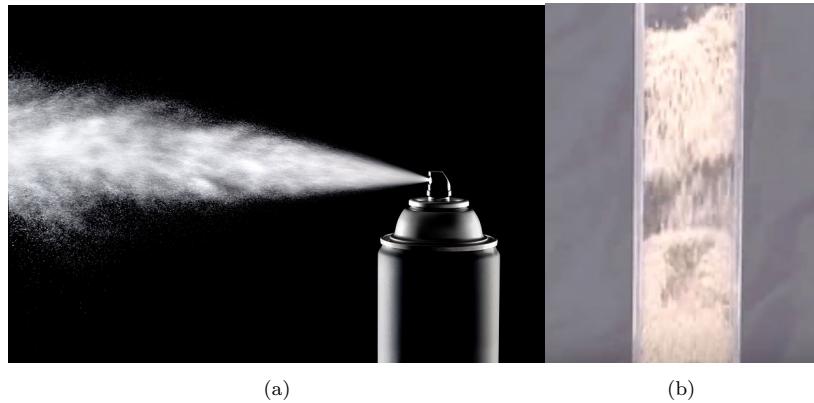


图 4.1: 多相系统示意图。a): 液滴喷雾。b): 小颗粒中通入空气。图片来源于网络。

年来，随着计算机计算能力的提高，计算流体力学（Computational Fluid Dynamic, CFD）依靠其独特的优势，被广泛的应用于多相流研究领域。

在这里，有必要对多相系统进一步进行区分。多相系统通常可以分为单分散系统以及多分散系统 [65]。在单分散系统（mono-disperse）中，离散相的属性为均一的。例如粒子具有相同的大小、温度、以及传输速度。在多分散系统（poly-disperse）中，离散相可以具有不同的属性，例如粒子可以具有不同的大小、温度或者速度。如图4.2所示，左图中的气泡可以近似分为是粒径单一的，因此可近似认为是单分散系统。右图中气泡的粒径差别非常大，可认为是多分散系统。另外一种重要的多分散系统为颗粒的其他属性均相同，但传输速度不同（如图4.3所示），例如大的颗粒移动速度较慢，小的颗粒移动速度较快。这种速度的非均匀分布，对多分散系统数学模型的描述产生了一定的挑战。在实际的工程应用中，单分散系统非常少见。但描述多分散系统的模型通常在数值上难以求解，因此多相流的研究一直是国际上的研究热点问题。

多相流体力学和多相计算流体力学同样也存在区别。多相流体力学侧重研究流动的现象，如微气泡的气泡减阻机理 [64]、颗粒团聚致湍流的物理描述 [23]、液滴气泡破碎的原因 [91, 42]。多相计算流体力学侧重研究如何求解多相流的数值模型，如双流体模型的双曲特征问题 [71]、界面重构的精准性问题 [75]、群体平衡模型的数值求解问题 [47]、颗粒动力学模化问题 [29]。从研究手段的角度来区分，多相流体力学主要通过实验研究自然界本质物理过程，多相计算流体力学主要通过数学手段和计算机研究如何求解多相流模型并编程植入以及验证。通常，多相流体力学的研究者会提出相应的多相流模型，然后多相计算流体力学研究者会将这些模型数值求解，并进行 CFD 模拟并验证。

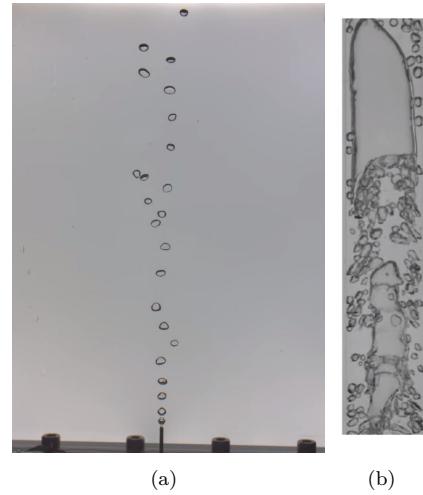


图 4.2: 近似单分散系统和多分散系统示意图。a): 近似单分散系统, b): 多分散系统。

4.1 多尺度模型

流体由分子组成。分子做随机热运动，且分子间距比分子尺度大得多。流体力学研究流体的运动，从对流体的解析层级角度来考虑。多相流体力学模型可以分为微观模型（Microscopic model）、介尺度模型（Mesoscopic model）以及宏观模型（Macroscopic model）。如图4.4所示，Sundaresan 等 [85] 将微观模型、介尺度模型以及宏观模型的应用尺度定位为毫米级、厘米级和米级。依据研究关注点的不同，应当使用不同的模型。例如，如果读者想研究小颗粒外部的边界层行为（如图4.5所示），那么必然要使用微观模型。如果读者只是关心宏观的相分数，那么使用宏观模型就可以。

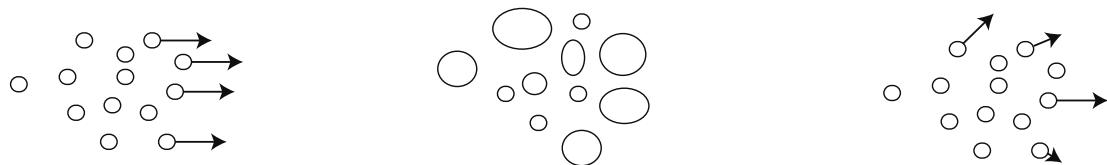


图 4.3: 近似单分散系统和多分散系统示意图。左: 所有粒子具有相同的特性 (单分散), 中: 粒子的直径不同 (多分散), 右: 粒子的移动速度不同 (多分散)。

4.1.1 微观模型

两相流中的微观模型近似为多相流的直接模拟（Direct Numerical Simulation, DNS）。多相流领域的 DNS 和单相流领域的 DNS 略有区分，单相流的 DNS 是指不调用任何模化

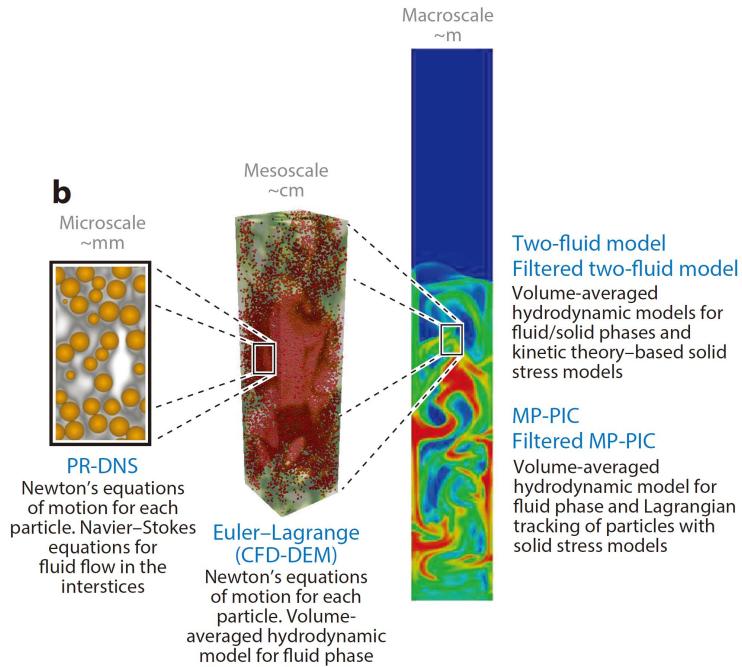


图 4.4: 多相系统的多尺度描述 [85]。

方法直接采用高阶空间格式以及高阶时间格式的直接模拟³，求解方法通常为谱方法以及有限差分法。多相流的 DNS 对求解精度并没有这么高的要求，主要通过求解的空间分辨率进行定义。在多相流的 DNS 中，直接求解 Navier-Stokes (N-S) 方程。不需要任何对界面的模化，不需要任何的附加模型。例如，在气固流动中，如果颗粒形状和位置固定，微观模型需要通过网格勾勒出相界面的拓扑结构，进一步的通过求解 N-S 方程获得流场结果，这种方法也被称之为颗粒解析的 DNS (Particle-Resolved DNS, PR-DNS) [87]。作为微观模型，PR-DNS 不需要对颗粒和流体间的作用力进行模化，但只能模拟小规模的系统（大约 10^5 个颗粒 [85]）。主要用于验证介尺度模型和宏观模型中的子模型。

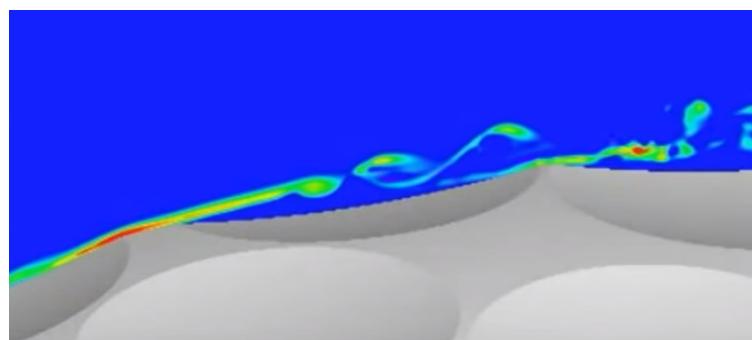


图 4.5: 颗粒表面的边界层行为。

³也有很多文献使用二阶有限体积法做 DNS，在这种情况下没有进行模化，但是精度较低，因此通常被称之为 quasi-DNS。

对于其他的流型如气液两相流，由于气液界面是变化的，因此微观模型需要能够对界面自发地进行捕获（如图4.6所示）。后者，这种相界面发生变形的直接模拟，往往比颗粒位置固定的气固流动模拟更为复杂。精准的预测相界面的变形需要对方程中的表面张力、粘度以及相间的密度进行合理的处理。现存的处理方法主要为体方法（Volume method）和面方法（Surface method），前者也被称之为界面捕获法（Surface capturing method），后者也被称之为界面拟合法（Surface fitting method）[92]。在界面捕获法中，界面通过变量重构而来。比较著名的界面捕获法为体分数法（Volume of Fluid, VOF）[44]。在界面拟合法中，界面通过标记点被直接拟合标记。比较著名的界面拟合法有水平集方法（Level-set method）[86] 以及界面跟踪法（Front-tracking method）[93]。

微观模型对网格分辨率和时间解析的要求非常高。例如在对颗粒表面边界层进行微观模型模拟的时候，可能会要求对颗粒表面的涡旋进行解析。解析这些小尺度涡旋需要非常高的计算资源。对于气液体系，微观模型目前只能用于模拟几十几百个数量级的气泡流动。目前，微观模型主要用于小微尺度的流动。虽然微观模型消耗的计算机资源过高，但是下文中即将提到的介尺度模型很大一部分依赖于微观模型的研究工作。例如，对气固体系的直接模拟可以预测一些实验很难监控的物理量，如实时的升力、曳力、虚拟质量力等[16]。因此，多相流的微观模拟，将在未来依然是一个学术上的研究热点问题。

4.1.2 宏观模型

宏观模型研究多相流体的宏观运动。研究的对象不是粒子本身，而是从这些粒子演变出来的物质模型，也即将整个粒子体系看作是连续介质。宏观模型认为粒子无间隙的分布于整个空间中。粒子的变量为空间以及时间的连续函数。宏观模型的变量为考虑大量粒子平均后的统计变量。既然要做平均，那么就涉及到平均的尺度问题。尺度选择过大，宏观变量会受到周围的影响而改变。尺度选择过小，变量会因为分子热运动导致变化。在多相计算流体力学中，宏观模型的尺度通常和粒子的尺度有关，也和粒子的运动尺度有关。粒子的尺度越大，对应的宏观模型的尺度就要增加。粒子的运动尺度很大，宏观模型的尺度也要增加。在错误的尺度上使用宏观模型进行分析，将产生不合理的结果。

讨论宏观模型的非适用性，需要定义克努森数（Knudsen number, Kn）。如图4.7所示，在分子动力学领域，克努森数可以理解为粒子平均自由程和流体特征尺度的比值。在多相流领域，克努森数主要用于描述粒子间的碰撞作用相对于自由传输的重要性。在克努森数较小的时候，粒子间的碰撞起主要作用，因此由于较为频繁的碰撞作用导致粒子的速度分布较为均衡，满足连续性假定。在克努森数大于 0.5 的时候，颗粒间的碰撞可以忽略，连续介质假定不再适用，在这种情况下使用 CFD 的宏观模型（如 N-S 方程）可能会引起错误的结果。在图4.8可以看出，在克努森数较小的情况下，N-S 方程是适用的。在克努森数较大的情况下，N-S 方程将不再适用。类似的，如果将空气分子当做离散的粒子，在稀薄

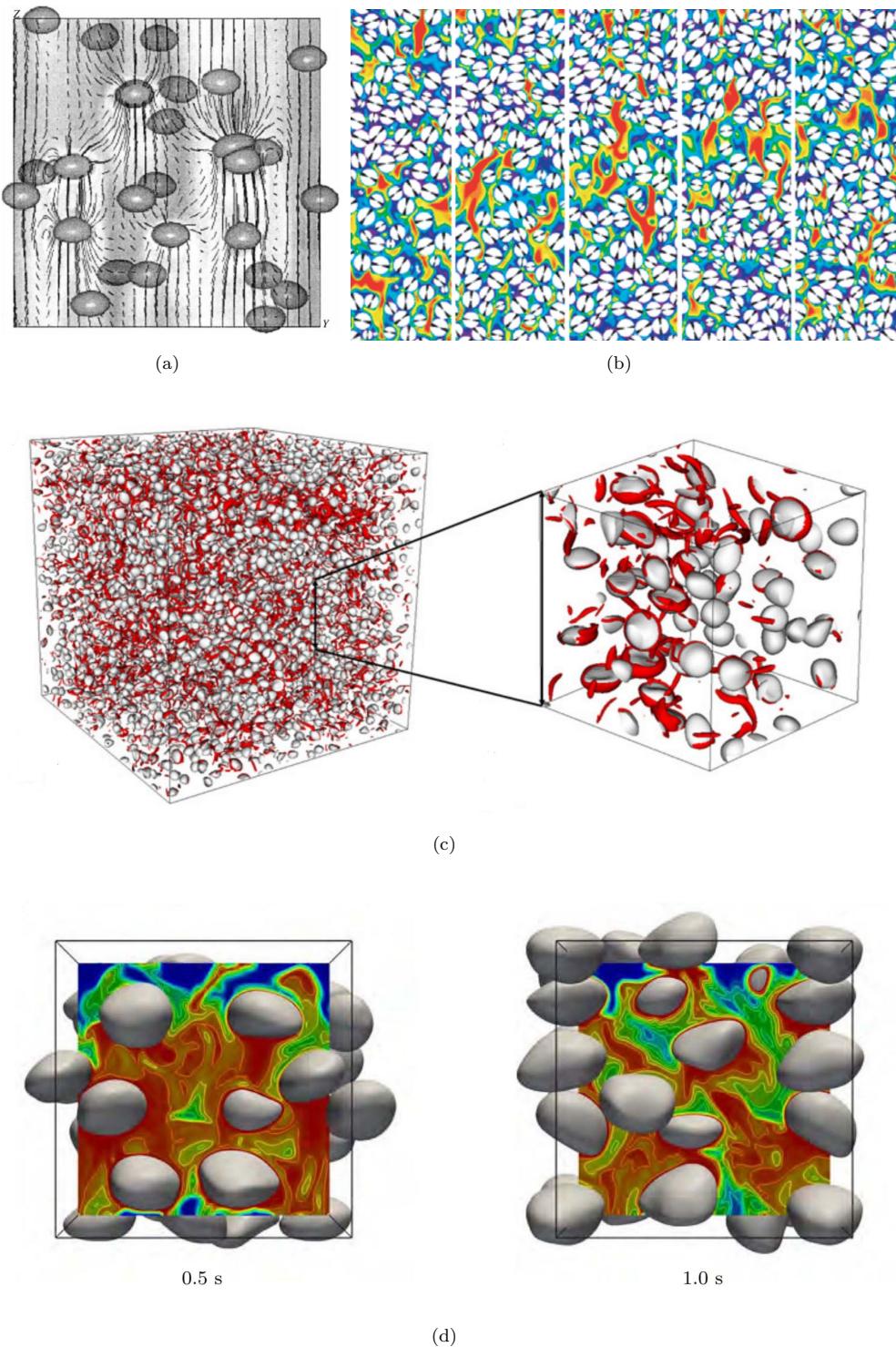


图 4.6: a): 使用 front tracking 直接模拟气泡形状 [90]。b): 直接模拟颗粒间隙的流体流动 [28]。c): 在雷诺数为 83 的情况下使用 VOF 模拟的共 3130 个液滴 [30]。d): 使用 front tracking 直接模拟气泡传质 [76]。

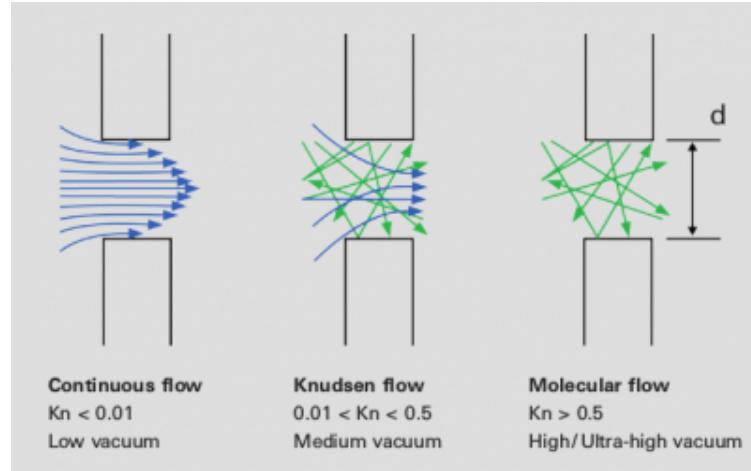


图 4.7: 克努森数的定义与示意图。在克努森数小于 0.01 的情况下，粒子频繁的碰撞近乎均衡，满足连续性介质假定。在克努森数大于 0.5 的情况下，粒子近乎无碰撞，不满足连续性介质假定。在克努森数介于之间的的时候，处于一种过渡状态。

气体领域，由于空气分子近似无碰撞，因此 N-S 方程也将失效，需要调用其他适用的模型。

在宏观模型中，连续相和离散相均在欧拉框架下被阐述。若考虑一个离散相和一个连续相，相应的模型被称之为双流体模型，也即欧拉-欧拉模型 [31]。若考虑多个离散相，相应的模型被称之为多流体模型。不同于下一节讨论的介尺度模型，宏观模型的隐性假定是每一相都可以通过水动力模型（Hydrodynamics model）来模化⁴。

宏观模型的未封闭项至少有两项：1) 界面动能/质量传递项：在微观模型中，界面的传递不需要模化，而是通过相边界条件来进行计算。在宏观模型中，相边界不能被解析，因此宏观模型的界面传递通常从微观模型进行修正而来。2) 应力张量项：离散相的速度波动并不一定产生于湍流，也有可能产生于源项。这类似于分子模拟的动力学理论-压力张量起源于分子的速度波动。在气-固流动中，宏观模型中的速度波动会导致产生一个附加的变量：颗粒应力张量（Granular stress tensor）。这些附加的压力张量以及颗粒应力张量并不是起源于湍流⁵。这两个未封闭项的处理难易不同。第一种未封闭项可以通过考虑更多的微观因素来提高模型的封闭精度。第二种的封闭并不容易。通常的做法是在模型中引入一个湍流源项，如气泡引致湍流（Bubble Induced Turbulence, BIT）。

4.1.3 介尺度模型

微观模型通常包含大量的流动信息，这些信息对于大尺度的工业设备并不是必要的。在微观模型和宏观模型之间，还存在介尺度模型。目前科学的研究领域，已经深入小到分子原子，大到行星恒星。但对于介尺度领域的研究还较为空白 [61]。介尺度模型可通过引

⁴ 水动力模型：克努森数非常小的情况下通过 Chapman-Enskog 模型进行展开的低阶矩模型。

⁵ 一些文章中指出将这些张量称之为湍流应力张量是不正确的 [35]。

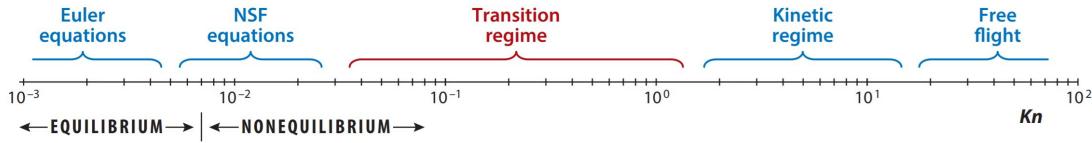


图 4.8: 依据克努森数定义的方程适用性 [89]。

入介尺度变量来减少这些流动变量的自由度。例如，考虑一个颗粒相，这些颗粒相具有不同的体积和不同的速度波动（即便是相同体积的颗粒也可能具有不同的速度波动）。介尺度模型不需要考虑每个颗粒的速度波动差异，其认为每个相同体积的颗粒具有同样的速度波动大小（一种条件速度分布）。介尺度模型是微观模型与宏观模型之间的一种模型。所有的宏观模型都可以从介尺度模型或微观模型获得。进一步可以区分为两条路径：

- 直接从微观模型推导宏观模型；
- 从微观模型推导介尺度模型，然后从介尺度模型推导出宏观模型；

若直接从微观模型推导宏观模型，则仅仅需要调用体平均（或集合平均技术）来获得矩方程，并通过本构方程来进行直接封闭 [31]。这种方法只涉及到物理方面的建模。在这里需要注意的是，宏观模型中通常将矩具体化为平均动量、平均质量等。若按照第二条路线，在从微观模型推导至介尺度模型的时候同样需要物理方面的建模，在从介尺度模型推导至宏观模型的时候则需要数学方面建模来进行封闭（如封闭介尺度变量的矩）。例如如果从粒子的普适性群体平衡模型出发，0 阶矩表示粒子的数量浓度，1 阶矩表示粒子的平均动能。因此，在第二条路线上，同时考虑了粒子的物理行为以及数学特征。目前在多相流领域，较为受欢迎的介尺度模型为欧拉-拉格朗日模型。同时，作为另一种介尺度模型，通过 CFD 和群体平衡模型的耦合进行求解也越来越引起研究学者的关注 [65, 59, 58, 55, 57, 37, 56, 54, 60]。

不同于宏观模型，欧拉-拉格朗日模型直接跟踪粒子的运动轨迹，并直接求解粒子间的碰撞。若继续考虑颗粒的形状、摩擦、粘连等物理属性，这种直接跟踪粒子的计算方法被称之为离散元法（Discrete Element Method, DEM）。将 CFD 和离散元法进行耦合，即 CFD-DEM。CFD-DEM 被大量的用于研究介尺度下的流动，尤其是气-固流动行为。但 CFD-DEM 对计算资源要求极高，通常仅仅被用于小尺度的模拟。粒子数量一般也小于百万个。

4.1.4 多尺度模型的适用性

各种尺度的模型由于模化的程度不同，具有不同的适用性。在这里通过一些多相体系的例子，讨论应该选用什么样的模型。例如在此处的讨论中，用户需要模拟气泡的聚并和破碎。这是一种介尺度层级的现象，因此需要调用微观模型或介尺度模型。宏观模型并不

能够预测类似的行为。在此处的讨论中，用户需要同时模拟气泡以及颗粒的行为，并考虑气泡和颗粒的界面交互作用。用户原本使用微观模型跟踪气泡的界面，同时调用拉格朗日模型处理颗粒行为。但需要注意的是，微观模型需要保证足够网格分辨率才能捕获气泡的界面（如一个三维的气泡内存在 90 个网格），同时介尺度模型需要保证网格充分的大（如一个网格内可包含 10 个颗粒）。在同一个计算域上调用不同尺度的模型，可能会出现问题。

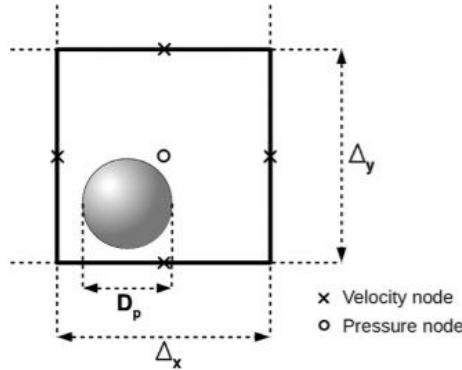


图 4.9: 离散相颗粒直径 D_p 和欧拉网格分辨率 Δx 之间的关系 [68]。

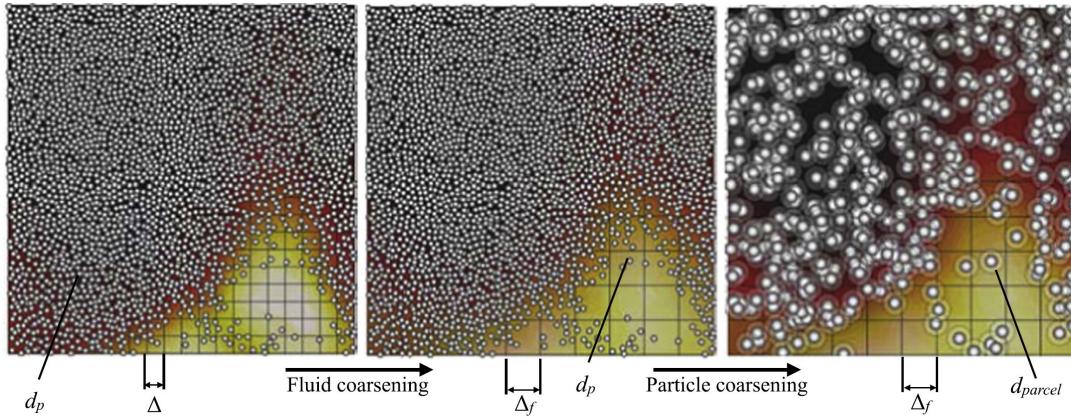


图 4.10: 左: 细网格欧拉-拉格朗日模拟。中: 糙网格欧拉-拉格朗日模拟示意图。右: MP-PIC 方法 [70]。

总体来讲，在同一个计算域上，需要优先保证模型的尺度小于需要预测的现象尺度。但需要注意的是，在使用宏观模型的时候，很多工作的网格分辨率均和离散相尺寸同量级。同时，在调用网格依赖类湍流模型时，模型的混用也可能需要适配。例如，如果连续相使用大涡模拟类模型，同时离散相选用拉格朗日模型。一方面要保证网格的分辨率足够高来对大尺度涡进行捕获，一方面还要保证网格单元足够大满足欧拉拉格朗日耦合的数学假定。参考图 4.9, Milelli et al. 认为颗粒直径 D_p 和欧拉网格分辨率 Δx 的关系应该满足 $D_p/\Delta x < 0.67$ [68]。Liu et al. 研究了欧拉-欧拉/大涡模拟中网格大小以及气泡直径对结果的影响 [63]。参

考图4.10中的研究方法可以看出对于使用混合方法模拟多相体系，其中的网格分辨率标准还需要进行大量研究。感兴趣的读者可参考其他文献资料 [72, 68]。

第五章 数值方法

5.1 稀疏线性系统求解器

CFD 方程在离散之后会形成下面的求解系统：

$$\mathcal{A}\mathbf{x} = \mathbf{b} \quad (5.1)$$

其中 \mathcal{A} 表示一个矩阵系统， \mathbf{x} 为待求的未知量， \mathbf{b} 表示源项。对于真实的 CFD 问题， \mathbf{x} 的未知量与网格数量相当。因此，如果使用大家非常熟悉的直接解法求解方程(5.1)是基本上不可行的。这是因为直接解法需要进行的操作次数是非常庞大的。例如，对于三维网格，每个方向存在 100 个网格点，那么会形成 100 万阶矩阵。对这个 100 万阶矩阵进行直接求解，需要使用 $100!$ 次操作。

在这种情况下，通常需要采用迭代的方法进行计算。迭代的方法看起来天生就是为了求解偏微分方程组而生。这是因为大量的偏微分方程组在离散的过程中，在调用格式的时候，通常仅仅调用相邻网格点的变量，即使调用高阶格式（三阶四阶），在网格数量很大的时候，离散形成的线性系统也是一个非常稀疏的求解系统。这里的稀疏是指变量 \mathcal{A} 大部分的元素都是 0。如果使用中心格式，对于一维问题， \mathcal{A} 的每一行只有 3 个元素，形成一个对角阵。对于二维问题， \mathcal{A} 的每一行只有 5 个元素。会形成一个带状（Banded）系统¹。很明显， \mathcal{A} 为一个稀疏矩阵。另外一个规律是，对于很多网格编号中规中矩的范例（如没有对网格编号进行打乱排布的结构网格）， \mathcal{A} 的形态也是非常规则的。典型的迭代方法有高斯赛德尔迭代法与雅克比迭代法。

5.1.1 基本迭代求解器

本书中将雅克比迭代求解器、高斯迭代求解器以及相关衍生分支分类为基本迭代求解器。雅克比迭代法的思想非常简单。对于矩阵 \mathcal{A} ，其可以分解为上三角矩阵 U 、下三角矩

¹一维问题如果调用高阶格式也会形成带状系统。同时，多维问题带状系统的带宽往往要比一维问题的大。

阵 L 以及对角阵 D^2 :

$$\mathcal{A} = D - U - L \quad (5.2)$$

在这种情况下, 那么第 $k+1$ 次迭代的解 \mathbf{x}^{k+1} 可以这样获得:

$$\mathbf{x}^{k+1} = D^{-1}[\mathbf{b} + (L + U)\mathbf{x}^k] \quad (5.3)$$

类似的, 高斯赛德尔迭代法可以这样求解:

$$\mathbf{x}^{k+1} = (I - D^{-1}L)^{-1}[D^{-1}\mathbf{b} + D^{-1}U\mathbf{x}^k] \quad (5.4)$$

对于各种不同的迭代法, 可以定义普适性方程:

$$\mathbf{x}^{k+1} = G\mathbf{x}^k + c \quad (5.5)$$

不同的迭代法, 对应不同的 G 与 c 。同时, 真实解 \mathbf{x} 满足方程:

$$(I - G)\mathbf{x} = c \quad (5.6)$$

定义 k 次迭代的误差为 \mathbf{e}^k , 有:

$$\mathbf{e}^k = \mathbf{x} - \mathbf{x}^k \quad (5.7)$$

例如, 第 0 次 (初始条件) 以及第 1 次的误差为:

$$\mathbf{e}^0 = \mathbf{x} - \mathbf{x}^0 \quad (5.8)$$

$$\mathbf{e}^1 = \mathbf{x} - \mathbf{x}^1 \quad (5.9)$$

如果将方程(5.6)、(5.5)代入到方程(5.9)有:

$$\mathbf{e}^1 = \mathbf{x} - \mathbf{x}^1 = G\mathbf{x} - (\mathbf{x}^1 - c) = G\mathbf{x} - G\mathbf{x}^0 = G(\mathbf{x} - \mathbf{x}^0) = G\mathbf{e}^0 \quad (5.10)$$

同理, 有:

$$\mathbf{e}^{k+1} = G\mathbf{e}^k = G^2\mathbf{e}^{k-1} = \dots = G^{k+1}\mathbf{e}^0 \quad (5.11)$$

可以看出 G 的作用类似一种缩放因子, 为了使得 \mathbf{e}^{k+1} 趋向于无穷小, 其谱半径需要小于 1, 即:

$$\rho(G) < 1 \quad (5.12)$$

其中 $\rho(G)$ 表示 G 的谱半径。

²在这里将 L 和 D 写成负号的形式, 是因为对于 PDE 离散系统, 对角系数与非对角系数通常为反号。例如我们来看一个简单的一维方程: $-\frac{\partial}{\partial x} \left(\frac{\partial \phi}{\partial x} \right) = 0$ 。在这里需要注意的是这个方程中的扩散项系数为负值。虽然其正负不影响计算结果, 但这个括号会影响对角线系数的正负。只有为负值的时候, 对角线系数为正。且只有为正值的时候, 其表示一个扩散方程。

5.1.2 多重网格求解器

然而，基本迭代求解器的收敛效率严重依赖于变量数。如图5.1所示，如果变量数量很多，收敛速率会非常慢，严重影响 CFD 计算的效率。

多重网格法直击传统迭代法的痛点，力求在大网格算例下，获得与小网格算例同样的收敛效率。举例，对于方程(5.1)中给定的稀疏线性系统，初始值假定为 \mathbf{x}^0 ，后续的迭代解被假定为 \mathbf{x}^k ，在这种情况下，有 k 次的残差：

$$\mathbf{r}^k = \mathbf{b} - \mathcal{A}\mathbf{x}^k \quad (5.13)$$

定义 k 次迭代的误差为 \mathbf{e}^k ，那么有：

$$\mathbf{e}^k = \mathbf{x} - \mathbf{x}^k \quad (5.14)$$

在这里需要注意的是，通过方程(5.14)是无法求出 \mathbf{e}^k 的，因为真值 \mathbf{x} 并不知道。将方程(5.14)代入到(5.13)有：

$$\mathcal{A}\mathbf{e}^k = \mathbf{r}^k \quad (5.15)$$

仔细观察方程(5.1)与(5.15)，会发现二者惊人的相似。方程(5.15)又被称为残差方程。其用来描述误差与残差的关系。理解方程(5.15)需要明确：

- 针对某 k 次的残差 \mathbf{r}^k ，存在一个 \mathbf{e}^k ；
- 如果可以获得 $\mathbf{e}^k, \mathbf{r}^k$ ，那么真值 \mathbf{x} 可以通过方程(5.13)获得；
- 方程(5.15)描述残差与误差的关系，其与方程(5.1)具有相同的形式，其中的 \mathbf{e}^k 也需要进行迭代求解 ($\mathbf{e}^{k,0} \rightarrow \mathbf{e}^{k,1} \rightarrow \mathbf{e}^{k,2} \rightarrow \mathbf{e}^{k,n} \approx \mathbf{e}^k$)；

多重网格的思想就是³：

- 光顺 (Smooth)：首先在原始网格上求解几次方程(5.1)，保证前期的残差快速下降；
- 限定 (Restriction)：在获得残差 \mathbf{r}^k 之后，求解几次方程(5.15)，只不过在求解的过程中，在相对糙的网格上进行求解（网格粗化）。争取快速获得一个误差 $\mathbf{e}^{k,n}$ ；
- 延拓 (Prolongation)：将粗网格的误差插值到细网格上；
- 在细网格上光顺几次，然后继续回到初始状态进行迭代；

在实际求解中。如图5.2所示，多重网格会调用多级的限定与延拓。

³中文译名参考宇波的工作 [14]。

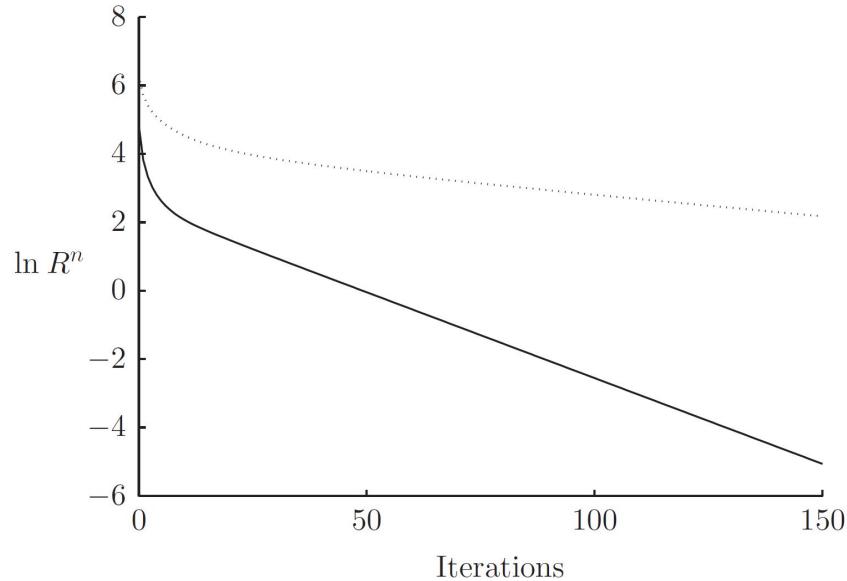


图 5.1: 高斯赛德尔迭代求解器残差收敛速率。实线: 网格数 1000, 虚线: 网格数 100000。

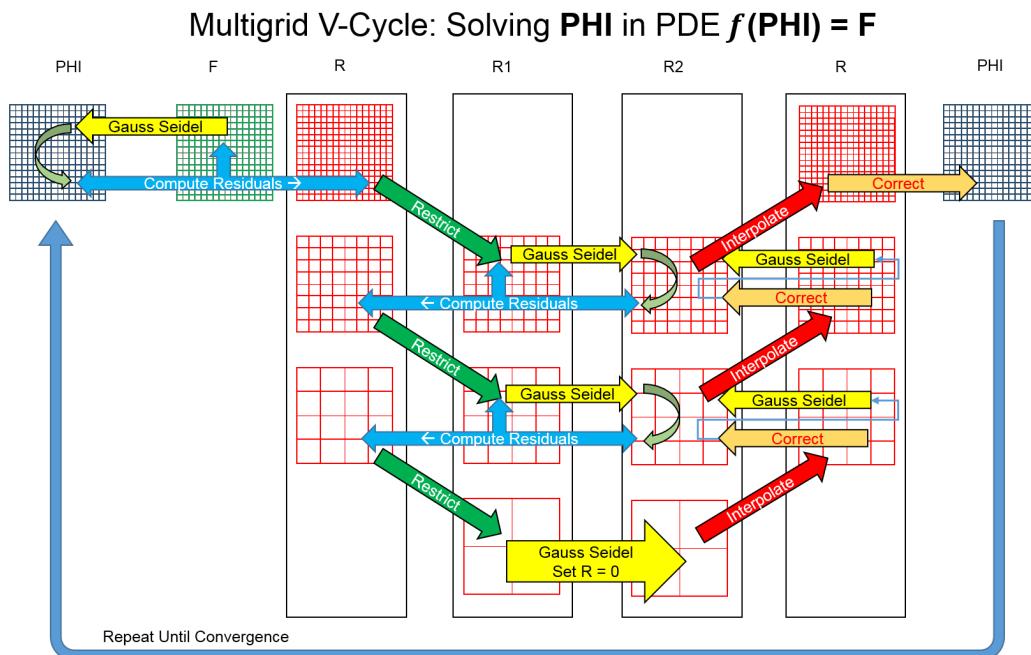


图 5.2: 多重网格实施思想 [6]:

第六章 拓展内容

本章内容不需要参加 GCFD 课程的学员掌握，仅做介绍，激发兴趣，理解即可。理解不了也没关系。

6.1 湍流

湍流的流动状态是非常瞬态的。如果将速度对时间做数据提取，会发现其呈现一个非常随机的波动状态。且流动中存在大量的涡。湍流过程将具有不同动量的流体单元“打散”且“混合”到一起。同时由于粘性的作用，速度的梯度会趋向于减小，并进一步的减小动能。这些动能被转化为内能。湍流有些情况是让人喜欢的，有些情况却让人厌恶。如果过程中需要加强传递与混合，那么湍流会让研究学者欣喜若狂。另一方面，对动能的掺混需要附加能量，这可能会大幅的增加能量消耗支出。最理想的情况下，是调用尽可能少的能量（如电能），获得最大的湍流效应来增加混合效果。

在下文中，会发现湍流的数值模拟需要引入大量的模型，并且会导致计算不稳定。但为什么大家都对湍流如此感兴趣？这是因为飞机要上天、要对天气预报进行预测，类似的过程都是湍流的。正所谓处处皆湍流。

最早对湍流进行研究的是 Reynolds。1895 年左右，Reynolds 提出了一种时间平均方法，因此，在后期相当一定的时间内，雷诺平均主要指的是时间平均。1877 年，Boussinesq 参考分子梯度扩散的数学形式，引入涡粘的概念。随后，Reynolds 和 Boussinesq 在湍流领域一夜成名，一直在现在，对两人的研究只需要点到为止，都不需要引用。

但是 Reynolds 和 Boussinesq 并没有对湍流模型进行求解。在 19 世纪，湍流模型的结果一直是一个迷。直到 1904 年，Prandtl 发现了边界层的概念（原始文献为德语版），并于 1925 年提出了混合长模型（德语版），并可以通过混合长来计算湍流粘度。在随后的近 30 年中，混合长湍流模型成为了其他湍流模型的理论基础。当代，Prandtl 的混合长模型通常被称之为零方程湍流模型。顾名思义，零方程湍流模型中，不存在传输方程，仅仅通过代数关系式就可以获得湍流粘度。进一步的，Prandtl 在 1945 年，提出了一种一方程湍流模型（德语版），其将湍流粘度与湍流动能的概念建立起来，并组建一个偏微分方程组来对湍流动能进行传输。

与此同时, Kolmogorov 认为除了应该考虑湍流动能, 还应该考虑一个变量 ω 。这个变量表示的是每单位体积单位时间的能量耗散。 ω 的倒数表示湍流的时间尺度。 $k\omega$ 则类似于湍流耗散率 ε 。1942 年, Kolmogorov 构建了 k 和 ω 的偏微分方程, 这是最早的一种两方程湍流模型。很遗憾, 当时并没有计算机能求解这种湍流模型。除了这些湍流模型之外, 最早的二阶矩湍流模型可以追溯到 Rotta 在 1951 年的工作(德语版)。二阶矩湍流模型也就是雷诺应力传输模型。所有这些模型都是在 1955 年之前提出的。

1960 年代之后, 大量的湍流模型开始涌现, 并可以通过计算机进行求解。2020 年, 如果将所有的湍流模型进行归类, 湍流的数值研究方法主要有三种: 直接模拟(Direct Numerical Simulation, DNS), 大涡模拟(Large Eddy Simulation, LES)以及雷诺平均法(Raynolds-averaged Navier-Stokes, RANS)。本节对这三种方法进行简述。

6.1.1 直接模拟 DNS

直接模拟是对湍流进行数值研究最精确的方法。直接模拟直接求解 N-S 方程, 不附加任何湍流模型。因此, 从概念上理解的话, 直接模拟是一种最简单的方法。上升到实践, 如果采用 OpenFOAM 进行直接模拟的话, 只需要将流动设置为层流即可。为了精确的捕获到流动中的湍流, 需要注意计算域大小。首先, 计算域的整体大小, 应该是流动中可能存在的最大的涡的尺度的数倍以上。同时, 为了捕获所有的能量耗散, 网格尺度还需要与最小的粘性尺度相符合。这种最小的尺度通常被称之为 Kolmogorov 尺度。

直接模拟预测的结果包含了大量的信息。例如, 图6.1为采用直接模拟捕获的近壁条带。但是, 这些信息并不是所有都至关重要。对于这种耗时且极为占据存储空间的数值方法, 直接模拟目前主要在学术界采用。随着计算机能力的提升, 2020 年, 一些个人工作站已经也可以进行一些直接模拟计算。

对于直接模拟的求解方法。有限体积法由于精度问题受限。主流的方法是有限差分法与谱方法。不管采用什么样的方法, 需要注意的就是方法的精度。例如, 对于一阶迎风格式, 其在数学上等于在方程中添加了一个数值耗散项。这种耗散项对算法有稳定性作用。对于稳态问题, 这种耗散可能无关紧要。但对于瞬态问题, 这种人工添加的耗散可能不会捕获尖锐的速度梯度导致直接模拟的结果失真, 丧失准确性。

另外一个极为重要的问题就是直接模拟的边界条件设置。对于初始条件, 其必须包含足够的流场细节。最好的情况下, 是给定一个从实验获得的真实的流场。但这是非常困难的。同样的, 对于边界条件, 尤其是进口边界条件, 必须包含足够的流场信息来“形成”湍流。

出口边界条件相对简单。通常采用的边界条件就是法向梯度为 0。但, 另外一种方式是使用一种对流非稳态出口边界条件。比如, 对 ϕ 变量, 可以调用下述边界条件:

$$\frac{\partial \phi}{\partial t} + \mathbf{U} \cdot \nabla \phi = 0 \quad (6.1)$$

在开源 CFD 软件 OpenFOAM 中，上述边界条件被称之为无反射边界条件¹。在壁面处，可以使用无滑移边界条件。但是通常在壁面处，会存在一些细微的非常重要的“条带”结构。这些结构需要非常精细的网格才能处理。另外需要注意的是，对称边界条件通常是不适用于直接模拟与大涡模拟的。因为虽然在雷诺平均中，可以将壁面当做对称面来处理，但是在直接模拟中，对称面的脉动速度必然不是对称的。

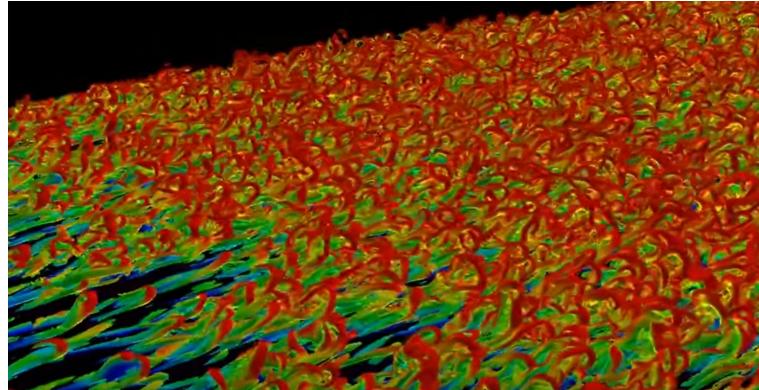


图 6.1: 湍流近壁条带结构 [10]。

6.1.2 大涡模拟 LES

湍流中的大涡往往比小涡更具有能量。这样来看，或许可以直接对大涡进行非模化的跟踪，将小涡模化处理。这就是大涡模拟的思想起源。大涡模拟同样是三维的并且是随时间瞬态变化的。在直接模拟不现实的情况下，通常可以采用大涡模拟来捕获流场结果。在一些领域（如气候科学、湍流燃烧等领域），大涡模拟已经成为不可或缺的研究方法。

不同于雷诺平均法，大涡模拟起始于空间滤波技术。各种大涡模拟方法都要确定一个 Δ ，其被称之为截断尺度（cutoff width），也称为滤波尺度（filtered width）。大于截断尺度的涡全部被解析，在空间滤波的过程中，小于截断尺度的涡全部被模化。在算法的角度，大涡模拟区分为不同的计算方法，如隐性滤波大涡模拟（Implicit filtered LES）、显性滤波大涡模拟（Explicit filtered LES）等。在隐性滤波大涡模拟中，最终植入的算法并不需要显性的进行滤波操作。最终的计算方程，隐性滤波大涡模拟与非稳态雷诺平均类方法（如大涡模拟的 Smagorinsky 模型与雷诺平均的 kEpsilon 模型），区别仅仅在于湍流粘度的模化。且雷诺平均的湍流粘度的大小远远大于隐性滤波大涡模拟。

与隐性滤波大涡模拟不同的是显性滤波大涡模拟，也即动态类模型。Germano et al. [38] 最早的意识到了 Smagorinsky 模型的模型参数需要依据流态的变化而变化。因此其提出一种动态的模型，其模型参数为一个空间相关可变的参数。其也被称之为显性滤波大涡模拟（Explicit filtered LES）。在显性滤波大涡模拟中，需要对 NS 方程应用两次滤波操

¹ 在[这里](#)有一些关于无反射边界条件在 DNS 模拟中的讨论。

作，第一次滤波操作等同于隐性滤波大涡模拟，这一步的滤波操作通常被称之为网格滤波（grid filter），第二次的滤波操作通常被称之为测试滤波（test filter）。测试滤波的尺度通常要大于网格滤波的尺度，并且会对湍流场产生附加的光顺作用。在第一次滤波方程形成之后，显性滤波大涡模拟再次应用测试滤波，会形成一套完全一样的方程。只不过变量均为测试滤波之后的变量，且亚格子应力发生变化。动态类大涡模拟模型如 dynamicKEqn 模型即为显性滤波大涡模拟。显性滤波大涡模拟需要显式的指定滤波方法，如盒子滤波或者高斯滤波之类。一些研究表示（如图6.2所示），显性滤波大涡模拟由于可以将 Δ 固定下来，因此可以达到网格无关解 [19]。隐性滤波大涡模拟之所以被称之为隐性，是因为没有第二次滤波模型的存在。且其第一次滤波操作通过离散格式以及网格大小来实现。隐性滤波大涡模拟中本身的亚格子模型已经附带了数值耗散。因此建议使用不附加耗散的数值格式，比如中心格式之类。目前大量的大涡模拟方法倾向于使用隐性滤波大涡模拟，显性滤波大涡模拟在学术上则使用较多。本文下述内容主要讨论隐性滤波大涡模拟。

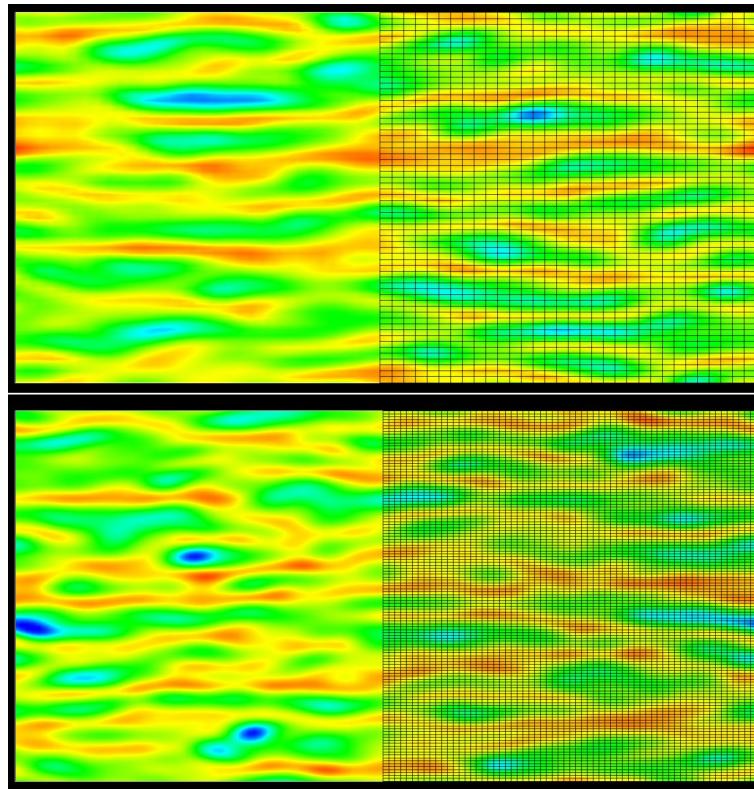


图 6.2: 显性滤波大涡模拟网格无关解 [19]。

大涡模拟需要定义一个速度变量，这个速度变量包含且仅仅包含大尺度的结构。因此，这个速度变量通常被称之为滤波速度（filtering velocity）。滤波速度跟滤波尺度 Δ 有关。大涡模拟认为尺度大于 Δ 就是大涡，尺度小于 Δ 就是小涡。大涡直接解析，小涡需要模化。

不同于直接模拟，由于大涡模拟需要进行模化，因此需要对控制方程进行推导。大涡

模拟中，最重要的模型在于将亚格子应力进行模化（Subgrid-Scale Stress, SGS）。

对于隐性滤波大涡模拟，在提升网格分辨率的情况下，截断误差以及亚格子应力模化的误差都随之减小。因此，隐性滤波大涡模拟不存在网格无关性这一说。跟直接模拟方法一样，大涡模拟的边界条件以及初始条件均异常敏感，需要复杂的方法进行给定。

最后值得一提的是，还有另外一种大涡模拟方法，其被称之为 MILES (Monotone Integrated Large Eddy Simulation)。Boris et al. [18] 最早的提出了 MILES 的概念，其将有界的通量修正算法 (Flux Corrected Transport) 与三阶格式相结合来实现大涡模拟的目的。需要注意的是，在 MILES 中不需要显性的调用亚格子模型。因此，MILES 与传统的 LES 的区别主要在于：1) MILES 的对流项格式需要通过通量修正算法来计算，2) MILES 不需要调用显性的亚格子模型。在 Fureby and Grinstein 的研究中 [36]，系统的对比了 MILES 与传统 LES 方法的数值等同性本源。

接下来我们看大涡模拟的推导过程。在大涡模拟中，需要定义一个滤波函数，目前有限体积法，使用最广也最简单的滤波函数为 Tophat 滤波：

$$\begin{aligned} G(\Delta) &= \frac{1}{\Delta^3}, \mathbf{x} \subset \Delta V, \\ G(\Delta) &= 0, \mathbf{x} \notin \Delta V \end{aligned} \quad (6.2)$$

其他还存在如主要应用于有限差分的高斯滤波，主要应用于谱方法的谱滤波方法（能谱在波长为 Δ/π 处精准截断）。大涡模拟的控制方程在推导的过程中，需要对变量进行滤波后并积分。考虑 Tophat 滤波函数，有滤波后的变量的定义：

$$\bar{\phi} = \int_V G(\Delta) \phi dV = \frac{1}{\Delta^3} \int_V \phi dV \quad (6.3)$$

$$\overline{\frac{\partial \phi}{\partial x}} = \int_V G(\Delta) \frac{\partial \phi}{\partial x} dV = \frac{1}{\Delta^3} \int_V \frac{\partial \phi}{\partial x} dV \quad (6.4)$$

如果 ΔV 不变，方程(6.5)中可以将微分符号提取出来。但在有限体积法中，网格大小一般是变化的。然而，同样可以将微分符号提取出来，只不过其增加了截断误差 ΔV^2 。这是可以接受的，因为本身传统有限体积法也充其量最高二阶精度。因此，方程(6.5)可以继续写为 [39]：

$$\overline{\frac{\partial \phi}{\partial x}} = \frac{1}{\Delta^3} \int_V \frac{\partial \phi}{\partial x} dV = \frac{\partial}{\partial x} \left(\frac{1}{\Delta^3} \int_V \phi dV \right) + O(\Delta V^2) = \frac{\partial \bar{\phi}}{\partial x} \quad (6.5)$$

因此，不可压缩大涡模拟的控制方程，在进一步进行滤波操作后有：

$$\nabla \cdot \bar{\mathbf{U}} = 0, \quad (6.6)$$

$$\frac{\partial \bar{\mathbf{U}}}{\partial t} + \nabla \cdot (\bar{\mathbf{U}} \bar{\mathbf{U}}) = -\frac{1}{\rho} \nabla \bar{p} + \nabla \cdot \nu (\nabla \bar{\mathbf{U}} + \nabla \bar{\mathbf{U}}^T), \quad (6.7)$$

需要注意的是方程(6.7)中的对流项 $\nabla \cdot (\bar{\mathbf{U}} \bar{\mathbf{U}})$ 的存在是因为 $\bar{\mathbf{U}} \bar{\mathbf{U}} \neq \bar{\mathbf{U}} \bar{\mathbf{U}}$ 。为了将 $\bar{\mathbf{U}} \bar{\mathbf{U}}$ 在方程中体现，可以简单的在方程(6.7)左右两边将这一项加上：

$$\frac{\partial \bar{\mathbf{U}}}{\partial t} + \nabla \cdot (\bar{\mathbf{U}} \bar{\mathbf{U}}) = -\frac{1}{\rho} \nabla \bar{p} + \nabla \cdot \nu (\nabla \bar{\mathbf{U}} + \nabla \bar{\mathbf{U}}^T) - \nabla \cdot (\bar{\mathbf{U}} \bar{\mathbf{U}}) + \nabla \cdot (\bar{\mathbf{U}} \bar{\mathbf{U}}) \quad (6.8)$$

将亚格子应力写为 τ_{sgs} :

$$\tau_{sgs} = \bar{\mathbf{U}} \bar{\mathbf{U}} - \bar{\mathbf{U}} \bar{\mathbf{U}} \quad (6.9)$$

方程(6.8)可以写为：

$$\frac{\partial \bar{\mathbf{U}}}{\partial t} + \nabla \cdot (\bar{\mathbf{U}} \bar{\mathbf{U}}) = -\frac{1}{\rho} \nabla \bar{p} + \nabla \cdot \nu (\nabla \bar{\mathbf{U}} + \nabla \bar{\mathbf{U}}^T) - \nabla \cdot \tau_{sgs} \quad (6.10)$$

各类大涡模拟模型，就在于将如何将亚格子应力 τ_{sgs} 进行模化。

τ_{sgs} 本质上可以分解为不同的贡献，其可以被分解为：

$$\tau_{sgs} = \overline{(\bar{\mathbf{U}} + \mathbf{U}')(\bar{\mathbf{U}} + \mathbf{U}')} - \bar{\mathbf{U}} \bar{\mathbf{U}} = (\bar{\mathbf{U}} \bar{\mathbf{U}} - \bar{\mathbf{U}} \bar{\mathbf{U}}) + (\bar{\mathbf{U}} \mathbf{U}' + \mathbf{U}' \bar{\mathbf{U}}) + \mathbf{U}' \mathbf{U}' \quad (6.11)$$

其中第一个括号内的项通常被称之为 Leonard 应力项，Leonard 在其工作中表示这一项可以通过解析的滤波速度场计算出来，不需要模化 [52]（在这里就涉及到了显性滤波大涡模拟）。第二项通常被称之为交叉项（cross term），其与能量从解析的流程传递到模化的流场有关。这一项表示能量的交互作用。第三项一般称之为亚格子雷诺应力项，表示模化后小涡之间的交互作用。

在对 τ_{sgs} 进行模化的过程中，可以将其分解为 3 项单独进行模化。但是目前主流的方式是将 τ_{sgs} 作为单独的一项作为模化，也即上文提到的隐性滤波大涡模拟。最直观的方法就是采用 Bousinessq 方法来处理：

$$\tau_{sgs} = -\nu_{sgs} (\nabla \bar{\mathbf{U}} + \nabla \bar{\mathbf{U}}^T) + \frac{2}{3} \nu_{sgs} (\nabla \cdot \bar{\mathbf{U}}) \mathbf{I} \quad (6.12)$$

其中 ν_{sgs} 表示大涡粘度。在 Smagorinsky 湍流模型中，大涡粘度可以表示为

$$\nu_{sgs} = C_{sgs} \Delta \sqrt{k_{sgs}} \quad (6.13)$$

其中 k_{sgs} 表示亚格子动能， C_{sgs} 表示无经验常数。对于各向同性湍流， $C_{sgs} \approx 0.2$ 。然而对于不同的雷诺数以及流形， C_{sgs} 倾向于并不是一个常数。例如，在开源 CFD 软件 OpenFOAM 中， C_{sgs} 的值被定义为 0.095。在这种情况下，大涡粘度小了一倍之多。在槽道流靠近壁面的区域， C_{sgs} 需要更小。这也就是 van Driest 阻尼模型的思想本源。另一种方式是，定义一个随时间与位置变化的 C_{sgs} ，这就是动态亚格子应力模型的出发点。

对于 Germano et al. 最早提出的显性滤波大涡模拟思想（也被称之为动态模型）[38]，其可以应用与任意一个隐性滤波大涡模拟之上。其更是一种模化思想。在动态类模型中，

其中的亚格子应力系数通过计算出来，非一个非均一的值，也即动态的。动态类模型在方程(6.10)的基础之上，再次进行一次滤波（test filter）有：

$$\frac{\partial \tilde{\bar{\mathbf{U}}}}{\partial t} + \nabla \cdot (\tilde{\bar{\mathbf{U}}} \tilde{\bar{\mathbf{U}}}) = -\frac{1}{\rho} \nabla \tilde{p} + \nabla \cdot \nu (\nabla \tilde{\bar{\mathbf{U}}} + \nabla \tilde{\bar{\mathbf{U}}}^T) - \nabla \cdot \mathbf{T}_{sgs} \quad (6.14)$$

其中

$$\mathbf{T}_{sgs} = \bar{\mathbf{U}} \bar{\mathbf{U}} - \tilde{\bar{\mathbf{U}}} \tilde{\bar{\mathbf{U}}} \quad (6.15)$$

后续的操作在这里不详细介绍，感兴趣的读者可以参考 Germano et al. 的原始论文 [38]。显性滤波大涡模拟仅存的自定义参数即为 test 滤波以及原始网格滤波（grid filter）的比值。Germano et al. [38] 建议其值为 2。下面我们证实两种滤波操作的区别。在一维情况下，网格 P 点的 test 滤波变量可以计算为：

$$\begin{aligned} \tilde{\bar{\phi}}_P &= \frac{1}{2\Delta x} \int_W^E \bar{\phi} dx = \frac{1}{2\Delta x} \left(\int_W^P \bar{\phi} dx + \int_P^E \bar{\phi} dx \right) = \frac{1}{2\Delta x} (\bar{\phi}_w \Delta x + \bar{\phi}_e \Delta x) \\ &= \frac{1}{2} \left(\frac{\bar{\phi}_W + \bar{\phi}_P}{2} + \frac{\bar{\phi}_E + \bar{\phi}_P}{2} \right) \end{aligned} \quad (6.16)$$

很明显， $\tilde{\bar{\phi}}_P \neq \bar{\phi}_P$ 。在高维的情况下，网格 P 点的 test 滤波变量会涉及到更多的网格面。

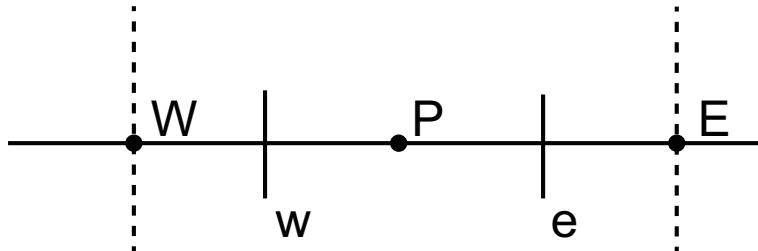


图 6.3: test 滤波与网格滤波。实线表示网格滤波的有限体积，虚线表示 test 滤波的有限体积。

6.1.3 雷诺平均 RANS

雷诺平均法目前是工程上应用最广泛的方法。例如，很多工程师对一些物体的升阻力、流体的混合程度感兴趣。在这种情况下使用直接模拟和大涡模拟就有些大材小用了。雷诺平均法也存在稳态与瞬态一说。对于稳态算例，所有的非稳态因素都被平均的过程抹掉了。对于瞬态算例，雷诺平均法应捕获相应的瞬态因素。参考图6.4，稳态的雷诺平均通常采用时间平均法，非稳态的雷诺平均通常采用集合平均法。

在进行雷诺平均的过程中，同样需要对模型进行封闭。再一次的，Boussinesq 假定发挥了重要的作用。Boussinesq 假定将未封闭的湍流应力与湍流粘度联合起来，这样，就把未知量转化为了湍流粘度变量。在获得湍流粘度之后，N-S 方程即可进行求解。

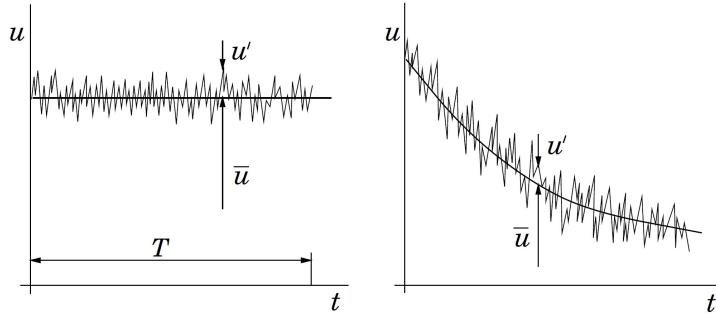


图 6.4: 时间平均和集合平均 [34]。

目前存在大量的湍流粘度封闭方法。如线性涡粘法、非线性涡粘法等。在线性涡粘法中，湍流粘度可以通过一个代数方程求出。比如，可以将湍流粘度与湍流动能以及湍流动能耗散率联合起来：

$$\nu_t = C_\mu \frac{k^2}{\varepsilon} \quad (6.17)$$

其中 C_μ 为模型自定义常数， k 为湍流动能， ε 为湍流动能耗散率。进一步的，其中的 k 与 ε 可以通过 $k - \varepsilon$ 湍流模型进行求解获得。除此之外，还有大量的其他模型可用于求解湍流粘度，如 Spalart-Allmaras 模型、 $k - \omega$ 模型等。依据未知的变量数，这些模型又可以分为一方程模型（如 Spalart-Allmaras 模型具有一个未知变量）、二方程模型（二个未知变量，如 $k - \omega$ 模型）等。

不同于大涡模拟与直接模拟，雷诺平均法对初始条件以及进口边界条件没有那么敏感。实际操作中，使用固定值进口边界条件以及零法向梯度出口边界条件即可。对于壁面，则需要调用壁面函数。在一些情况下，壁面函数对流场结果是高度相关的。同时，还存在一些不需要壁面函数的雷诺平均模型，比如 Lam-Bremhorst $k - \varepsilon$ 湍流模型 [48]。这些模型也通常被称之为低雷诺数湍流模型。

大涡模拟与雷诺平均法在使用上具有以下差别：

- 大涡模拟更加适用于分离流、钝体绕流（包含复杂混乱的湍流结构）、以及湍流转捩；
- 雷诺平均法将所有的湍流尺度进行了模化，大涡模拟仅仅对小尺度涡进行模化，对大尺度涡进行解析；

6.1.4 二维 DNS/LES

目前学术界普遍的看法是：“无三维涡，则无湍流”。“只有在三维的情况下，流体自身的涡旋才能伸展。”但是大量的文献依旧通过二维大涡模拟、甚至二维直接模拟进行相关研究。例如，Bouris and Bergeles 的文章中表示 [20]

“针对本文的算例，确实大量的文献表明大涡模拟只能进行三维计算，因为二维模拟缺少相关的涡旋特征。但是，本文的研究同样表明，这样一棒子打死二维大涡模拟也不是那么容易的。我们的结果表示二维大涡模拟相对于任何的雷诺平均法都优越。然而，从物理本质来考虑，除非计算机硬件存在限制，最好还是进行三维大涡模拟。”

在 Roohi et al. 的这篇文章中，作者们表示 [77]:

“本文使用二维大涡模拟的原因主要是其可以节省计算资源。正如我们的研究在 3.3 节中讨论到，针对非稳态空化流动，雷诺平均法获得的流场并不让人满意。因此，二维大涡模拟必然比雷诺平均更为优越。并且，前人也有大量的研究表明，二维大涡模拟可以获得跟实验值非常贴近的结果。因此，对于某些特定的情况，二维大涡模拟是精准的。很明显，我们的模拟结果和实验的符合也验证了这个说法。”

同样，一些研究学者非常激进，比如 Breuer 在研究中表示 [21]:

“我们的研究表明二维的大涡模拟，甚至直接模拟完全是扯淡的。这是因为物理上流动的物质相互影响是三维的，即使是那些看起来非常二维的计算流域也是如此。”

总体来看，目前学术界，对于二维直接模拟/大涡模拟的观点，尚不统一，仍需探索。

6.1.5 RANS-LES 混合模型

RANS 类模型添加瞬态项可以演变为非稳态模式，也通常被称之为 URANS 模型。然而，即使网格分辨率足够高，时间步长足够小，URANS 同样不能提供精准的湍流能谱。目前学术界普遍认为这是由于 RANS 进行的平均操作所导致。因此，URANS 只能解析那些非常低频的涡以及大尺度流动变化。参考图 6.5，对于类似的圆柱绕流，URANS 无论如何都并不能够预测圆柱后的各种频率的涡结构，仅仅能够预测低频的涡脱落，且基本为相同频率。完全达不到真正瞬态模拟的解析程度。Young 等的研究也表明 URANS，即使是三维的，其预测的结果也大体与二维 URANS 类似，并没有显著提高 [99]。

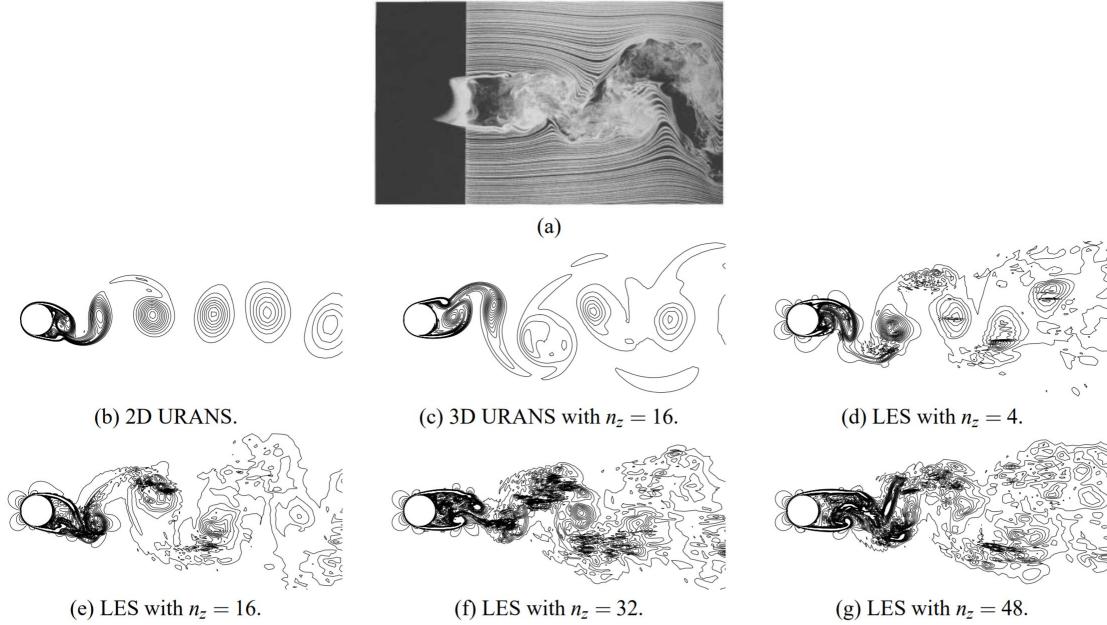


图 6.5: a): 圆柱绕流实验检测流场。b-g): 不同湍流模型预测的涡等高线图 (0.5-10) [99]。

为了克服 URANS 的这个问题，可以调用尺度自适应模拟 (Scale-Adaptive Simulation, SAS)。SAS 的概念基于 URANS，区别在于在 URANS 的基础之上，增加一个源项。这个源项的物理作用将对湍流能谱进行区分，并进一步的识别不同频率的涡。举例：对于 kOmegaSSTSAS 模型，定义 von Karman 尺度 L_{vk} ，并且将源项与 von Karman 尺度相关联。如果 ω 方程的源项足够大，会导致 ω 增大，同时湍流粘度降低 ($\nu_t = k/\omega$)，在这种情况下的粘度阻尼效应降低，高频率涡将可以很好地保持 [83]。

除了 SAS 之外，分离涡模拟 (Detached Eddy Simulation, DES) 也是非常常用的混合类模型。分离涡模型认为使用大涡模拟进行全场的涡旋捕获是没必要的且不可能的。这是因为壁面处由于湍流尺度通常会变的非常小，如果调用 LES 则需要铺设大量的网格资源。因此分离涡模型在壁面处使用 RANS 进行计算，在主流区采用 LES。以 SpalartAllmaras 模型举例，SpalartAllmaras 模型²中的 $\tilde{\nu}$ 存在耗散项 $C_{w1}f_w(\tilde{\nu}/y)^2$ ，其中 y 表示壁面距离。SpalartAllmaras 模型中的耗散项与产生项的整体作用³将 $\tilde{\nu}$ 正比于 Sy^2 。类似的，如果将 $\tilde{\nu}$ 与网格尺度 Δ 联系起来，即令 $\tilde{\nu}$ 正比于 $S\Delta^2$ 。这种思想就是 DES 的本源。在 Spalart 等的文章中，将 y 重新定义为 \tilde{y} [81]：

$$\tilde{y} = \min(y, C_{DES}\Delta) \quad (6.18)$$

其中 C_{DES} 为模型参数。在这种情况下，如果 $y > C_{DES}\Delta$ ，模型演变为 LES。如果 $y < C_{DES}\Delta$ ，模型为标准 SpalartAllmaras 模型。考虑整个计算域，尤其是在边界层区域，通

² 模型方程可以参考附录一节。

³ S 的定义参考附录中的 SpalartAllmaras 模型。

常会存在高度各向异性的网格单元。Spalart 等的工作将 Δ 定义为网格三个维度中的最大尺度：

$$\Delta = \max(\Delta_x, \Delta_y, \Delta_z) \quad (6.19)$$

因此，在壁面附近，通常 $y \ll \Delta$ ，即壁面调用标准 SpalartAllmaras 模型。在主流区，通常网格倾向为各向同性，并且主流区的壁面距离通常很大，即 $y \gg \Delta$ ，这时，则演变为 LES。

然而有些情况下，在层流边界层内，也会导致 DES 演变为 LES，大幅度的降低湍流粘度，导致涡的产生。这种现象被称之为网格引致分离现象（Grid-Induced Separation, GIS）。在 DES 束手无策的情况下，延迟分离涡模拟（Delayed-DES, DDES）可以很好的解决这个问题。DDES 在 DES 的基础上，附加一个开关函数，强制在边界层内调用 RANS 模拟，在边界层外则为 DES。

6.2 结构网格与非结构网格

结构网格与非结构网格的区别主要在于网格节点的编号形式。结构网格的网格节点通常可以按照次序排列，对于三维网格，通常按照 i, j, k 的形式排列。例如 y 方向上游与下游的网格序号可以表示为 $i, j - 1, k$, $i, j + 1, k$ 等，上游的上游节点可以表示为 $i, j - 2, k$ 。由于结构网格寻址方便，因此在结构网格上很容易拓展高阶格式。非结构网格则不限定于网格编号形式。其通过复杂的网格连接性来定义网格编号，例如需定义网格节点序号、定义网格节点与面相连接的序号等。结构网格思想更偏向于有限差分。有限体积法则不限定于任何网格类型，更适用于非结构网格。

由于有限差分法诞生较早，有限体积法诞生较晚。有限差分主要使用规则的结构网格来进行。同时，对于方程离散的各个项，有限差分的方式更易于理解。因此，CFD 教材中大量的采用结构网格与有限差分结合的方式进行介绍。现存商业/开源 CFD 软件为了追求普适性，很少使用结构网格的方式进行寻址。在这里强调一些常见的误区：

- 结构网格单元不一定是四边形或六面体。结构网格单元也可以是三角形等。如图6.6所示，其中的三角形网格也可以用 i, j, k 进行标识。
- 四边形、六面体网格也可能是非结构网格。如图6.7所示，是一种六面体非结构网格，由于几何中心部位进行了糙化处理，无法通过 i, j, k 进行寻址，这是一种多块六面体非结构网格。

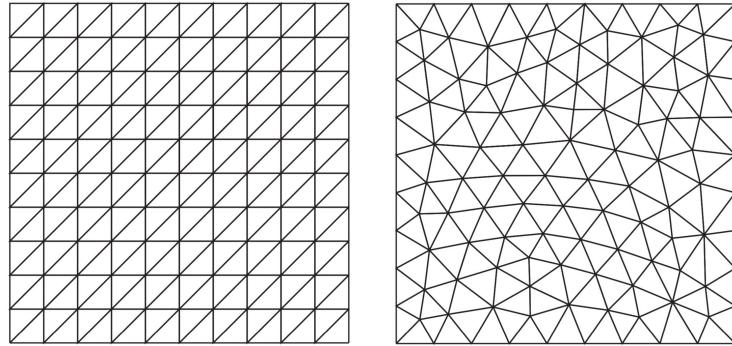


图 6.6: 三角形结构网格 (左) 与非结构网格 (右)。

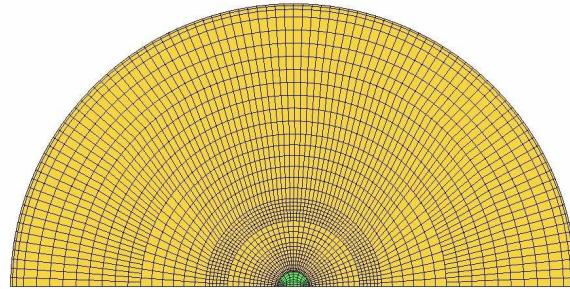


图 6.7: 六面体非结构网格 [37]。

6.3 动理学方程

6.3.1 麦克斯韦分布与高斯分布

正如第一章中所指出的, N-S 方程并不是描述流动的最底层的形式。更为底层的数学模型为玻尔兹曼方程。其在不同的领域有不同的称呼。例如在群体平衡模型研究领域, 被称之为普适性群体平衡模型, 在空气动力学领域被称之为动理学方程, 在喷雾燃烧领域被称之为威廉玻尔兹曼方程。三维的玻尔兹曼方程可以写为

$$\frac{\partial f}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{U} f) + \nabla_{\mathbf{U}} \cdot (\mathbf{A} f) = \mathbb{C}, \quad (6.20)$$

其中 f , 即 $f(t, \mathbf{x}, \mathbf{U})$, 表示关于气体分子的速度分布函数, \mathbf{U} 表示气体分子的速度, \mathbf{A} 表示受力引起的加速项, \mathbb{C} 表示碰撞项。

本节我们主要关心 f 。既然 f 为分布, 其就可以选用不同的分布形式。例如高斯分布、

log-normal 分布等。在动理学研究领域，可认为气体分子的运动呈现麦克斯韦分布，即⁴：

$$f = \frac{\rho}{(2\pi\theta)^{3/2}} \exp \frac{-(u - \bar{u})^2 - (v - \bar{v})^2 - (w - \bar{w})^2}{2\theta} \quad (6.21)$$

其中 $\bar{u}, \bar{v}, \bar{w}$ 表示平均速度的三个分量， θ 表示除掉密度后的压力，在统计上表示方差。

同时， f 也可以假定其为高斯分布

$$f = \frac{\rho}{(2\pi)^{3/2}\sqrt{|\boldsymbol{\theta}|}} \exp \left(-\frac{1}{2} \begin{bmatrix} u - \bar{u} \\ v - \bar{v} \\ w - \bar{w} \end{bmatrix}^T \cdot \boldsymbol{\theta}^{-1} \cdot \begin{bmatrix} u - \bar{u} \\ v - \bar{v} \\ w - \bar{w} \end{bmatrix} \right) \quad (6.22)$$

其中 $\boldsymbol{\theta}$ 为对称二阶张量：

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \quad (6.23)$$

其中 $\theta_{ij} = \theta_{ji}, i \neq j$ 。且有压力：

$$p = \rho \frac{\theta_{11} + \theta_{22} + \theta_{33}}{3} \quad (6.24)$$

麦克斯韦分布为高斯分布的各向同性形式。例如，若假定 $\boldsymbol{\theta}$ 的非对角元素为 0，同时认为 $\theta_{11} = \theta_{22} = \theta_{33} = \theta$ 。高斯分布则演变为麦克斯韦分布。同时，如果考虑一维的情况，高斯分布即麦克斯韦分布。

二维情况下的高斯分布可以写为

$$f = \frac{\rho}{2\pi\sqrt{|\boldsymbol{\theta}|}} \exp \left(-\frac{1}{2} \begin{bmatrix} u - \bar{u} \\ v - \bar{v} \end{bmatrix}^T \cdot \boldsymbol{\theta}^{-1} \cdot \begin{bmatrix} u - \bar{u} \\ v - \bar{v} \end{bmatrix} \right) \quad (6.25)$$

统计学领域更常见的写法是将 $\boldsymbol{\theta}$ 替换为符号 $\boldsymbol{\Sigma}$ ：

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{pmatrix} = \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \quad (6.26)$$

在这种情况下， $\boldsymbol{\Sigma}$ 中的相干系数 ρ' 与 $\boldsymbol{\theta}$ 的关系为

$$\rho' = \frac{\theta_{12}}{\sqrt{\theta_{11}\theta_{22}}}, \theta_{11} = \sigma_1^2, \theta_{22} = \sigma_2^2, \quad (6.27)$$

⁴如果将 θ 和温度联系起来： $\theta = kT/m = p/\rho$ ，有：

$$f = \rho \left(\frac{m}{2\pi kT} \right)^{3/2} \exp \frac{m}{kT} \frac{-(u - \bar{u})^2 - (v - \bar{v})^2 - (w - \bar{w})^2}{2}$$

同时， $2\pi\theta$ 的幕中的 $3/2$ 中的 3 表示三维。也可以另 $\lambda = m/2kT$ ，则有：

$$f = \rho \left(\frac{\lambda}{\pi} \right)^{3/2} \exp \left(-\lambda ((u - \bar{u})^2 - (v - \bar{v})^2 - (w - \bar{w})^2) \right)$$

同时，本文讨论的均为单原子情况。在双原子情况下，要考虑附加自由度的影响。

同时，方程(6.25)中 \exp 函数内的表达式也可以写为：

$$\begin{bmatrix} u - \bar{u} \\ v - \bar{v} \end{bmatrix}^T \cdot \boldsymbol{\Sigma}^{-1} \cdot \begin{bmatrix} u - \bar{u} \\ v - \bar{v} \end{bmatrix} = \frac{1}{1 - \rho'^2} \left(\frac{(u - \bar{u})^2}{\sigma_1^2} + \frac{(v - \bar{v})^2}{\sigma_2^2} - \frac{2\rho'(u - \bar{u})(v - \bar{v})}{\sigma_1\sigma_2} \right) \quad (6.28)$$

(6.25)中的分母为：

$$2\pi\sqrt{|\boldsymbol{\theta}|} = \sigma_1\sigma_2\sqrt{1 - \rho'^2} \quad (6.29)$$

这样，方程(6.25)可以重写为：

$$f = \frac{\rho}{2\pi\sigma_1\sigma_2\sqrt{1 - \rho'^2}} \exp \left(-\frac{1}{2(1 - \rho'^2)} \left(\frac{(u - \bar{u})^2}{\sigma_1^2} + \frac{(v - \bar{v})^2}{\sigma_2^2} - \frac{2\rho'(u - \bar{u})(v - \bar{v})}{\sigma_1\sigma_2} \right) \right) \quad (6.30)$$

6.3.2 矩方程、五矩模型（欧拉方程）、十矩模型

继上一节的讨论，一维形式的玻尔兹曼方程可以写为

$$\frac{\partial f(u)}{\partial t} + \frac{\partial u f(u)}{\partial x} + \frac{\partial A f(u)}{\partial u} = \mathbb{C} \quad (6.31)$$

其中 u 表示速度的 x 方向分量， A 表示 \mathbf{A} 在 x 方向上的分量。如果不考虑加速项与碰撞项，一维玻尔兹曼方程降级为如下形式：

$$\frac{\partial f(u)}{\partial t} + \frac{\partial u f(u)}{\partial x} = 0 \quad (6.32)$$

在这里引入矩的概念，这里讨论的矩即为统计学中的矩。 $f(u)$ 的 k 阶矩表示为：

$$m_k = \int u^k f(u) du \quad (6.33)$$

其低阶矩存在一定的物理意义。同时，其低阶矩还取决于具体的 $f(u)$ 形态。

第一种情况（最简单的情况），可以假定 $f(u)$ 的形态为狄拉克函数的加和：

$$f(u) = \sum \rho_i \delta(u - u_i) \quad (6.34)$$

在这种情况下， $f(u)$ 变为间断的函数。进一步的，间断的 $f(u)$ 的 0 阶矩表示为：

$$m_0 = \int f(u) du \equiv \rho \quad (6.35)$$

$f(u)$ 的 1 阶矩表示为：

$$m_1 = \int f(u) u du \equiv \rho \bar{u} \quad (6.36)$$

以及 2 阶矩表示为：

$$m_2 = \int f(u) u^2 du \equiv \rho \bar{u} \bar{u} \quad (6.37)$$

其中 \bar{u} 表示气体分子运动的平均速度。如果更进一步的假定 $f(u)$ 为单值分布函数，有：

$$f(u) = \rho_1 \delta(u - u_1) \quad (6.38)$$

其仅具有两个未知量，即 ρ_1, u_1 。其可以通过 2 个传输方程进行求解。如果对方程(6.32)取矩有

$$\begin{aligned} \partial_t m_0 + \partial_x m_1 &= 0 \\ \partial_t m_1 + \partial_x m_2 &= 0 \end{aligned} \quad (6.39)$$

依据矩的定义，方程(6.39)也即

$$\begin{aligned} \partial_t \rho_1 + \partial_x \rho_1 \bar{u}_1 &= 0 \\ \partial_t \rho_1 \bar{u}_1 + \partial_x \rho_1 \bar{u}_1 \bar{u}_1 &= 0 \end{aligned} \quad (6.40)$$

方程(6.40)中分别对应连续性方程和无压力无粘性力的动量方程。整个方程系统也被称之为无压力气体动力学方程。在矩方法研究领域也被称之为二阶矩模型 [24]。方程(6.40)呈现弱双曲特征。

第二种情况，如果假定 $f(u)$ 为双值分布函数，有：

$$f(u) = \rho_1 \delta(u - u_1) + \rho_2 \delta(u - u_2) \quad (6.41)$$

其具有 4 个未知量，即 ρ_1, u_1, ρ_2, u_2 。其可以通过 4 个传输方程进行求解。如果对方程(6.32)取矩有

$$\begin{aligned} \partial_t m_0 + \partial_x m_1 &= 0 \\ \partial_t m_1 + \partial_x m_2 &= 0 \\ \partial_t m_2 + \partial_x m_3 &= 0 \\ \partial_t m_3 + \partial_x m_4 &= 0 \end{aligned} \quad (6.42)$$

方程(6.42)构成四矩模型。相对于低阶矩模型（如二阶矩模型），高阶矩模型（如四阶模型）可以调用更多的速度节点（如 u_1, u_2 ）。在颗粒流领域，只有高阶矩模型才能预测颗粒轨迹交叉。

6.3.3 欧拉方程（一个推导实例）

上文将 $f(u)$ 假定为离散的狄拉克函数（思想与积分矩方法极为相似）。本节考虑更为现实的 $f(u)$ 形态。一维的动理学方程可以写为：

$$\frac{\partial f}{\partial t} + \frac{\partial u f}{\partial x} = \frac{1}{\tau} (g - f) \quad (6.43)$$

f 为粒子分布函数, g 表示平衡情况下的分布函数, τ 表示碰撞时间尺度。在这里假定 1) f 分布遵循麦克斯韦分布, 2) 碰撞均衡。因此有 $f = g$ 。这样, 方程(6.43)演变为

$$\frac{\partial g}{\partial t} + \frac{\partial ug}{\partial x} = 0 \quad (6.44)$$

首先对方程(6.44)取零阶矩有:

$$\frac{\partial m_0}{\partial t} + \frac{\partial m_1}{\partial x} = 0 \quad (6.45)$$

取一阶矩有:

$$\frac{\partial m_1}{\partial t} + \frac{\partial m_2}{\partial x} = 0 \quad (6.46)$$

取二阶矩有:

$$\frac{\partial m_2}{\partial t} + \frac{\partial m_3}{\partial x} = 0 \quad (6.47)$$

方程(6.45)、(6.46)、(6.47)共有三个矩方程, 在三维情况下, 有五个矩方程。因此, 其也被称之为五矩模型。同时, 依据矩的定义, g 的 0 阶矩表示为:

$$m_0 = \int g du \equiv \rho \quad (6.48)$$

1 阶矩表示为:

$$m_1 = \int g u du \equiv \rho \bar{u} \quad (6.49)$$

2 阶矩表示为⁵:

$$m_2 = \int g u^2 du \equiv \rho \bar{u} \bar{u} + \rho \theta \quad (6.50)$$

3 阶矩表示为:

$$m_3 = \int g u^3 du \equiv \rho \bar{u} \bar{u} \bar{u} + 3\rho \bar{u} \theta \quad (6.51)$$

其中 $\rho \theta$ 为一种温度相关量, 在颗粒流中表示颗粒温度, 在空气动力学中表示压力 p ⁶。因此有二阶矩的另外一种表达形式:

$$m_2 = \rho \bar{u} \bar{u} + p \quad (6.52)$$

同时有比能 E (机械能与比内能之和) 为:

$$\rho E = \frac{1}{2} \rho \bar{u} \bar{u} + \frac{1}{2} \rho \theta = \frac{1}{2} \rho \bar{u} \bar{u} + \frac{1}{2} p \quad (6.53)$$

同样, 若对三阶矩乘以 $1/2$, 有另外一种形式为:

$$\frac{1}{2} m_3 = \frac{1}{2} (\rho \bar{u} \bar{u} \bar{u} + 3\rho \theta) = (\rho E + p) u \quad (6.54)$$

⁵注意这里的二阶矩的定义区别于方程(6.37)。因为这里的 g 为麦克斯韦分布, 而不是狄拉克函数。

⁶由于 $\theta = kT/m$, 有 $p = \rho/(2\lambda)$ 。

综合上述方程，有主要变量形式的一维欧拉方程：

$$\begin{aligned}\partial_t \rho + \partial_x \rho \bar{u} &= 0 \\ \partial_t \rho \bar{u} + \partial_x \rho \bar{u} \bar{u} &= -\partial_x p \\ \partial_t \rho E + \partial_x \rho Eu &= -\partial_x pu\end{aligned}\tag{6.55}$$

第一个方程为密度方程，第二个方程为速度方程，第三个方程为能量方程。同理，在三维情况下，可获得五个矩方程（三维欧拉方程），分别对应密度、三个方向的速度方程以及能量方程。在五矩方程中，压力为各向同性（压力二阶张量的非对角线元素为0）。除了麦克斯韦分布， $f(u)$ 还可以假定为高斯分布，在这种情况下，三维情况下方程演变为 10 个矩方程，也被称之为十矩方程，十矩方程中压力为各向异性（压力二阶张量的非对角线元素不为0）。

总之，针对动理学方程，若假定不同的 f 分布函数，可以获得不同的矩方程（宏观方程）。

6.4 Von Neumann 稳定性分析与截断误差

6.4.1 变量分离法

对于方程求解，一种方法被称之为变量分离法 [69]。现在用一个特别简单的例子举例说明，考虑下述方程：

$$\frac{dy}{dx} = 5xy\tag{6.56}$$

可以进行移项有：

$$\frac{dy}{y} = 5x dx\tag{6.57}$$

方程(6.57)可以对方程左右两边同时积分并获得解。这是一个非常简单并且好理解的例子。

下面考虑一个抛物线偏微分方程组：

$$\frac{\partial \phi}{\partial t} = \frac{\partial}{\partial x} \left(\frac{\partial \phi}{\partial x} \right), t > 0, 0 < x < 1\tag{6.58}$$

其边界条件与初始条件为：

$$\begin{aligned}\phi(0, t) &= \phi(1, t) = 0 \\ \phi(x, 0) &= \phi^0(x), 0 < x < 1\end{aligned}\tag{6.59}$$

若使用分离变量法求解，我们首先假定 ϕ 函数具有下面的形式，即将关于 t 变量的函数与 x 变量的函数分开：

$$\phi(x, t) = f(x)g(t)\tag{6.60}$$

在这种情况下，对 $\phi(x, t)$ 求导数有：

$$\begin{aligned}\frac{\partial \phi}{\partial t} &= f \frac{\partial g}{\partial t} \\ \frac{\partial \phi}{\partial x} &= g \frac{\partial f}{\partial x} \\ \frac{\partial}{\partial x} \left(\frac{\partial \phi}{\partial x} \right) &= g \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right)\end{aligned}\tag{6.61}$$

在这种情况下，方程(6.58)即为：

$$\frac{1}{g} \frac{\partial g}{\partial t} = \frac{1}{f} \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right)\tag{6.62}$$

可以看出，方程(6.62)左侧是一个关于 t 的函数，右侧是关于 x 的一个函数，若想让其对任意 x, t 都满足，则必然其为一个常数。如果我们令这个常数为 $-k^2$ ，其有解为：

$$g = e^{-k^2 t}, f = A \sin kx + B \cos kx\tag{6.63}$$

即

$$\phi(x, t) = e^{-k^2 t} (A \sin kx + B \cos kx)\tag{6.64}$$

其中 A, B 为未知系数。很明显，其中 $\exp(-k^2 t)$ 中负号的作用，可以保证 $\phi(x, t)$ 在任何时间 t 下都是有界的。进一步，只要解符合方程(6.64)的形式，都满足条件。因此，下述方程

$$\phi(x, t) = \sum_{m=1}^{\infty} e^{-k_m^2 t} (A_m \sin k_m x + B_m \cos k_m x)\tag{6.65}$$

必然也是解。但其中存在若干的未知系数。这些未知系数可以通过初始条件以及边界条件来求得。

首先依据初始条件 $\phi(0, t) = 0$ ，有：

$$\sum_{m=1}^{\infty} e^{-k_m^2 t} B_m = 0\tag{6.66}$$

其解只能是 $B_m = 0$ 。在这种情况下，方程(6.70)变为：

$$\phi(x, t) = \sum_{m=1}^{\infty} e^{-k_m^2 t} A_m \sin k_m x\tag{6.67}$$

再次，通过 $\phi(1, t) = 0$ ，有：

$$\sum_{m=1}^{\infty} e^{-k^2 t} A_m \sin k_m = 0\tag{6.68}$$

其解只能是

$$k_m = m\pi\tag{6.69}$$

其中 m 为整数。这样，方程(6.67)变为：

$$\phi(x, t) = \sum_{m=1}^{\infty} A_m e^{-(m\pi)^2 t} \sin m\pi x \quad (6.70)$$

熟悉傅里叶理论的可以看出方程(6.70)恰巧是傅里叶 \sin 级数展开形式。其中的系数 A_m 可以通过初始条件逐级求出。因此有 [4]：

$$A_m = 2 \int_0^1 \phi^0(x) \sin m\pi x dx \quad (6.71)$$

在这种情况下，可获得 $\phi(x, t)$ 函数的精确解。需要注意的是。为了获得 A_m 的值，只有在 $\phi^0(x)$ 非常简单的情况下，方程(6.71)才具有精确解，否则，可能需要进行数值积分来获得数值解。再次，方程(6.70)中存在方程的叠加，只有在进行充分的叠加之后，才可能获得任意的型线。实际上，很难进行无穷次叠加，因此，必然存在误差。可以说，方程(6.70)在完美情况下，即 $\phi^0(x)$ 非常简单，同时进行无穷次叠加的情况下，可以维持精确解的情况。若不能满足，必然会有误差的出现。但是最重要的，获得如此的解是因为使用了变量分离法，但在很多情况下，偏微分方程并不能够使用变量分离法进行精确求解。

还需要注意的是，方程(6.58)还有另外一种解的形式为：

$$\phi(x, t) = e^{-k^2 t} \cdot e^{ikx} \quad (6.72)$$

其中 i 为虚数单位。对上述方程求导，可以证明其为方程(6.58)的一个精确解。

6.4.2 抛物线方程稳定性分析、无条件稳定

现在对 x 方向的网格进行均分，每个网格单元长度为 Δx ，那么，第 j 个网格单元的 x 为 $j\Delta x$ 。令第 j 个网格单元，第 n 个时间步的 ϕ_j^n 值为 $(t = n, x = j\Delta x)$

$$\phi_j^n = \lambda^n e^{ikj\Delta x} \quad (6.73)$$

因为 $\phi^{n+1}/\phi_j^n = \lambda$ ，因此其中的 λ 表示一个增长指数的概念，若方程的解为有界的，必然有

$$|\lambda| = |\phi_j^{n+1}/\phi_j^n| \leq 1 \quad (6.74)$$

否则 ϕ 将呈现无穷大。

现在对方程(6.58)的时间项进行一阶离散、空间项进行显性中心离散有：

$$\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} = \frac{\phi_{j+1}^n - 2\phi_j^n + \phi_{j-1}^n}{(\Delta x)^2} \quad (6.75)$$

参考方程(6.73)，将其代入到方程(6.75)有：

$$\frac{\lambda^{n+1} e^{ikj\Delta x} - \lambda^n e^{ikj\Delta x}}{\Delta t} = \frac{\lambda^n e^{ik(j+1)\Delta x} - 2\lambda^n e^{ikj\Delta x} + \lambda^n e^{ik(j-1)\Delta x}}{(\Delta x)^2} \quad (6.76)$$

整理后有：

$$\lambda = 1 + \frac{\Delta t}{(\Delta x)^2} (e^{ik\Delta x} - 2 + e^{-ik\Delta x}) \quad (6.77)$$

依据欧拉三角公式有：

$$\lambda = 1 - 2 \frac{\Delta t}{(\Delta x)^2} (1 - \cos k\Delta x) = 1 - 4 \frac{\Delta t}{(\Delta x)^2} \sin^2 \left(\frac{1}{2} k\Delta x \right) \quad (6.78)$$

可以看出，在

$$\Delta t \leq \frac{1}{2} (\Delta x)^2 \quad (6.79)$$

的情况下，可以满足 Von Neumann 稳定性 $|\lambda| \leq 1$ 。

可以看出，方程(6.79)施加的是一个非常严格的条件，需要时间步长非常小才能保证稳定。如果减小 Δx ，那么时间步长需要减小的更快。如果更换其他的离散格式，或许会有改变。在上文中使用的是时间显性离散方程。如果对空间项调用隐性中心格式离散，有：

$$\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} = \frac{\phi_{j+1}^{n+1} - 2\phi_j^{n+1} + \phi_{j-1}^{n+1}}{(\Delta x)^2} \quad (6.80)$$

参考上文的推导，有：

$$\lambda = \frac{1}{1 + 4 \frac{\Delta t}{\Delta x^2} \sin^2 \frac{1}{2} k\Delta x} \quad (6.81)$$

可以看出有 $0 < \lambda < 1$ 。因此，隐性格式是无条件稳定的。

6.4.3 截断误差

另外一种方式是采用泰勒展开的方法进行截断误差分析。考虑方程(6.75)，其中的变量 ϕ_j^{n+1} 可以理解为定义为 x_j, t^{n+1} 网格点的变量，那么其可以表示为：

$$\phi_j^{n+1} = \phi(x_j, t^{n+1}) = \phi(x_j, t^n + \Delta t) = \phi_j^n + \frac{\partial \phi}{\partial t} \Delta t + \frac{1}{2} \frac{\partial^2 \phi}{\partial t^2} \Delta t^2 + \mathcal{O}(\Delta t^2) \quad (6.82)$$

同理， ϕ_{j+1}^n 可以表示为：

$$\begin{aligned} \phi_{j+1}^n &= \phi(x_{j+1}, t^n) = \phi(x_j + \Delta x, t^n) = \phi_j^n + \frac{\partial \phi}{\partial x} \Delta x + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Delta x^2 + \frac{1}{6} \frac{\partial^3 \phi}{\partial x^3} \Delta x^3 \\ &\quad + \frac{1}{24} \frac{\partial^4 \phi}{\partial x^4} \Delta x^4 + \mathcal{O}(\Delta x^4) \end{aligned} \quad (6.83)$$

$$\begin{aligned} \phi_{j-1}^n &= \phi(x_{j-1}, t^n) = \phi(x_j - \Delta x, t^n) = \phi_j^n - \frac{\partial \phi}{\partial x} \Delta x + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Delta x^2 - \frac{1}{6} \frac{\partial^3 \phi}{\partial x^3} \Delta x^3 \\ &\quad + \frac{1}{24} \frac{\partial^4 \phi}{\partial x^4} \Delta x^4 + \mathcal{O}(\Delta x^4) \end{aligned} \quad (6.84)$$

将方程(6.83)和(6.84)代入到方程(6.75)中的左右两边有：

$$\begin{aligned}\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} &= \frac{\frac{\partial \phi}{\partial t} \Delta t + \frac{1}{2} \frac{\partial^2 \phi}{\partial t^2} \Delta t^2 + \mathcal{O}(\Delta t^2)}{\Delta t} = \frac{\partial \phi}{\partial t} + \frac{1}{2} \frac{\partial^2 \phi}{\partial t^2} \Delta t \\ \frac{\phi_{j+1}^n - 2\phi_j^n + \phi_{j-1}^n}{(\Delta x)^2} &= \frac{\frac{\partial^2 \phi}{\partial x^2} \Delta x^2 + \frac{1}{12} \frac{\partial^4 \phi}{\partial x^4} \Delta x^4 + \mathcal{O}(\Delta x^4)}{(\Delta x)^2} = \frac{\partial^2 \phi}{\partial x^2} + \frac{1}{12} \frac{\partial^4 \phi}{\partial x^4} \Delta x^2\end{aligned}\quad (6.85)$$

即：

$$\frac{\partial \phi}{\partial t} - \frac{\partial^2 \phi}{\partial x^2} = -\frac{1}{2} \frac{\partial^2 \phi}{\partial t^2} \Delta t + \frac{1}{12} \frac{\partial^4 \phi}{\partial x^4} \Delta x^2 \quad (6.86)$$

方程(6.86)等号右边的即为截断误差 T 。可以看出，截断误差 T 在 $\Delta t, \Delta x$ 趋向于 0 的情况下同时趋向于 0。同时，可以看出方程(6.75)是时间一阶、空间二阶的格式。同时，可以注意到截断误差第一项为一个正的扩散项，可能对求解产生不稳定性。

6.5 双曲系统与高阶格式

双曲系统 (hyperbolic system) 的偏微分方程求解为应用数学的一大领域。大量的数学问题都可以通过双曲偏微分方程进行求解。从历史的角度来看，CFD 大量的对流格式都起源于对双曲系统的研究，尤其是对可压缩欧拉方程的研究，衍生了大量的离散格式以及求解方法。一些简单的问题如 Burgers 方程、潜水方程都对离散格式的提出起到了重要的推动作用。但是自上世纪五十年代起，研究学者的最终极目标就是提出一个可以应用与欧拉方程的离散格式。这也与 CFD 的起源有关。二战期间，Los Alamos 实验室为了研究原子弹需要对激波进行捕获，CFD 则作为一种数学方法来对其进行预测。二战期间大量的研究工作都是保密的。直至九十年代才被公开。欧拉方程由于其非线性的特点，会在求解过程中发生间断。随之而然的，会导致产生激波。如何对激波进行精确的捕获则是各种离散格式要解决的问题。大量的前人的工作围绕如何提出一个精准的离散格式而进行。

双曲系统的离散格式分为很多不同种形式。由于双曲系统的偏微分方程可以分解为特征变量形式，进一步转化为若干黎曼问题，因此有一大类离散格式要依据波的传输方向来进行求解，也即迎风类格式。另一方面，一些算法在提出的过程中，将黎曼扇区域进行包含，不对积分区域内的黎曼问题进行显性的求解，这种格式通常被称之为中心格式。还有一些方法将非线性的偏微分方程进行线性化，从而可以求出系数矩阵的特征值以及特征向量并进行求解。这也就是 Roe 方法的思想 [74]。Godunov 方法 [40] 则在每个网格面上对黎曼问题进行时间步的推进并进行精准求解，即首先对每个网格面上对变量进行重组获得型线 (reconstruct)，然后将变量进行 Δt 时间步推进在每个网格面上求解黎曼问题 (evolve)，最后将变量在网格上进行平均获得体场 (average)，整个流程也被称之为 REA 过程。类似的还有 HLL 格式、Osher 格式、AUSM 格式、FVS 方法等，每个方法都源自于对双曲系统的特征传输特性的不同的理解。

仅仅讨论不可压缩流动的离散格式，忽略双曲系统的各种奇妙的求解方法是不完美的。OpenFOAM 中也对中心类格式进行了植入。因此本节主要讨论有限体积法下的双曲系统的各种数值问题，同时顺其自然的引出各种离散格式以及求解方法。

6.5.1 黎曼问题

黎曼问题通常指的是求解一个具有特殊初值的双曲方程。这个初值存在一个不连续点。图6.8所示为求解具有单一特征值的双曲方程 $\phi_t + u\phi_x = 0$ ，其中 u 为常数。假定 $\phi^t(x)$ 为初始分布，那其解为 $\phi^{t+\Delta t}(x) = \phi^t(x - u\Delta t)$ 。

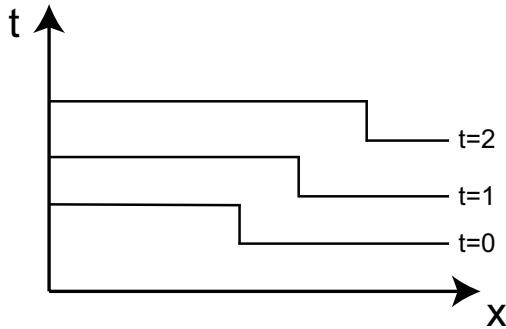


图 6.8: 经典黎曼问题不同时间步下的场分布。

6.5.2 双曲系统、特征变量

在上一节我们考虑的是单一常系数传输方程 $\phi_t + u\phi_x = 0$ 。若考虑若干耦合在一起的方程组，以下述方程为例：

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} &= 0 \\ \frac{\partial \rho u}{\partial t} + \frac{\partial \rho u^2 + P(\rho)}{\partial x} &= 0. \end{aligned} \quad (6.87)$$

注意，这里求解的为作为一个整体的守恒变量 $\rho, \rho u$ ，而非原始变量 ρ, u 。之所以其为耦合在一起的，是因为变量 $\rho, \rho u$ 在每个方程中都存在，二者互相影响。且其为一套非线性方程组。在这里我们来推导特征变量的控制方程。

首先用 \mathcal{U} 矢量表示

$$\mathcal{U} = \begin{cases} \rho \\ \rho u \end{cases} \quad (6.88)$$

也即

$$\frac{\partial \mathcal{U}}{\partial t} = \begin{cases} \frac{\partial \rho}{\partial t} \\ \frac{\partial \rho u}{\partial t} \end{cases} = \begin{cases} \frac{\partial \mathcal{U}_1}{\partial t} \\ \frac{\partial \mathcal{U}_2}{\partial t} \end{cases} \quad (6.89)$$

$$\frac{\partial f(\mathcal{U})}{\partial x} = \begin{cases} \frac{\partial \rho u}{\partial x} \\ \frac{\partial \rho u^2 + P(\rho)}{\partial x} \end{cases} = \begin{cases} \frac{\partial \mathcal{U}_2}{\partial x} \\ \frac{\partial \mathcal{U}_2^2 / \mathcal{U}_1 + P(\mathcal{U}_1)}{\partial x} \end{cases} \quad (6.90)$$

因此，方程(6.87)可以写为下述系统：

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial f(\mathcal{U})}{\partial x} = 0 \quad (6.91)$$

其中：

$$f(\mathcal{U}) = \begin{cases} \mathcal{U}_2 \\ \mathcal{U}_2^2 / \mathcal{U}_1 + P(\mathcal{U}_1) \end{cases} \quad (6.92)$$

方程(6.91)在双曲研究领域通常被称之为守恒法则 (conservation law)。需要注意的是，这种方程形式中的 \mathcal{U} 被假定为是可微分的，也就是说变量被假定为是光滑的。既然变量假定为光滑的，则进一步调用链条法则：

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial f(\mathcal{U})}{\partial \mathcal{U}} \frac{\partial \mathcal{U}}{\partial x} = 0 \quad (6.93)$$

其中 $\frac{\partial f(\mathcal{U})}{\partial \mathcal{U}}$ 即为 Jacobian 矩阵 \mathcal{A} (参考6.14节)。其可以写为：

$$\mathcal{A} = \frac{\partial f(\mathcal{U})}{\partial \mathcal{U}} = \begin{cases} 0, & 1 \\ -\mathcal{U}_2^2 / \mathcal{U}_1^2 + P'(\mathcal{U}_1), & 2\mathcal{U}_2 / \mathcal{U}_1 \end{cases} \quad (6.94)$$

针对 \mathcal{A} ，在这里要注意一个问题：

- 若 \mathcal{A} 为常量，则构成线性双曲系统。
- 若 \mathcal{A} 为依赖 \mathcal{U} 的变量，则构成准线性 (quasi-linear) 双曲系统。
- 若 \mathcal{A} 为依赖 x 的变量，则构成可变系数 (variable-coeffieicnt) 双曲系统。

现考虑线性双曲系统，进一步的，若 \mathcal{A} 继续可以被对角化，则有：

$$\mathcal{A} = \mathbf{R} \Lambda \mathbf{R}^{-1}, \mathbf{R}^{-1} \mathcal{A} = \Lambda \mathbf{R}^{-1} \quad (6.95)$$

其中 Λ 为一个对角阵，且对角线系数 λ_1, λ_2 各不相同。 \mathbf{R} 表示特征向量 \mathbf{r}^i 组建的矩阵：

$$R = [\mathbf{r}^1 | \mathbf{r}^2] \quad (6.96)$$

将方程(6.93)左右两边同时乘以 \mathbf{R}^{-1} ，有：

$$\mathbf{R}^{-1} \frac{\partial \mathcal{U}}{\partial t} + \Lambda \mathbf{R}^{-1} \frac{\partial \mathcal{U}}{\partial x} = 0 \quad (6.97)$$

令：

$$\mathcal{W} = \mathbf{R}^{-1} \mathcal{U} \quad (6.98)$$

有：

$$\frac{\partial \mathcal{W}}{\partial t} + \Lambda \frac{\partial \mathcal{W}}{\partial x} = 0 \quad (6.99)$$

也即：

$$\begin{aligned} \frac{\partial \mathcal{W}_1}{\partial t} + \frac{\partial \lambda_1 \mathcal{W}_1}{\partial x} &= 0 \\ \frac{\partial \mathcal{W}_2}{\partial t} + \frac{\partial \lambda_2 \mathcal{W}_2}{\partial x} &= 0 \end{aligned} \quad (6.100)$$

其中 \mathcal{W} 被称之为特征变量。同时可以注意到，方程(6.100)为解耦的。求解时，其可以被看做为两个单独的黎曼问题，具有不同的传输速度 λ_1, λ_2 。

同时可以看出，若可以获得 \mathcal{W} 的解， \mathcal{U} 也可以求出：

$$\mathcal{U} = \mathbf{R}\mathcal{W} = [\mathbf{r}^1 | \mathbf{r}^2] \begin{bmatrix} \mathcal{W}_1 \\ \mathcal{W}_2 \end{bmatrix} = \mathcal{W}_1 \mathbf{r}^1 + \mathcal{W}_2 \mathbf{r}^2 = \sum \mathcal{W}_i \mathbf{r}^i \quad (6.101)$$

注意其中 \mathcal{W}_i 为 \mathcal{W} 的元素， \mathbf{r}^i 表示特征向量。可以看出， \mathcal{U} 即为特征向量的叠加。 \mathcal{W}_i 表示波的强度。定义黎曼问题间断处每个波的强度差为 α_i ，有：

$$\alpha_i = \mathcal{W}_i^r - \mathcal{W}_i^l \quad (6.102)$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \mathcal{W}_1^r - \mathcal{W}_1^l \\ \mathcal{W}_2^r - \mathcal{W}_2^l \\ \mathcal{W}_3^r - \mathcal{W}_3^l \end{bmatrix} \quad (6.103)$$

考虑图6.9的黎曼问题，在 t_0 时刻，左侧波的强度分别为 $\mathcal{W}_l^i, i = 1, 2, 3$ ，右侧波的强度分别为 $\mathcal{W}_r^i, i = 1, 2, 3$ 。左右侧波强度的差值则为：

$$\sum \alpha_i \mathbf{r}^i = \sum (\mathcal{W}_i^r - \mathcal{W}_i^l) \mathbf{r}^i = [\mathbf{r}^1 | \mathbf{r}^2] \begin{bmatrix} \mathcal{W}_1^r - \mathcal{W}_1^l \\ \mathcal{W}_2^r - \mathcal{W}_2^l \\ \mathcal{W}_3^r - \mathcal{W}_3^l \end{bmatrix} = \mathbf{R}\boldsymbol{\alpha} \quad (6.104)$$

$$\mathbf{R}\mathcal{W}_r = \mathbf{R}\mathcal{W}_l + \mathbf{R}\boldsymbol{\alpha} \quad (6.105)$$

现在看图6.9的解。在 t_0 时刻，间断左侧的初始值分别为 $\mathcal{W}_1^l, \mathcal{W}_2^l, \mathcal{W}_3^l$ ，间断右侧的初始值分别为 $\mathcal{W}_1^r, \mathcal{W}_2^r, \mathcal{W}_3^r$ 。随着时间步的推进，针对区域 1，所有特征变量均没有跨域 λ_1 特征线，因此区域 1 的值保持不变。针对区域 2， \mathcal{W}_1 特征变量的值为 \mathcal{W}_1^r ，其他的特征变量保持不变为 $\mathcal{W}_2^l, \mathcal{W}_3^l$ 。其他区域同理。因此，若认为黎曼问题左侧的初始值为：

$$\mathcal{U}_l^{t_0} = \mathbf{R}\mathcal{W}_l \quad (6.106)$$

则区域 1 的解也为 $\mathcal{U}_l^{t_0}$ 。现考虑区域 2，区域 2 相对于区域 1 有一个跃阶 $(\mathcal{W}_1^r - \mathcal{W}_1^l)\mathbf{r}^1$ ，因此区域 2 的解可以写为：

$$\mathbf{R}\mathcal{W}_l + (\mathcal{W}_1^r - \mathcal{W}_1^l)\mathbf{r}^1 = \mathbf{R}\mathcal{W}_l + \alpha_1 \mathbf{r}^1 \quad (6.107)$$

同理，区域 3 的解可以写为：

$$\mathbf{R}\mathcal{W}_l + (\mathcal{W}_1^r - \mathcal{W}_1^l)\mathbf{r}^1 + (\mathcal{W}_2^r - \mathcal{W}_2^l)\mathbf{r}^2 = \mathbf{R}\mathcal{W}_l + \alpha_1\mathbf{r}^1 + \alpha_2\mathbf{r}^2 \quad (6.108)$$

区域 4 的解可以写为：

$$\mathbf{R}\mathcal{W}_l + (\mathcal{W}_1^r - \mathcal{W}_1^l)\mathbf{r}^1 + (\mathcal{W}_2^r - \mathcal{W}_2^l)\mathbf{r}^2 + (\mathcal{W}_3^r - \mathcal{W}_3^l)\mathbf{r}^3 = \mathbf{R}\mathcal{W}_l + \alpha_1\mathbf{r}^1 + \alpha_2\mathbf{r}^2 + \alpha_3\mathbf{r}^3 \quad (6.109)$$

结合方程(6.105)，可见区域 4 的解也为 $\mathbf{R}\mathcal{W}_r$ 。

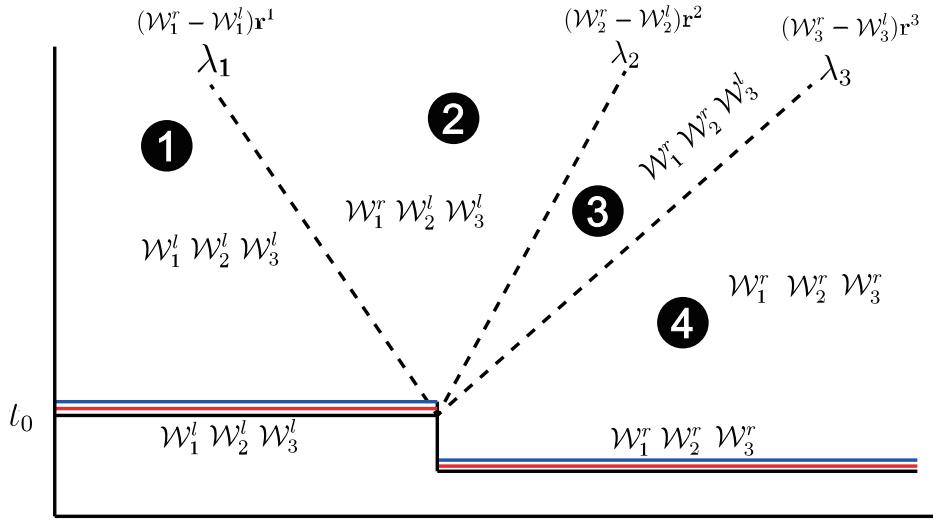


图 6.9: 三特征值黎曼问题解的区域。三个特征线分别对应三个 Rankine-Hugoniot 间断。

6.5.3 积分形式与守恒方程

前述章节推导了各种积分形式的控制方程。在双曲系统研究领域，由于间断存在的可能性，主要是用积分形式的控制方程，如方程(3.36)。现对其进行回顾。

将方程(3.36)移项，并整理为方程(6.91)的形式：

$$\int_V \mathcal{U}(t + \Delta t) dV = \int_V \mathcal{U}(t) dV - \int_t^{t+\Delta t} f(\mathcal{U}(t')) dt' \quad (6.110)$$

为方便讨论问题，一维形式可以写为：

$$\int_{\Delta x} \mathcal{U}(t + \Delta t) dx = \int_{\Delta x} \mathcal{U}(t) dx - \left(\int_t^{t+\Delta t} f(\mathcal{U}(t')_{i+\frac{1}{2}}) dt' - \int_t^{t+\Delta t} f(\mathcal{U}(t')_{i-\frac{1}{2}}) dt' \right) \quad (6.111)$$

两边同时除以 Δx 有：

$$\begin{aligned} \frac{1}{\Delta x} \int_{\Delta x} \mathcal{U}(t + \Delta t) dx &= \frac{1}{\Delta x} \int_{\Delta x} \mathcal{U}(t) dx \\ &\quad - \frac{1}{\Delta x} \left(\int_t^{t+\Delta t} f(\mathcal{U}(t')_{i+\frac{1}{2}}) dt' - \int_t^{t+\Delta t} f(\mathcal{U}(t')_{i-\frac{1}{2}}) dt' \right) \end{aligned} \quad (6.112)$$

其中 $\frac{1}{\Delta x} \int_{\Delta x} \mathcal{U}(t + \Delta t) dx$ 表示 \mathcal{U} 在 $t + \Delta t$ 时间步下，网格区间内的体平均值，用 $\bar{\mathcal{U}}_i^{t+\Delta t}$ 来表示。同理，当前时间步的体平均值用 $\bar{\mathcal{U}}_i^t$ 来表示。方程可以继续写为：

$$\bar{\mathcal{U}}_i^{t+\Delta t} = \bar{\mathcal{U}}_i^t - \frac{\Delta t}{\Delta x} \left(\int_t^{t+\Delta t} f(\mathcal{U}(t')_{i+\frac{1}{2}}) dt' - \int_t^{t+\Delta t} f(\mathcal{U}(t')_{i-\frac{1}{2}}) dt' \right) \quad (6.113)$$

方程(6.113)的意义很明确。控制体网格下变量的下一个时间步的值，等于当前时间步的值，以及网格通量的贡献之和。 $\int_t^{t+\Delta t} f(\mathcal{U}(t')_{i+\frac{1}{2}}) dt'$ 表示定义在网格面 $i + \frac{1}{2}$ ，时间步长 Δt 内通量的影响。其可以继续简写为：

$$\bar{\mathcal{U}}_i^{t+\Delta t} = \bar{\mathcal{U}}_i^t - \frac{\Delta t}{\Delta x} (\mathcal{F}_{i+\frac{1}{2}} - \mathcal{F}_{i-\frac{1}{2}}) \quad (6.114)$$

同时需要注意的是，方程(6.114)是精准的。但其尚未封闭。因此，各种格式的意义，即对通量进行封闭。

6.5.4 封闭与基架点

封闭的根本是通过当前时间步的值，来对通量 \mathcal{F} 进行模化。考虑最简单的情况，我们认为网格 $i + \frac{1}{2}$ 处的通量，仅仅跟相邻的网格点的变量有关。即：

$$\mathcal{F}_{i+\frac{1}{2}} = \mathcal{F}(\bar{\mathcal{U}}_i^t, \bar{\mathcal{U}}_{i+1}^t) \quad (6.115)$$

$$\bar{\mathcal{U}}_i^{t+\Delta t} = \bar{\mathcal{U}}_i^t - \frac{\Delta t}{\Delta x} (\mathcal{F}(\bar{\mathcal{U}}_i^t, \bar{\mathcal{U}}_{i+1}^t) - \mathcal{F}(\bar{\mathcal{U}}_{i-1}^t, \bar{\mathcal{U}}_i^t)) \quad (6.116)$$

具体 $\mathcal{F}(\bar{\mathcal{U}}_i^t, \bar{\mathcal{U}}_{i+1}^t)$ 采用什么样的形式，与离散格式有关。不管什么样的格式，方程(6.116)仅仅与 $i, i-1, i+1$ 网格点的变量有关。因此，本方法采用的是三点基架点 (three-point stencil)。

6.5.5 中心格式

方程(6.115)最简单的形式，即网格面的通量为相邻网格单元的值的一种算术平均：

$$\mathcal{F}_{i+\frac{1}{2}} = \mathcal{F}(\bar{\mathcal{U}}_i^t, \bar{\mathcal{U}}_{i+1}^t) = \frac{1}{2} (f(\mathcal{U}_i^t) + f(\mathcal{U}_{i+1}^t)) \quad (6.117)$$

将方程(6.117)代入到(6.116)有：

$$\bar{\mathcal{U}}_i^{t+\Delta t} = \bar{\mathcal{U}}_i^t - \frac{\Delta t}{2\Delta x} (f(\mathcal{U}_{i+1}^t) - f(\mathcal{U}_{i-1}^t)) \quad (6.118)$$

需要注意的是，中心格式是无条件不稳定的。

6.5.6 Lax-Friedrichs 格式

参考中心格式，Lax-Friedrichs 格式的通量定义为：

$$\mathcal{F}_{i+\frac{1}{2}} = \mathcal{F}(\bar{\mathcal{U}}_i^t, \bar{\mathcal{U}}_{i+1}^t) = \frac{1}{2} (f(\mathcal{U}_i^t) + f(\mathcal{U}_{i+1}^t)) - \frac{\Delta x}{2\Delta t} (\mathcal{U}_{i+1}^t - \mathcal{U}_i^t) \quad (6.119)$$

从方程形式来看, Lax-Friedrichs 格式像是在中心格式上添加了一定的贡献来获得稳定性。在这里回忆一下扩散项, 扩散项的形式为 $-\frac{\partial}{\partial x}(\Gamma \frac{\partial u}{\partial x})$, 对其在 $i + \frac{1}{2}$ 处离散有

$$-\Gamma_{i+\frac{1}{2}} \frac{U_{i+1}^t - U_i^t}{\Delta x^2} \quad (6.120)$$

对比方程(6.119), 可以发现 Lax-Friedrichs 格式即为在中心格式上添加了一个系数 $\Gamma = 0.5\Delta x^2/\Delta t$ 的数值扩散项。正是这个数值扩散项的存在, 使得 Lax-Friedrichs 格式可以稳定。

需要注意的是, 当前讨论的 Lax-Friedrichs 格式有时也被称之为全域 Lax-Friedrichs 格式。这是因为附加的扩散项的系数 $\Delta x/\Delta t$ 固定。这个扩散系数还可以通过一个局部的值来替换, 也即局部 Lax-Friedrichs 格式。

6.5.7 Kurganov-Tadmor 格式

参考 Lax-Friedrichs 格式, Kurganov-Tadmor 格式的通量定义为:

$$\mathcal{F}_{i+\frac{1}{2}} = \mathcal{F}(\bar{U}_i^t, \bar{U}_{i+1}^t) = \frac{1}{2}(f(U_i^t) + f(U_{i+1}^t)) - \frac{1}{2}a_{i+\frac{1}{2}}(U_{i+1}^t - U_i^t) \quad (6.121)$$

其中 $a_{i+\frac{1}{2}}$ 表示黎曼扇区域的最大波速。Kurganov-Tadmor 格式认为 Lax-Friedrichs 格式的数值耗散系数过大, 因此在 $a_{i+\frac{1}{2}} < \Delta x/\Delta t$ 的情况下, Kurganov-Tadmor 格式的数值耗散更小。理论上来讲, $a_{i+\frac{1}{2}} < \Delta x/\Delta t$ 表示当前时间步 Δt 下, 黎曼扇区域在两个网格中心点之间。

可以看出 Kurganov-Tadmor 格式与局部 Lax-Friedrichs 格式本质上是一致的。在一些教材中, 二者称呼混用 [41, 53]。在一些教材中也被称之为 Rusanov 格式 [41, 88]。

6.5.8 Godunov 格式

Godunov 格式的通量计算是精准的, 其采用解析解来进行, 并不包含任何假设, 因此 Godunov 格式的通量仅仅用下式来进行表达:

$$\mathcal{F}_{i+\frac{1}{2}} = \mathcal{F}_{i+\frac{1}{2}}^g \quad (6.122)$$

但这并不证明 Godunov 求解的结果是精准的, 因为其还涉及到网格点到网格面的变量重构算法 (图6.10所示的即为一阶重构)。且通量为时间积分形式。从历史上来讲, 在 Godunov 格式开发之前, 主要是用有限差分来求解线性问题。但有限差分由于处理间断能力不足, 将其拓展并处理非光滑问题难上加难。因此 Godunov 格式可以认为是有限体积法的起源。Godunov 在开发这种有限体积类算法的过程中, 引入了三种主要流程 (思想), 这虽然在现在看来是顺其自然的。但是在上世纪 50 年代, 只有 Godunov 格式开发出来之后, 后人才相继开发出更多的高阶格式。

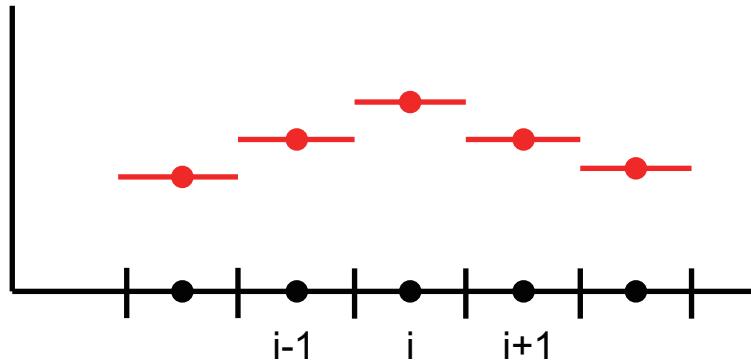


图 6.10: 图中红点为有限差分假定的变量值。图中红线为有限体积法假定的变量分布。

如图6.10所示，有限差分法在每个网格上认为存在一点，并且这个点代表了网格的值。Godunov思想的第一步是认为在整个网格内部变量存在一定的分布，图6.10所示的为分段常数分布（piecewise constant distribution），其为一种一阶假定。这一步也被称之为重构（reconstruct）步。如图6.11所示，Godunov格式的第二步在每个网格交界面求解黎曼问题的精确解，也即演变（evolve）步骤。在求解每个网格交界面的黎曼问题之后，如图6.12所示，Godunov格式的第三步需要将变量分布进行重组（reconstruct）。然后进行下一个时间步的计算。

在实施的过程中，在每个网格交界面求解黎曼问题是不可能的。因此，衍生出一些近似黎曼求解器，如上文提到的 Lax-Friedrichs 格式。近似黎曼格式并不需要在每个网格面求解黎曼问题，例如 Kurganov-Tadmor 格式将黎曼扇进行包含，黎曼扇内部的变量演变并不关心。

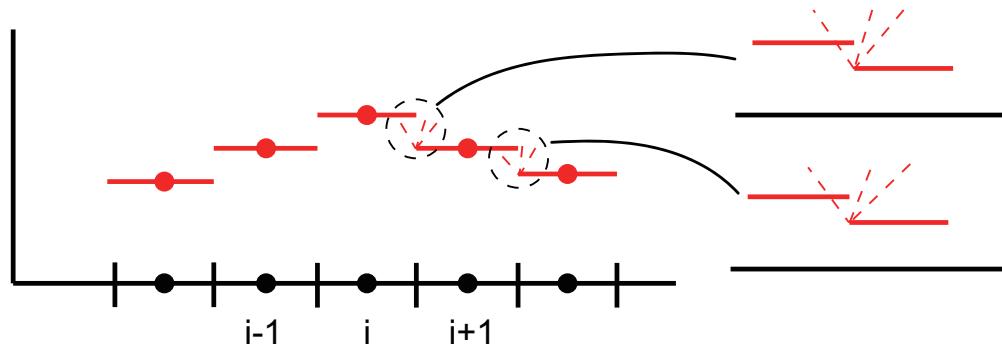


图 6.11: Godunov 第二步在每个网格交界面求解黎曼问题的精确解。

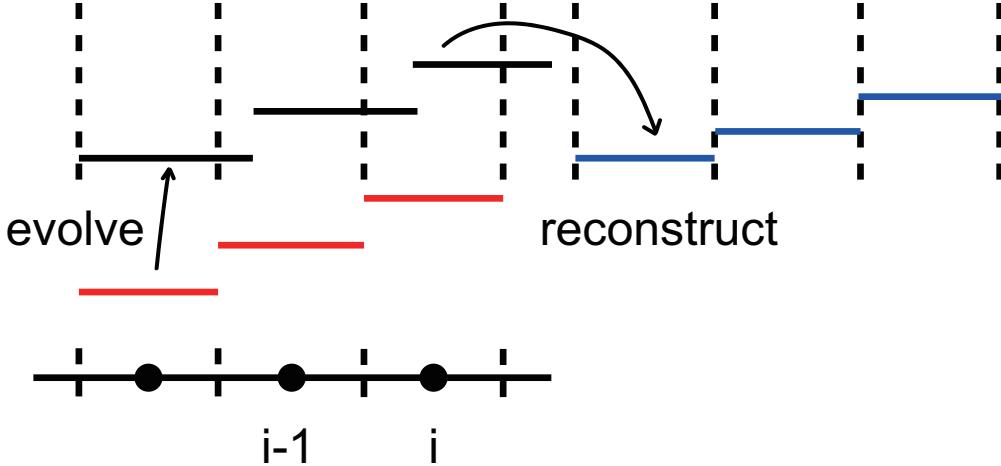


图 6.12: Godunov 第三步在每个网格将变量进行重组。

6.6 为什么 $-\frac{2}{3}\mu(\nabla \cdot \mathbf{U})\mathbf{I}$ 为体膨胀系数项?

在方程(3.81)中, 可压缩流体的 τ 中需要考虑 $-\frac{2}{3}\mu(\nabla \cdot \mathbf{U})\mathbf{I}$ 。其中的 $-\frac{2}{3}\mu$ 也经常被称为体膨胀/压缩粘度。这是因为 $-\frac{2}{3}\mu(\nabla \cdot \mathbf{U})\mathbf{I}$ 可以转换为密度相关变量:

$$-\frac{2}{3}\mu(\nabla \cdot \mathbf{U})\mathbf{I} = -\frac{2}{3}\mu \left(\frac{1}{\rho} \frac{D\rho}{Dt} \right) \mathbf{I} \quad (6.123)$$

可见, $\frac{1}{\rho} \frac{D\rho}{Dt}$ 为密度的物质导数项, 从第3.3.3节可知, 其表示拉格朗日框架下密度小微团的变化率。因此, 即跟体膨胀相关。

6.7 什么是应力的 Deviatoric 与 Hydrostatic 部分?

在 CFD 中, 通常要将二阶张量 σ 的 deviatoric 部分和 hydrostatic 部分做区分, 如:

$$\sigma = \sigma_{dev} + \sigma_{hyd} \quad (6.124)$$

用下面这个例子可以表明 σ_{dev} 和 σ_{hyd} 的计算关系。如果

$$\sigma = \begin{pmatrix} 50 & 30 & 20 \\ 30 & -20 & -10 \\ 20 & -10 & 10 \end{pmatrix} \quad (6.125)$$

则正应力 hydrostatic 部分可以这样计算:

$$\sigma_{hyd} = \frac{50 - 20 + 10}{3} = 13.3 \quad (6.126)$$

$$\boldsymbol{\sigma}_{hyd} = \begin{pmatrix} 13.3 & 0 & 0 \\ 0 & 13.3 & 0 \\ 0 & 0 & 13.3 \end{pmatrix} \quad (6.127)$$

偏应力 deviatoric 部分可以这样计算：

$$\boldsymbol{\sigma}_{dev} = \boldsymbol{\sigma} - \boldsymbol{\sigma}_{hyd} = \begin{pmatrix} 36.7 & 30 & 20 \\ 30 & -33.3 & -10 \\ 20 & -10 & -3.3 \end{pmatrix} \quad (6.128)$$

可以看出， $\boldsymbol{\sigma}_{dev}$ 的迹是 0，也就是说，对于一个二阶张量的偏应力部分，其正应力部分为 0。

二阶张量 $\boldsymbol{\sigma}$ 的正应力与偏应力部分也可以这样推导：

$$\boldsymbol{\sigma}_{hyd} = \frac{1}{3}\text{tr}(\boldsymbol{\sigma})\mathbf{I} \quad (6.129)$$

$$\boldsymbol{\sigma}_{dev} = \boldsymbol{\sigma} - \boldsymbol{\sigma}_{hyd} = \boldsymbol{\sigma} - \frac{1}{3}\text{tr}(\boldsymbol{\sigma})\mathbf{I} \quad (6.130)$$

为什么要将二阶张量分为正应力部分与偏应力部分，这是因为压力 p 可以看做一个正应力部分，剪切应力 $\boldsymbol{\tau}$ 可以看做是偏应力部分，二者可以构成一个整体的二阶张量 $\boldsymbol{\sigma}$ ：

$$\boldsymbol{\sigma} = \boldsymbol{\sigma} - \frac{1}{3}\text{tr}(\boldsymbol{\sigma})\mathbf{I} + \frac{1}{3}\text{tr}(\boldsymbol{\sigma})\mathbf{I} \quad (6.131)$$

令

$$-p\mathbf{I} = \frac{1}{3}\text{tr}(\boldsymbol{\sigma})\mathbf{I} \quad (6.132)$$

有：

$$\boldsymbol{\sigma} = -p\mathbf{I} + \left(\boldsymbol{\sigma} - \frac{1}{3}\text{tr}(\boldsymbol{\sigma})\mathbf{I} \right) = \boldsymbol{\sigma}_{hyd} + \boldsymbol{\sigma}_{dev} \quad (6.133)$$

在这种情况下，动量方程右侧可以理解为：

$$-\nabla p + \nabla \cdot \boldsymbol{\tau} = \nabla \cdot \boldsymbol{\sigma} \quad (6.134)$$

一些教材偏向使用 $\nabla \cdot \boldsymbol{\sigma}$ 来进行讲解，其中 $\boldsymbol{\sigma}$ 被称之为柯西应力张量。本笔记则采用 $-\nabla p + \nabla \cdot \boldsymbol{\tau}$ ，因为其更有易于理解。

6.8 二阶张量不变量：Invariants

给定一个二阶张量，可以定义其各种不变量。给定一个矩阵 \mathbf{A} ，其最基本的三个不变量分别是 I_1, I_2, I_3 ：

$$\begin{aligned} I_1 &= \text{tr}(\mathbf{A}) = \lambda_1 + \lambda_2 + \lambda_3 \\ I_2 &= \frac{1}{2} ((\text{tr}(\mathbf{A}))^2 - \text{tr}(\mathbf{A}^2)) = \lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3 \\ I_3 &= \det(\mathbf{A}) = \lambda_1 \lambda_2 \lambda_3 \end{aligned} \quad (6.135)$$

其中 λ 为 \mathbf{A} 的特征值。其中 I_1 被称之为第一不变量, I_2 被称之为第二不变量, I_3 被称之为第三不变量。不变量也可以通过 \mathbf{A} 的元素来进行表示:

$$\begin{aligned} I_1 &= A_{11} + A_{22} + A_{33} \\ I_2 &= A_{11}A_{22} + A_{22}A_{33} + A_{11}A_{33} - A_{12}A_{21} - A_{23}A_{32} - A_{13}A_{31} \\ I_3 &= -A_{13}A_{22}A_{31} + A_{12}A_{23}A_{31} + A_{13}A_{21}A_{32} - A_{11}A_{23}A_{32} - A_{12}A_{21}A_{33} + A_{11}A_{22}A_{33} \end{aligned} \quad (6.136)$$

6.9 如何理解雷诺应力项中的 $\frac{2}{3}\rho k \mathbf{I}$?

将动量方程左侧的对流项进行雷诺平均后, 会出现雷诺应力项:

$$\overline{\nabla \cdot (\rho \mathbf{U} \mathbf{U})} = \nabla \cdot (\bar{\rho} \overline{\mathbf{U} \mathbf{U}}) + \nabla \cdot (\bar{\rho} \overline{\mathbf{U}' \mathbf{U}'}) \quad (6.137)$$

$\bar{\rho} \overline{\mathbf{U}' \mathbf{U}'}$ 这一项若放在动量方程右侧会出现一个负号, 即方程右侧为 $= \dots + (-\bar{\rho} \overline{\mathbf{U}' \mathbf{U}'})$ 。我们把 $-\bar{\rho} \overline{\mathbf{U}' \mathbf{U}'}$ 称之为雷诺应力 $\boldsymbol{\tau}_t$ ⁷, 其为一个二阶对称张量。在线性涡粘类湍流模型中, 雷诺应力项可通过 Bousinessq 近似来进行封闭。考虑湍流动能的定义:

$$\bar{\rho} k = \frac{1}{2} \bar{\rho} \overline{\mathbf{U}' \cdot \mathbf{U}'} = \frac{1}{2} \text{tr}(\bar{\rho} \overline{\mathbf{U}' \mathbf{U}'}) \quad (6.138)$$

雷诺应力项通常被分为正应力部分与偏应力部分:

$$-\bar{\rho} \overline{\mathbf{U}' \mathbf{U}'} = -\bar{\rho} \overline{\mathbf{U}' \mathbf{U}'} - \frac{1}{3} \text{tr}(-\bar{\rho} \overline{\mathbf{U}' \mathbf{U}'}) \mathbf{I} + \frac{1}{3} \text{tr}(-\bar{\rho} \overline{\mathbf{U}' \mathbf{U}'}) \mathbf{I} \quad (6.139)$$

将方程(6.138)代入到(6.139)有:

$$-\bar{\rho} \overline{\mathbf{U}' \mathbf{U}'} = -\bar{\rho} \overline{\mathbf{U}' \mathbf{U}'} + \frac{2}{3} \bar{\rho} k \mathbf{I} - \frac{2}{3} \bar{\rho} k \mathbf{I} \quad (6.140)$$

其中的 $-\frac{2}{3} \bar{\rho} k \mathbf{I}$ 通常与动量方程中的压力梯度项 $\nabla \bar{p}$ 组合到一起:

$$-\nabla \tilde{\bar{p}} = -\nabla \bar{p} - \nabla \cdot \left(\frac{2}{3} \bar{\rho} k \mathbf{I} \right) \quad (6.141)$$

另外一种说法是, 对于非超音速流动, $-\frac{2}{3} \bar{\rho} k \mathbf{I}$ 可以忽略⁸。随后, 方程(6.140)中的 $-(\bar{\rho} \overline{\mathbf{U}' \mathbf{U}'} - \frac{2}{3} \bar{\rho} k \mathbf{I})$ 可以继续通过 Bousinessq 近似附加湍流粘度进行模化, 将其转化为 $\overline{\mathbf{U}}$ 的方程进而封闭。

⁷注意, 雷诺应力项在方程右侧为负号: $\dots = \dots - \nabla \cdot (\rho \overline{\mathbf{U}' \mathbf{U}'})$

⁸参考 NASA 的介绍。

6.10 变量有界性

某些传输的变量具备有界的特征，比如体积分数 α 严格的介于 0 和 1 之间，密度则严格的大于 0。在离散的过程中，高阶格式会产生一些越界的解，这会带来收敛问题。考虑波传输方程

$$\frac{\partial \alpha}{\partial t} + \mathbf{U} \cdot \nabla \alpha = 0 \quad (6.142)$$

其中 \mathbf{U} 表示传输速度。其解 α 为严格有界的。因此在求解传输方程的时候，方程(6.142)优先选择。但方程(6.142)为非守恒形式并且不存在通量函数⁹。因此在对方程(6.142)离散求解时候，其通常改写为

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\mathbf{U} \alpha) - \alpha \nabla \cdot \mathbf{U} = 0 \quad (6.143)$$

由于存在对速度的散度，因此解决了通量函数不存在的问题。同时，虽然连续性方程 $\nabla \cdot \mathbf{U} = 0$ 意味着第三项可以忽略，即求解

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\mathbf{U} \alpha) = 0 \quad (6.144)$$

但在稳态算法的迭代步中， $\nabla \cdot \mathbf{U} \neq 0$ 。因此，只有求解方程(6.143)才能保证变量的有界。另外，瞬态算法可以保证每一个时间步都收敛 ($\nabla \cdot \mathbf{U} - 0$ 降低至压力残差以下)，因此可以直接求解方程(6.144)。但对于需要有界的变量来讲，只有求解方程(6.143)，同时调用有界的离散格式（如 vanLeer 格式）才能保证变量的有界。

上述理论可以通过开源 CFD 软件 OpenFOAM 进行测试和验证，在 OpenFOAM 中的稳态算例中，对流项的离散格式需要添加**bounded**关键词，例如湍流动能对流项指定为¹⁰

```
1 div(phi,k)      bounded Gauss upwind;
```

这意味着求解方程(6.143)的形式。一些瞬态算例，某些变量同样指定为¹¹

```
1 div(phi,s)      bounded Gauss limitedLinear 1;
```

这也意味着求解方程(6.143)的形式。某些变量（如速度）则指定为

```
1 div(phi,U)      Gauss LUST grad(U);
```

这意味着求解方程(6.144)的形式。其表示并没有对速度进行严格有界处理。对于湍流动能也通过求解方程(6.144)的形式来处理，其有界性通过**bound**函数来保证。

在实际测试中，相比方程(6.144)，方程(6.143)虽然有很大改善，但依然不能严格的保证变量的有界。在这种情况下，则需要一些更高级的处理方式来进行，如反扩散高阶格式。

⁹考虑在 i 网格点对 $\mathbf{U} \cdot \nabla \alpha$ 进行离散，在离散时， \mathbf{U} 在 $i + \frac{1}{2}, i - \frac{1}{2}$ 可以具有不同的值。同时 $i + \frac{1}{2}, i - \frac{1}{2}$ 上的面积应该也具有不同的值。比如一个进口面积较大，出口面积较小的计算域，必然进口速度小，出口速度大。但若对方程(6.142)直接离散不能体现面积的影响。

¹⁰[tutorials/incompressible/simpleFoam/motoBike](#)

¹¹[tutorials/incompressible/pisoFoam/LES/pitzDaily](#)

6.11 守恒与非守恒、守恒变量与原始变量

间断解对于求解的控制方程与算法来说是一种挑战。在3.5.3节描述了守恒与非守恒方程的转化关系。在3.5.5节讨论了积分形式的控制方程。但为什么他们被称之为守恒形式和非守恒形式？从物理与数学形式上看他们有什么意义？方程为什么要区分为积分形式与微分形式？原始变量与守恒变量在求解上哪一个更优？

从数学角度来讲，如果考虑 x 方向动量的导数，网格如图6.13所示，字母表示网格单元编号，数字表示面编号。对于 B 网格点，对于动量 ρu 这个守恒变量进行有限体积法离散为：

$$\int_1^2 \frac{\partial \rho u}{\partial x} dx = (\rho u)_2 - (\rho u)_1 \quad (6.145)$$

对 A, B, C 网格点进行离散并加和有：

$$\begin{aligned} \int_0^1 \frac{\partial \rho u}{\partial x} dx + \int_1^2 \frac{\partial \rho u}{\partial x} dx + \int_2^3 \frac{\partial \rho u}{\partial x} dx &= (\rho u)_1 - (\rho u)_0 + (\rho u)_2 - (\rho u)_1 + (\rho u)_3 - (\rho u)_2 \\ &= (\rho u)_3 - (\rho u)_0 = \int_0^3 \frac{\partial \rho u}{\partial x} dx \end{aligned} \quad (6.146)$$

也即不管如何加和，最后只有边界通量的影响，内部点互相约去，即通量永远是守恒的，进来多少，出去多少。相反，对于动量 ρu 这个守恒变量，将其拆分为原始变量 ρ, u 同时进

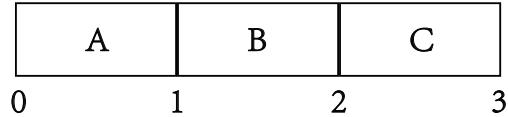


图 6.13: 一维有限体积网格。

行有限体积法离散为：

$$\int_1^2 \frac{\partial \rho u}{\partial x} dx = \int_1^2 \left(u \frac{\partial \rho}{\partial x} + \rho \frac{\partial u}{\partial x} \right) dx = \tilde{u}(\rho_2 - \rho_1) + \tilde{\rho}(u_2 - u_1) \quad (6.147)$$

其中的面上的 $\tilde{u}, \tilde{\rho}$ 需要计算而来，不同的格式对应不同的结果。如果定义为网格的平均值，同时对 A, B, C 网格点进行离散并加和有：

$$\begin{aligned} \int_0^1 \left(u \frac{\partial \rho}{\partial x} + \rho \frac{\partial u}{\partial x} \right) dx + \int_1^2 \left(u \frac{\partial \rho}{\partial x} + \rho \frac{\partial u}{\partial x} \right) dx + \int_2^3 \left(u \frac{\partial \rho}{\partial x} + \rho \frac{\partial u}{\partial x} \right) dx = \\ u_A(\rho_1 - \rho_0) + \rho_A(u_1 - u_0) + u_B(\rho_2 - \rho_1) + \rho_B(u_2 - u_1) + u_C(\rho_3 - \rho_2) + \rho_C(u_3 - u_2) \end{aligned} \quad (6.148)$$

很明显，方程(6.148)的右边并不能够互相约去，也即为非守恒。

从物理意义来讲，在守恒法则下，守恒变量方程表示的是“真正”守恒的方程，原始变量方程表示的是“虚假”守恒的方程。例如考虑一维欧拉方程，守恒变量方程可以表示

为：

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} &= 0 \\ \frac{\partial \rho u}{\partial t} + \frac{\partial \rho uu}{\partial x} &= 0\end{aligned}\tag{6.149}$$

将动量方程展开：

$$\begin{aligned}\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + \rho \frac{\partial u}{\partial x} &= 0 \\ \frac{\partial \rho u}{\partial t} + \frac{\partial \rho uu}{\partial x} &= u \frac{\partial \rho}{\partial t} + \rho \frac{\partial u}{\partial t} + u \frac{\partial \rho u}{\partial x} + \rho u \frac{\partial u}{\partial x} = 0\end{aligned}\tag{6.150}$$

结合连续型方程，其可以写为原始变量方程：

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} &= 0 \\ \frac{\partial u}{\partial t} + \frac{\partial^{\frac{1}{2}} uu}{\partial x} &= 0\end{aligned}\tag{6.151}$$

对比方程(6.149)与(6.151)，可以看出二者都表示守恒法则，不过前者表示质量与动量的守恒，后者表示质量与“速度”的守恒。由于速度并不是守恒的变量，因此方程(6.151)表示的是一种“虚假”守恒的状态。但数学过程是严谨的，那么方程(6.151)是否可以求解呢？答案是可以的，只要解是光滑的。若存在间断，“虚假”守恒的方程会预测不同的结果。

6.12 分离式求解器与耦合式求解器

在求解 CFD 变量的情况下，可以使用分离式求解器以及耦合式求解器。这是因为 CFD 方程组通常存在若干的变量，比如 p , \mathbf{U} 。在使用分离式求解器的时候，针对每个变量进行次序求解。例如对于 p , \mathbf{U} ，分离式求解器将求解 p 方程, u 方程, v 方程, w 方程。如果要求解 p 方程，那其中的除了 p 变量之外的所有变量都只能通过已知量来替换。因此分离式求解器可以看做是一种解耦。耦合式求解器则同时求解相关变量。例如在形成矩阵系统时， p 方程, u 方程, v 方程, w 方程将组成一个大的矩阵。在进行求解的时候， p 、 u 、 v 、 w 一次性被求出。例如，下面的方程即为 4 网格点分离式求解器组建的压力方程以及 x 方向速度方程，其为两套矩阵系统：

$$\begin{bmatrix} * & * & 0 & 0 \\ * & * & * & 0 \\ 0 & * & * & * \\ 0 & 0 & * & * \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = \begin{bmatrix} * \\ * \\ * \\ * \end{bmatrix}, \quad \begin{bmatrix} * & * & 0 & 0 \\ * & * & * & 0 \\ 0 & * & * & * \\ 0 & 0 & * & * \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} * \\ * \\ * \\ * \end{bmatrix}\tag{6.152}$$

下面是一个耦合式求解器的范例：

$$\begin{bmatrix} * & * & 0 & 0 & * & * & 0 & 0 \\ * & * & * & 0 & * & * & * & 0 \\ 0 & * & * & * & 0 & * & * & * \\ 0 & 0 & * & * & 0 & 0 & * & * \\ * & * & 0 & 0 & * & * & 0 & 0 \\ * & * & * & 0 & * & * & * & 0 \\ 0 & * & * & * & 0 & * & * & * \\ 0 & 0 & * & * & 0 & 0 & * & * \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{bmatrix} \quad (6.153)$$

很明显，耦合式求解器一次性的求解出相关变量，并且为一个 block 矩阵，并且变量通过一个矩阵系统耦合起来，耦合性更强。

6.13 显性离散、隐性离散

在离散 CFD 方程的时候，可以对变量进行显性离散以及隐性离散。二者的区别主要体现对离散后的矩阵系数的影响上。考虑速度 \mathbf{U} 方程：

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U} \mathbf{U}) - \nabla \cdot (\nu \nabla \mathbf{U}) = -\nabla \frac{p}{\rho}, \quad (6.154)$$

其中的每一项在离散的过程中，都可以区分为显性离散以及隐性离散。其中的时间项，隐性离散可以写为：

$$\frac{\partial \mathbf{U}}{\partial t} = \frac{\mathbf{U}^{t+\Delta t} - \mathbf{U}^t}{\Delta t} \quad (6.155)$$

其中的 $\mathbf{U}^{t+\Delta t}$ 为未知变量，进入到矩阵系统成为待求变量。 $1/\Delta t$ 成为了矩阵系统的对角线系数。 $\mathbf{U}^t/\Delta t$ 作为已知量，进入到矩阵系统的源项部分。如果时间项显性离散，则有：

$$\frac{\partial \mathbf{U}}{\partial t} = \frac{\mathbf{U}^t - \mathbf{U}^{t-\Delta t}}{\Delta t} \quad (6.156)$$

方程(6.156)的右侧全部进入矩阵源项部分。不对矩阵系数产生影响。

6.14 Jacobian

给定自变量 $\mathbf{x} = [x_1, x_2, \dots, x_n]$ 以及待求函数 $\mathbf{y} = [y_1, y_2, \dots, y_n]$ ，假定函数关系为：

$$\mathbf{y} = f(\mathbf{x}) \quad (6.157)$$

也即：

$$\begin{aligned} y_1 &= f_1(x_1, x_2, \dots, x_n), \\ y_2 &= f_2(x_1, x_2, \dots, x_n), \\ &\dots, \\ y_n &= f_n(x_1, x_2, \dots, x_n), \end{aligned} \tag{6.158}$$

对于上述的方程组，可以从数学上定义其 Jacobian 矩阵为：

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{6.159}$$

Jacobian 在流体力学中的应用主要涉及到体积微元以及坐标转换。考虑原始坐标系 (α, β, γ) ，在将体积 V_0 投影至新坐标系 (x, y, z) 之后，其体积将会发生变化。Jacobian 矩阵被定义为：

$$\mathbf{J} = \frac{\partial(x, y, z)}{\partial(\alpha, \beta, \gamma)} = \begin{pmatrix} \frac{\partial x}{\partial \alpha} & \frac{\partial x}{\partial \beta} & \frac{\partial x}{\partial \gamma} \\ \frac{\partial y}{\partial \alpha} & \frac{\partial y}{\partial \beta} & \frac{\partial y}{\partial \gamma} \\ \frac{\partial z}{\partial \alpha} & \frac{\partial z}{\partial \beta} & \frac{\partial z}{\partial \gamma} \end{pmatrix} \tag{6.160}$$

Jacobian 被定义为 Jacobian 矩阵的行列式：

$$J = |\mathbf{J}| \tag{6.161}$$

这样，体积变化可以表示为：

$$dV = \frac{\partial(x, y, z)}{\partial(\alpha, \beta, \gamma)} d\alpha d\beta d\gamma = J dV_0 \tag{6.162}$$

同时参考方程(3.48)以及速度散度的定义，有：

$$\nabla \cdot \mathbf{U} = \frac{1}{J} \frac{D\mathbf{J}}{Dt} \tag{6.163}$$

可参考其他资料获取更详细的讨论 [3].

6.15 非牛顿流体

上文的讨论均假定流体为牛顿流体，即流体的粘度为常数（剪切应力和形变率的比值为定值），剪切应力越大，形变的速度呈现同比例的变化。也即方程(3.81)中的 μ 为定值。在非牛顿流体中， μ 表示为一个跟时空有关的变量。非牛顿流体在日常生活中非常常见，如果酱、洗洁精、血液、泥浆、蛋清、等。举例，淀粉溶液在轻轻搅动的时候较易，快速搅动的时候较难。这是因为淀粉溶液的粘度和剪切应力呈现非线性的关系。剪切应力越大，粘

度越高，形变率越慢，越难以搅动。非牛顿流体可以按照以下标准进行区分，剪切变稀流体的粘度随着剪切应力增加而降低，剪切增稠流体的粘度随着剪切应力增加而增加，宾汉流体在剪切力小于一定值的情况下，呈现固体的特性，粘弹性流体的粘度具备粘性，同时具备固体类似的弹性。图6.14为开源 CFD 软件 OpenFOAM 模拟的粘弹性两相流流动。



图 6.14: 通过开源 CFD 软件 OpenFOAM 模拟的粘弹性流体两相流 [33]。

在 CFD 中，牛顿流体的动量方程中的剪切应力 τ 可以通过方程(3.81)转变为 \mathbf{U} 项进而封闭。但由于粘度为一个随时空变化的量，因此动量方程中会存在一个附加项。现对此进行推导。考虑粘性项 $\nabla \cdot (\mu(\nabla \mathbf{U} + \nabla \mathbf{U}^T))$ ，有：

$$\nabla \cdot (\mu(\nabla \mathbf{U} + \nabla \mathbf{U}^T)) = \nabla \cdot (\mu \nabla \mathbf{U}) + \nabla \cdot (\mu \nabla \mathbf{U}^T) \quad (6.164)$$

现重点关注 $\nabla \cdot (\mu \nabla \mathbf{U}^T)$ ，将其展开（考虑速度仅有 2 个分量）：

$$\begin{aligned} \nabla \cdot (\mu \nabla \mathbf{U}^T) &= \nabla \cdot \begin{bmatrix} \mu \frac{\partial u_1}{\partial x} & \mu \frac{\partial u_1}{\partial y} \\ \mu \frac{\partial u_2}{\partial x} & \mu \frac{\partial u_2}{\partial y} \end{bmatrix} = \left[\begin{array}{l} \frac{\partial}{\partial x} \left(\mu \frac{\partial u_1}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial u_2}{\partial x} \right) \\ \frac{\partial}{\partial x} \left(\mu \frac{\partial u_1}{\partial y} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial u_2}{\partial y} \right) \end{array} \right] \\ &= \left[\begin{array}{l} \frac{\partial \mu}{\partial x} \frac{\partial u_1}{\partial x} + \mu \frac{\partial}{\partial x} \left(\frac{\partial u_1}{\partial x} \right) + \frac{\partial \mu}{\partial y} \frac{\partial u_2}{\partial x} + \mu \frac{\partial}{\partial y} \left(\frac{\partial u_2}{\partial x} \right) \\ \frac{\partial \mu}{\partial x} \frac{\partial u_1}{\partial y} + \mu \frac{\partial}{\partial x} \left(\frac{\partial u_1}{\partial y} \right) + \frac{\partial \mu}{\partial y} \frac{\partial u_2}{\partial y} + \mu \frac{\partial}{\partial y} \left(\frac{\partial u_2}{\partial y} \right) \end{array} \right] = \left[\begin{array}{l} \frac{\partial \mu}{\partial x} \frac{\partial u_1}{\partial x} + \frac{\partial \mu}{\partial y} \frac{\partial u_2}{\partial x} \\ \frac{\partial \mu}{\partial x} \frac{\partial u_1}{\partial y} + \frac{\partial \mu}{\partial y} \frac{\partial u_2}{\partial y} \end{array} \right] \\ &= \begin{bmatrix} \frac{\partial u_1}{\partial x}, \frac{\partial u_2}{\partial x} \\ \frac{\partial u_1}{\partial y}, \frac{\partial u_2}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial \mu}{\partial x} \\ \frac{\partial \mu}{\partial y} \end{bmatrix} = \nabla \mathbf{U} \cdot \nabla \mu \quad (6.165) \end{aligned}$$

因此，非牛顿流体动量方程中的粘性项的最终形式为：

$$\nabla \cdot (\mu(\nabla \mathbf{U} + \nabla \mathbf{U}^T)) = \nabla \cdot (\mu \nabla \mathbf{U}) + \nabla \mathbf{U} \cdot \nabla \mu \quad (6.166)$$

其中 $\nabla \mathbf{U} \cdot \nabla \mu$ 在代码植入的过程中要体现出来。另一方面，需要植入与速度相关的粘度变化，也即本构方程。在一些较为简单的非牛顿流体中（如宾汉流体），可以通过将 μ 表示为关于时空的函数进行计算植入。在一些复杂的非牛顿流体中（如粘弹性流体的 Giesekus 模型），则需要植入 τ 的传输方程，然后对其求解散度项中代入到动量方程中再进行显性离散计算。

目前在 OpenFOAM 中植入的经典的非牛顿流体粘度计算模型主要有 powerLaw、HerschelBulkley、CrossPowerLaw、Casson、BirdCarreau 模型。在计算粘度之前，需要计算形变速率标量 γ :

$$\gamma = \sqrt{2\sqrt{\mathbf{S} : \mathbf{S}}} \quad (6.167)$$

其中 \mathbf{S} 在方程(7.105)中已定义。Bird-Carreau 模型定义粘度为:

$$\nu = \nu_\infty + (\nu_0 - \nu_\infty) [1 + (k\dot{\gamma})^a]^{(n-1)/a} \quad (6.168)$$

其中系数 a 默认值为 2。其他的参数需要用户自行给定。CrossPowerLaw 模型定义粘度为:

$$\nu = \nu_\infty + \frac{\nu_0 - \nu_\infty}{1 + (m\dot{\gamma})^n} \quad (6.169)$$

其中所有参数需要用户自行给定。

6.16 算子分裂法

CFD 算法中，经常会讨论到算子分裂法。算子分裂的英文解释为 Operator Splitting。对于 CFD 中的方程，有些情况下，按项去求解会比较简单。考虑下面的无量纲方程:

$$\frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} = \frac{\partial}{\partial x} \left(\frac{\partial \phi}{\partial x} \right) \quad (6.170)$$

在求解过程中，可以按项来处理，先求

$$\frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} = 0 \quad (6.171)$$

在获得解 ϕ^* 后，再求

$$\frac{\partial \phi}{\partial t} = \frac{\partial}{\partial x} \left(\frac{\partial \phi}{\partial x} \right) \quad (6.172)$$

获得解 ϕ^{**} 。这样求解有一个非常特殊的好处。那就是可以在不同的阶段调用不同的格式。例如在方程(6.171)中，其为一个纯对流方程，可以调用显性迎风高阶格式。在方程(6.172)中，其为一个含有扩散项的时间推进方程，可调用较大时间步的隐性格式。

6.17 总温、驻点温度

对于流动的流体，可以定义几种不同的温度。例如，静温用来衡量无流动的流体的温度。静温的大小主要取决于分子动力学，其取决于分子的热运动。对于流动的流体，可以定义动温，其与流动的动能有关。同时，也可以定义总温，即静温与动温之和。流体的总温取决于马赫数。如果在计算域内突然出现静止点（驻点，局部速度为 0 或者近似于 0 的区域）。驻点处的温度也被称之为驻点温度，其与总温几乎相同。这是因为在驻点处，可以使流体以绝热过程完全静止时，那么它的动能将全部转化为内能，这时，动温为 0，总温即驻点温度。流体的总温与静温的关系为：

$$T_0 = \left(1 + \frac{\gamma - 1}{2} Ma^2\right) T \quad (6.173)$$

其中 T_0 表示总温， T 表示静温， γ 表示热容比， Ma 表示马赫数。总温在经过激波区域，其值不变。

第七章 OpenFOAM 模型速查

7.1 OpenFOAM 中的湍流模型

7.1.1 RANS 湍流模型

线性-可压缩-多相 kEpsilon 目前应用最广泛的模型就是 kEpsilon 湍流模型。最早的工作可以追溯到 1945 年抗日战争时期诞生于国立清华大学 [25]。依据 k 的数学定义，可以得到湍流动能的准确的传输方程，该方程右侧存在湍流动能耗散项或者湍流频率项。在 kEpsilon 模型中， k 方程右侧调用湍流动能耗散。在湍流动能耗散已知的情况下， k 方程可解。依据 ε 的定义，可以得到 epsilon 的准确的传输方程，若进一步调用湍流频率 ω 与 ε 的关系，还可以推导出 kOmega 模型。但 epsilon 的准确的传输方程并不封闭。因此 epsilon 方程需要进行大量的模化，并由此产生大量的 kEpsilon 湍流模型变种。OpenFOAM 中的 kEpsilon 模型为 Launder and Spalding 提出的 kEpsilon 湍流模型 [51]。其也被称为标准 kEpsilon 模型。kEpsilon 模型在求解过程中非常稳健因此被广泛应用于各种普适性流动中。kEpsilon 湍流模型最大的缺点是对于存在逆压梯度的流动预测不够准确。这主要是由于 kEpsilon 大部分情况下会低估湍流动能耗散率，也就是预测的湍流尺度过高。kEpsilon 模型为一种高雷诺数模型，仅仅适用于高雷诺数流动。在应用于低雷诺数流动的情况下，需要在壁面附加低雷诺数修正模型。kEpsilon 模型中的 C_μ 为常数，很多工作认为使用一个变量来模化 C_μ 可以获得更好的结果。

$$\begin{aligned} \frac{\partial \alpha \rho \varepsilon}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \varepsilon) - \nabla \cdot (\alpha \rho D_\varepsilon \nabla \varepsilon) &= C_1 \alpha \rho G \frac{\varepsilon}{k} - \left(\left(\frac{2}{3} C_1 - C_3 \right) \alpha \rho (\nabla \cdot \mathbf{U}) \varepsilon \right) - C_2 \alpha \rho \frac{\varepsilon}{k} \varepsilon + S_\varepsilon \\ \frac{\partial \alpha \rho k}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} k) - \nabla \cdot (\alpha \rho D_k \nabla k) &= \alpha \rho G - \frac{2}{3} \alpha \rho (\nabla \cdot \mathbf{U}) k - \alpha \rho \frac{\varepsilon}{k} k + S_k \\ G = \nu_t \left(\nabla \mathbf{U} + \nabla \mathbf{U}^T - \frac{2}{3} (\nabla \cdot \mathbf{U}) \mathbf{I} \right) : \nabla \mathbf{U}, \nu_t &= C_\mu \frac{k^2}{\varepsilon}, C_\mu = 0.09, C_1 = 1.44, C_2 = 1.92, C_3 = 0 \\ \sigma_k = 1, \sigma_\varepsilon = 1.3, D_k &= \frac{\nu_t}{\sigma_k} + \nu, D_\varepsilon = \frac{\nu_t}{\sigma_\varepsilon} + \nu \end{aligned}$$

线性-可压缩-多相 RNGkEpsilon RNGkEpsilon 湍流模型采用重整化 NS 方程进行模化，意图包含小尺度湍流的影响。OpenFOAM 中植入的 RNGkEpsilon 湍流模型来自于

Yakhot 的工作 [98]。在 RNGkEpsilon 被提出时，强调此湍流模型的数学连贯性，批判 kEpsilon 的验证算例不充分，认为 RNGkEpsilon 是一个更普适性的湍流模型。经验表明，RNGkEpsilon 在一些算例中确实对预测结果略有改善 [82]，但求解略微刚性。RNGkEpsilon 和 kEpsilon 模型的方程形式完全一样，只不过模型系数是经过理论分析而来。需要注意的是，目前一些商软中植入了提高版本的更适用于旋转流动的 RNGkEpsilon 湍流模型。在此需要跟 OpenFOAM 的原始版本进行区分（区别主要在于湍流粘度项）。

$$\frac{\partial \alpha \rho \varepsilon}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \varepsilon) - \nabla \cdot (\alpha \rho D_\varepsilon \nabla \varepsilon) = (C_1 - R) \alpha \rho G \frac{\varepsilon}{k} - \left(\left(\frac{2}{3} C_1 - C_3 \right) \alpha \rho (\nabla \cdot \mathbf{U}) \varepsilon \right) - C_2 \alpha \rho \frac{\varepsilon}{k} \varepsilon + S_\varepsilon$$

$$\frac{\partial \alpha \rho k}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} k) - \nabla \cdot (\alpha \rho D_k \nabla k) = \alpha \rho G - \frac{2}{3} \alpha \rho (\nabla \cdot \mathbf{U}) k - \alpha \rho \frac{\varepsilon}{k} k + S_k$$

$$G = \nu_t S_2, S_2 = \left(\nabla \mathbf{U} + \nabla \mathbf{U}^T - \frac{2}{3} (\nabla \cdot \mathbf{U}) \mathbf{I} \right) : \nabla \mathbf{U}, \eta = \frac{\sqrt{|S_2|} k}{\varepsilon}, R = \frac{\eta (-\eta/\eta_0 + 1)}{\beta \eta^3 + 1}, \nu_t = C_\mu \frac{k^2}{\epsilon}$$

$$C_\mu = 0.0845, C_1 = 1.42, C_2 = 1.68, \eta_0 = 4.38, C_3 = 0, \sigma_k = 0.71942, \sigma_\varepsilon = 0.71942, \beta = 0.012$$

$$D_k = \frac{\nu_t}{\sigma_k} + \nu, D_\varepsilon = \frac{\nu_t}{\sigma_\varepsilon} + \nu$$

线性-可压缩-多相 realizableKE realizableKE 湍流模型认为标准的 kEpsilon 湍流模型可能会产生负的正应力，这是不可能的（在 CFD 领域通常被称之为是不可实现的）。realizableKE 湍流模型针对标准湍流模型最重要的两个区别在于 1) C_μ 不再是一个常数而是一个变量，2) 湍流粘度模型基于可实现性理论，可确保法向雷诺应力为正。同样，realizableKE 湍流模型的 k 方程并没有变化，主要变动的是 epsilon 方程。从 realizableKE 湍流模型提出的原始文章数据中可以发现 [79]，realizableKE 湍流模型对旋转流动的预测提高较为明显，对于自由剪切、边界层流动、以及后向台阶流和标准 kEpsilon 模型近似相同。

$$\frac{\partial \alpha \rho \varepsilon}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \varepsilon) - \nabla \cdot (\alpha \rho D_\varepsilon \nabla \varepsilon) = C_1 \alpha \rho \sqrt{S_2} \varepsilon - \frac{C_2 \alpha \rho \varepsilon}{k + \sqrt{\nu \varepsilon}} \varepsilon + S_\varepsilon$$

$$\frac{\partial \alpha \rho k}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} k) - \nabla \cdot (\alpha \rho D_k \nabla k) = \alpha \rho G - \frac{2}{3} \alpha \rho \nabla \cdot \mathbf{U} k - \alpha \rho \frac{\varepsilon}{k} k + S_k$$

$$S_2 = 2 \left| \frac{\nabla \mathbf{U} + (\nabla \mathbf{U})^T}{2} - \frac{1}{3} \text{tr} \left(\frac{\nabla \mathbf{U} + (\nabla \mathbf{U})^T}{2} \right) \mathbf{I} - \frac{1}{3} (\nabla \cdot \mathbf{U}) \mathbf{I} \right|^2, \eta = \frac{\sqrt{S_2} k}{\varepsilon}$$

$$G = \nu_t \left(\nabla \mathbf{U} + \nabla \mathbf{U}^T - \frac{2}{3} (\nabla \cdot \mathbf{U}) \mathbf{I} \right) : \nabla \mathbf{U}$$

$$\mathbf{S} = \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2} - \frac{1}{3} (\nabla \cdot \mathbf{U}) \mathbf{I}, W = \frac{2\sqrt{2}((\mathbf{S} \cdot \mathbf{S}) : \mathbf{S})}{S_2^{3/2}}, \phi_s = \frac{1}{3} \arccos \left(\min \left(\max \left(\sqrt{6}W, -1 \right), 1 \right) \right)$$

$$A_s = \sqrt{6} \cos \phi_s, C_1 = \max \left(\frac{\eta}{5 + \eta}, 0.43 \right), U_s = \left(\frac{S_2}{2} + \left| \frac{\nabla \mathbf{U} - \nabla \mathbf{U}^T}{2} \right|^2 \right)^{\frac{1}{2}}, C_\mu = A_0 + \frac{A_s U_s k}{\varepsilon}$$

$$A_0 = 4, C_2 = 1.9, \sigma_k = 1, \sigma_\varepsilon = 1.2, D_k = \frac{\nu_t}{\sigma_k} + \nu, D_\varepsilon = \frac{\nu_t}{\sigma_\varepsilon} + \nu, \nu_t = \frac{1}{C_\mu} \frac{k^2}{\varepsilon}$$

线性-可压缩-多相-低雷诺数 LaunderSharmaKE 标准 kEpsilon、RNGkEpsilon、以及 realizableKEpsilon 模型最大的缺点就是他们都是高雷诺数湍流模型。在壁面网格 $y^+ > 30$ 的时候，高雷诺数湍流模型需要结合壁面函数来使用。然而如果网格在壁面足够细，例如在 $y^+ < 5$ 的情况下，就可以不结合壁面函数使用高雷诺数湍流模型了么？答案是未必。在这种情况下，高雷诺数湍流模型同样需要附加阻尼函数，即形成低雷诺数湍流模型。LaunderSharmaKE 湍流模型就属于低雷诺数湍流模型。低雷诺数模型跟高雷诺数模型的区别主要在于前者的 C_μ 需要乘以一个系数 f_μ 来体现壁面附近分子粘度的作用。在使用的过程中需要保证 $y^+ \approx 1$ [50]。低雷诺数湍流模型定义湍流雷诺数 Re 为 $k^2/\nu/\varepsilon$ ，在 kEpsilon 模型的基础上建立与湍流雷诺数的关系式，对湍流动能在壁面附近的产生添加阻尼，以及湍流动能耗散率在壁面附近的消耗进行修正，从而可以应用于低雷诺数领域。低雷诺数湍流模型同样被称之为附加阻尼函数 (damping function) 的湍流模型。低雷诺数湍流模型的第一层网格点要布置在粘性支层， y^+ 建议在 1 以下，不建议超过 4。同时建议在 $y^+ < 11.5$ 的区域至少布置 5 个网格点。同时，壁面处的湍流动能边界条件可以赋值为固定值 0 或者一个非常小的值。湍流动能耗散率可以为 0 (LaunderSharmaKE) 或零法向梯度 (LamBremhorstKE)。

$$\begin{aligned} \frac{\partial \alpha \rho \varepsilon}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \varepsilon) - \nabla \cdot (\alpha \rho D_\varepsilon \nabla \varepsilon) &= C_1 \alpha \rho G \frac{\varepsilon}{k} - \left(\left(\frac{2}{3} C_1 - C_3 \right) \alpha \rho (\nabla \cdot \mathbf{U}) \varepsilon \right) - C_2 f_2 \alpha \rho \frac{\varepsilon}{k} \varepsilon + \alpha \rho E + S_\varepsilon \\ \frac{\partial \alpha \rho k}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} k) - \nabla \cdot (\alpha \rho D_k \nabla k) &= \alpha \rho G - \frac{2}{3} \alpha \rho (\nabla \cdot \mathbf{U}) k - \alpha \rho \frac{\varepsilon + D}{k} k + S_k \\ G &= \nu_t \left(\nabla \mathbf{U} + \nabla \mathbf{U}^T - \frac{2}{3} (\nabla \cdot \mathbf{U}) \mathbf{I} \right) : \nabla \mathbf{U}, E = 2 \nu \nu_t |\nabla \nabla \mathbf{U}|^2, f_2 = 1 - 0.3 \exp \left(- \min \left(\frac{k^4}{\nu^2 \varepsilon^2}, 50 \right) \right) \\ D &= 2 \nu \left| \nabla (\sqrt{k}) \right|^2, f_\mu = \exp \left(\frac{-3.4}{\left(1 + \frac{k^2}{50 \nu \varepsilon} \right)^2} \right), \nu_t = C_\mu f_\mu \frac{k^2}{\varepsilon} \\ C_\mu &= 0.09, C_1 = 1.44, C_2 = 1.92, C_3 = 0, \sigma_k = 1, \sigma_\varepsilon = 1.3, D_k = \frac{\nu_t}{\sigma_k} + \nu, D_\varepsilon = \frac{\nu_t}{\sigma_\varepsilon} + \nu \end{aligned}$$

线性-不可压缩-单相-低雷诺数 LamBremhorstKE LamBremhorstKE 湍流模型也为一种湍流模型，其与 LaunderSharmaKE 湍流模型非常类似，实际上，大部分的低雷诺数湍流模型之间的区别仅仅在于 f_μ, f_2, f_1 函数以及模型参数。LamBremhorstKE 湍流模型的 f_μ 与 Re_t 以及 Re_y 均相关，LaunderSharmaKE 湍流模型的 f_μ 仅仅与 Re_t 相关。

$$\begin{aligned} \frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\mathbf{U} \varepsilon) - \nabla \cdot (D_\varepsilon \nabla \varepsilon) &= C_{\varepsilon_1} f_1 G \frac{\varepsilon}{k} - C_{\varepsilon_2} f_2 \frac{\varepsilon}{k} \varepsilon \\ \frac{\partial k}{\partial t} + \nabla \cdot (\mathbf{U} k) - \nabla \cdot (D_k \nabla k) &= G - \frac{\epsilon}{k} k \end{aligned}$$

$$G = \nu_t (\nabla \mathbf{U} + \nabla \mathbf{U}^T) : \nabla \mathbf{U}, \nu_t = C_\mu f_\mu \frac{k^2}{\varepsilon}, R_t = \frac{k^2}{\nu \varepsilon}$$

$$f_1 = 1 + \left(\frac{0.05}{f_\mu} \right)^3, f_2 = 1 - \exp(-R_t^2), f_\mu = \left(1 - \exp \left(-0.0165 \frac{\sqrt{k}y}{\nu} \right) \right)^2 \left(1 + \frac{20.5}{R_t} \right)$$

$$C_\mu = 0.09, C_{\varepsilon_1} = 1.44, C_{\varepsilon_2} = 1.92, \sigma_\varepsilon = 1.3, D_k = \nu_t + \nu, D_\varepsilon = \frac{\nu_t}{\sigma_\varepsilon} + \nu$$

线性-不可压缩-单相-低雷诺数 LienLeschziner 前述低雷诺数湍流模型的湍流动能耗散率方程存在一切共同点与区别，LamBremhorstKE 的 ε 传输方程中存在 f_1, f_2 项，LaunderSharmaKE 中存在 f_2, E 项。LienLeschziner 湍流模型则同时考虑了 f_1, f_2, E 项 ($f_1 = 1$)。其中 f_1, E 项主要影响壁面附近湍流变量的峰值大小。 f_2 项的作用主要限制在粘性支层内，起到一种阻尼的作用。

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\mathbf{U} \varepsilon) - \nabla \cdot (D_\varepsilon \nabla \varepsilon) = C_{\varepsilon,1} G \frac{\varepsilon}{k} - C_{\varepsilon,2} f_2 \frac{\varepsilon}{k} \varepsilon + E$$

$$\frac{\partial k}{\partial t} + \nabla \cdot (\mathbf{U} k) - \nabla \cdot (D_k \nabla k) = G - \frac{\varepsilon}{k} k$$

$$G = \nu_t (\nabla \mathbf{U} + \nabla \mathbf{U}^T) : \nabla \mathbf{U}, R_t = \frac{k^2}{\nu \varepsilon}, \nu_t = C_\mu f_\mu \frac{k^2}{\varepsilon}, f_\mu = \frac{1 - e^{-A_\mu y^*}}{1 - e^{-A_\varepsilon y^*}}, f_2 = 1 - 0.3 e^{-R_t^2}$$

$$y^* = \frac{\sqrt{k}y}{\nu}, le = \kappa y (1 - e^{-A_\varepsilon y^*}), E = C_{\varepsilon,2} C_\mu^{0.75} \left(\frac{f_2 \sqrt{k} \varepsilon}{le} \right) e^{-A_E y^{*2}}, D_\varepsilon = \frac{\nu_t}{\sigma_\varepsilon} + \nu, D_k = \frac{\nu_t}{\sigma_k} + \nu$$

$$C_{\varepsilon,1} = 1.44, C_{\varepsilon,2} = 1.92, \sigma_k = 1, \sigma_\varepsilon = 1.3, C_\mu = 0.09, \kappa = 0.41, A_\mu = 0.016$$

$$A_\varepsilon = 0.263, A_E = 0.00222$$

非线性-不可压缩-单相-高雷诺数 ShihQuadraticKE 前述 kEpsilon 以及变种均调用的各向同性 Boussinesq 假定来进行模化雷诺应力。在充分发展的条件下会导致正应力相同（例如在二维情况下 $\tau_{11} = \tau_{22} = 0$ ）。ShihQuadraticKE 认为这种各向同性假定会在一些湍流引致二次流的情况下带来较大的偏差 [78]。其认为假定雷诺应力与速度梯度的线性关系是不合理的，需要附加非线性修正。因此在 Shih 的模型中，直接对非线性的雷诺应力 τ_n 进行计算。在计算的过程中，需要获取 k 和 ε 的值，因此需要 k 和 ε 的传输方程。在 OpenFOAM 中，其也被称之为 ShihQuadraticKE。其中的 Quadratic 体现为雷诺应力为速度梯度的 2 次幂关系（存在 $\mathbf{S} \cdot \mathbf{S}$ ）。需要注意的是 ShihQuadraticKE 并没有对 τ_n 进行求解，否则将演变为雷诺应力模型（也被称之为二阶矩模型）。ShihQuadraticKE 湍流模型在推导的过程中并没有考虑分子粘度，因此为一个高雷诺数湍流模型，同样可以调用壁面函数。非线性类模型与线性类模型的区别在于雷诺应力项散度的计算，线性类模型这一项可以通过速度计算而来： $\nabla \cdot \boldsymbol{\tau} = \nabla \cdot (\mu_t \nabla \mathbf{U}) + \nabla \cdot (\mu_t \text{dev2}(\nabla \mathbf{U}^T))$ ，非线性模型除了要考虑线性贡献，还需要考虑非线性的贡献： $\nabla \cdot \boldsymbol{\tau} + \nabla \cdot \boldsymbol{\tau}_n$ 。

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\mathbf{U} \varepsilon) - \nabla \cdot (D_\varepsilon \nabla \varepsilon) = C_{\varepsilon,1} G \frac{\varepsilon}{k} - C_{\varepsilon,2} \frac{\varepsilon}{k} \varepsilon$$

$$\frac{\partial k}{\partial t} + \nabla \cdot (\mathbf{U}k) - \nabla \cdot (D_k \nabla k) = G - \frac{\varepsilon}{k}k$$

$$G = \nu_t (\nabla \mathbf{U} + \nabla \mathbf{U}^T - \tau_n) : \nabla \mathbf{U}, \mathbf{S} = \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2}, \mathbf{W} = \frac{\nabla \mathbf{U} - \nabla \mathbf{U}^T}{2}, \bar{s} = \sqrt{2} \frac{k}{\varepsilon} |\mathbf{S}|, \bar{w} = \sqrt{2} \frac{k}{\varepsilon} |\mathbf{W}|$$

$$\boldsymbol{\tau}_n = \frac{k^3}{\varepsilon^2(C_\beta + \bar{s}^3)} \left(C_{\beta_1} \left(\mathbf{S} \cdot \mathbf{S} - \frac{1}{3} \text{tr}(\mathbf{S} \cdot \mathbf{S}) \mathbf{I} \right) \right.$$

$$\left. + C_{\beta_2} (\mathbf{S} \cdot \mathbf{W} + (\mathbf{S} \cdot \mathbf{W})^T) + C_{\beta_3} \left(\frac{\mathbf{W} \cdot \mathbf{W} + (\mathbf{W} \cdot \mathbf{W})^T}{2} - \frac{1}{3} \text{tr}(\mathbf{W} \cdot \mathbf{W}) \mathbf{I} \right) \right)$$

$$C_\mu = \frac{2/3}{C_{\mu_1} + \bar{s} + C_{\mu_2} \bar{w}}, \nu_t = C_\mu \frac{k^2}{\varepsilon}, C_{\varepsilon,1} = 1.44, C_{\varepsilon,2} = 1.92, \sigma_k = 1$$

$$\sigma_\varepsilon = 1.3, C_{\mu_1} = 1.25, C_{\mu_2} = 0.9, C_\beta = 1000, C_{\beta_1} = 3, C_{\beta_2} = 15, C_{\beta_3} = -19, D_k = \frac{\nu_t}{\sigma_k} + \nu, D_\varepsilon = \frac{\nu_t}{\sigma_\varepsilon} + \nu$$

非线性-不可压缩-单相-LienCubicKE

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\mathbf{U} \varepsilon) - \nabla \cdot (D_\varepsilon \nabla \varepsilon) = C_{\varepsilon,1} G \frac{\varepsilon}{k} - C_{\varepsilon,2} f_2 \frac{\varepsilon}{k} + E$$

$$\frac{\partial k}{\partial t} + \nabla \cdot (\mathbf{U}k) - \nabla \cdot (D_k \nabla k) = G - \frac{\varepsilon}{k}k$$

$$G = \nu_t (\nabla \mathbf{U} + \nabla \mathbf{U}^T - \tau_n) : \nabla \mathbf{U}, y^* = \frac{\sqrt{k}y}{\nu}, \mathbf{S} = \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2}, \mathbf{W} = \frac{\nabla \mathbf{U} - \nabla \mathbf{U}^T}{2}$$

$$\bar{s} = \sqrt{2} \frac{k}{\varepsilon} |\mathbf{S}|, \bar{w} = \sqrt{2} \frac{k}{\varepsilon} |\mathbf{W}|, C_\mu = \frac{2/3}{C_{\mu_1} + \bar{s} + C_{\mu_2} \bar{w}}, \nu_t = C_\mu f_\mu \frac{k^2}{\varepsilon}$$

$$\tau_n = f_\mu k \frac{k^2 / \varepsilon^2}{C_\beta + \bar{s}^3}$$

$$\left(C_{\beta_1} \left(\mathbf{S} \cdot \mathbf{S} - \frac{1}{3} \text{tr}(\mathbf{S} \cdot \mathbf{S}) \mathbf{I} \right) + C_{\beta_2} (\mathbf{S} \cdot \mathbf{W} + (\mathbf{S} \cdot \mathbf{W})^T) + C_{\beta_3} \left(\frac{\mathbf{W} \cdot \mathbf{W} + (\mathbf{W} \cdot \mathbf{W})^T}{2} - \frac{1}{3} \text{tr}(\mathbf{W} \cdot \mathbf{W}) \mathbf{I} \right) \right)$$

$$- f_\mu k \left(\frac{C_\mu k}{\varepsilon} \right)^3 (C_{\gamma,1} |\mathbf{S}|^2 - C_{\gamma,2} |\mathbf{W}|^2) \mathbf{S} - f_\mu k \left(\frac{C_\mu k}{\varepsilon} \right)^3 C_{\gamma,4} ((\mathbf{S} \cdot \mathbf{S}) \cdot \mathbf{W} + ((\mathbf{S} \cdot \mathbf{S}) \cdot \mathbf{W})^T)$$

$$f_\mu = (1 - e^{-A_\mu y^*}) \left(1 + \frac{2\kappa}{C_\mu^{0.75} y^*} \right), f_2 = 1 - 0.3e^{-R_t^2}, R_t = \frac{k^2}{\nu \varepsilon}, E = C_{\varepsilon,2} C_\mu^{0.75} \left(\frac{f_2 \sqrt{k} \varepsilon}{l e} \right) e^{-A_E y^{*2}}$$

$$l e = \frac{\kappa y}{1 + \frac{2\kappa}{C_\mu^{0.75} y^*}}, C_{\varepsilon,1} = 1.44, C_{\varepsilon,2} = 1.92, \sigma_k = 1, \sigma_\varepsilon = 1.3$$

$$C_{\mu_1} = 1.25, C_{\mu_2} = 0.9, C_\beta = 1000, C_{\beta_1} = 3, C_{\beta_2} = 15, C_{\beta_3} = -19, C_{\gamma,1} = 16, C_{\gamma,2} = 16$$

$$C_{\gamma,4} = -80, C_\mu = 0.09, \kappa = 0.41, D_k = \frac{\nu_t}{\sigma_k} + \nu, D_\varepsilon = \frac{\nu_t}{\sigma_\varepsilon} + \nu, A_\nu = 0.0198, A_E = 0.00375$$

线性-可压缩-多相 BuoyantKE kEpsilon 湍流模型是基于常密度流动而开发。对于一些密度差异比较大的流动，如燃烧、暖通等领域，密度变化会对湍流结构产生影响。因此，传

统 kEpsilon 模型需要考虑浮升力的作用。因此，BuoyantKE 湍流模型在 **kEpsilon** 模型上添加 k, ε 方程源项：

$$S_k = -G_c k, S_\varepsilon = -\frac{C_1 \tanh(|v|/u)}{G_c} \varepsilon, v = \frac{\mathbf{g}}{|\mathbf{g}|} \cdot \mathbf{U}, u = \left| \mathbf{U} - \frac{\mathbf{g}}{|\mathbf{g}|} v \right|, G_c = C_g C_\mu \alpha k (\mathbf{g} \cdot \nabla \rho) / \varepsilon, C_g = 1$$

可压缩-多相-雷诺应力 LRR LRR 是一个雷诺应力传输模型 [49]，也即二阶矩模型。同时为一个高雷诺数湍流模型，壁面处可以附加壁面函数求解。由于雷诺应力具有 6 个分量，因此雷诺应力湍流模型至少需要求解六个传输方程，若需要包含传热，可能需要求解更多的方程。传统的 kEpsilon 类方法在应力场变化比较复杂，尤其是那些旋转较为明显，或者包含体积力以及二次流的情况下预测偏差较大。在使用雷诺应力类模型的时候，需要给定雷诺应力边界条件以及初始条件。一般情况下，进口的雷诺应力可以通过湍流强度以及特征长度（比如管径）来获得，如 [94]： $\tau_{11} = k, \tau_{22} = 0.5k, \tau_{33} = 0.5k, \tau_{12} = \tau_{21} = \tau_{13} = 0$ 。历史上，雷诺应力传输模型并没有广为使用。其主要原因是相对于两方程模型，雷诺应力传输模型 1) 需要求解六个传输方程，更加不稳定以及消耗更多的计算资源，2) 在一些工况下可以对预测结果具有较大的改善，但在一些情况下反而较差。3) 雷诺应力模型中的源项很多都需要进行模化，尤其是其中的压力应力再分配项，同时，很多项的物理机制尚不明确。Peter Bradshaw 曾这样评价雷诺应力传输模型 [8]

“RSM is the most disappointing thing for the modelling community because we put so much hope into it, but tends out to be not as successful.”

$$\begin{aligned} \frac{\partial \alpha \rho \varepsilon}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \varepsilon) - \nabla \cdot (\alpha \rho \mathbf{D}_{\varepsilon,eff} \cdot \nabla \varepsilon) &= C_{\varepsilon,1} \alpha \rho G \frac{\varepsilon}{k} - C_{\varepsilon,2} \alpha \rho \frac{\varepsilon}{k} \varepsilon \\ \frac{\partial \alpha \rho \mathbf{R}}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \mathbf{R}) - \nabla \cdot (\alpha \rho \mathbf{D}_{\mathbf{R},eff} \cdot \nabla \mathbf{R}) + C_1 \alpha \rho \frac{\varepsilon}{k} \mathbf{R} &= \alpha \rho \mathbf{P} - \left(\frac{2}{3} (1 - C_1) \mathbf{I} \right) \alpha \rho \varepsilon - C_2 \alpha \rho \left(\mathbf{P} - \frac{1}{3} \text{tr}(\mathbf{P}) \mathbf{I} \right) \\ \mathbf{P} = -(\mathbf{R} \cdot \nabla \mathbf{U} + (\mathbf{R} \cdot \nabla \mathbf{U})^T), k = 0.5 (\mathbf{R}_{xx} + \mathbf{R}_{yy} + \mathbf{R}_{zz}), G = 0.5 |\mathbf{P}_{xx} + \mathbf{P}_{yy} + \mathbf{P}_{zz}| \\ \mathbf{D}_{\varepsilon,eff} = C_{\varepsilon} \frac{k}{\varepsilon} \mathbf{R} + \nu \mathbf{I}, \mathbf{D}_{\mathbf{R},eff} = C_s \frac{k}{\varepsilon} \mathbf{R} + \nu \mathbf{I} \\ C_\mu = 0.09, C_1 = 1.8, C_2 = 0.6, C_{\varepsilon,1} = 1.44, C_{\varepsilon,2} = 1.92, C_s = 0.25, C_\varepsilon = 0.15, \kappa = 0.41 \\ C_{r1} = 0.5, C_{r2} = 0.3 \end{aligned}$$

可压缩-多相-雷诺应力 SSG

$$\begin{aligned} \frac{\partial \alpha \rho \varepsilon}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \varepsilon) - \nabla \cdot (\alpha \rho \mathbf{D}_{\varepsilon,eff} \cdot \nabla \varepsilon) &= C_{\varepsilon,1} \alpha \rho G \frac{\varepsilon}{k} - C_{\varepsilon,2} \alpha \rho \frac{\varepsilon}{k} \varepsilon \\ \frac{\partial \alpha \rho \mathbf{R}}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \mathbf{R}) - \nabla \cdot (\alpha \rho \mathbf{D}_{\mathbf{R},eff} \cdot \nabla \mathbf{R}) + \left(\frac{C_1 \varepsilon + C_{1,s} G}{2} \frac{\alpha \rho}{k} \right) \mathbf{R} &= \alpha \rho \mathbf{P} - (((2 - C_1) \varepsilon - C_{1,s} G) \alpha \rho) \frac{1}{3} \mathbf{I} + C_2 \alpha \rho \varepsilon \left(\mathbf{b} \cdot \mathbf{b} - \frac{1}{3} \text{tr}(\mathbf{b} \cdot \mathbf{b}) \mathbf{I} \right) + \alpha \rho k (C_3 - C_{3,s} |\mathbf{b}|) \left(\mathbf{S} - \frac{1}{3} \text{tr}(\mathbf{S}) \mathbf{I} \right) \\ &\quad + C_4 \alpha \rho k \left(\mathbf{b} \cdot \mathbf{S} + (\mathbf{b} \cdot \mathbf{S})^T - \frac{2}{3} \text{tr}(\mathbf{b} \cdot \mathbf{S}) \mathbf{I} \right) + C_5 \alpha \rho k \left(\mathbf{b} \cdot \boldsymbol{\Omega} + (\mathbf{b} \cdot \boldsymbol{\Omega})^T \right) \end{aligned}$$

$$\mathbf{P} = -(\mathbf{R} \cdot \nabla \mathbf{U} + (\mathbf{R} \cdot \nabla \mathbf{U})^T), k = 0.5 (\mathbf{R}_{xx} + \mathbf{R}_{yy} + \mathbf{R}_{zz}), \boldsymbol{\Omega} = \frac{\nabla \mathbf{U} - \nabla \mathbf{U}^T}{2}, \mathbf{S} = \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2}$$

$$\mathbf{b} = \frac{\mathbf{R} - \frac{1}{3}\text{tr}(\mathbf{R})\mathbf{I}}{2k}, G = 0.5|\mathbf{P}_{xx} + \mathbf{P}_{yy} + \mathbf{P}_{zz}|, \mathbf{D}_{\varepsilon,eff} = C_{\varepsilon} \frac{k}{\varepsilon} \mathbf{R} + \nu \mathbf{I}, \mathbf{D}_{R,eff} = C_s \frac{k}{\varepsilon} \mathbf{R} + \nu \mathbf{I}, \nu_t = C_{\mu} \frac{k^2}{\varepsilon}$$

$$C_{\mu} = 0.09, C_1 = 3.4, C_{1,s} = 1.8, C_2 = 4.2, C_3 = 0.8, C_{3,s} = 1.3$$

$$C_4 = 1.25, C_5 = 0.4, C_{\varepsilon,1} = 1.44, C_{\varepsilon,2} = 1.92, C_s = 0.25, C_{\varepsilon} = 0.15$$

线性-可压缩-多相 SpalartAllmaras SpalartAllmaras 湍流模型是一个具备问题属性的湍流模型。其在提出的时候就是针对空气动力学领域，例如近音速流动以及机翼设计。本质上，两方程湍流模型也是为了通过两个变量来求解湍流粘度。SpalartAllmaras 湍流模型直接对湍流粘度构建传输方程，因此不需要其他的传输方程，构成一方程湍流模型。

$$\frac{\partial \alpha \rho \tilde{\nu}}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \tilde{\nu}) - \nabla \cdot (\alpha \rho D_{\tilde{\nu}} \nabla \tilde{\nu}) - \frac{C_{b2}}{\sigma_{\nu_t}} \alpha \rho |\nabla \tilde{\nu}|^2 = C_{b1} \alpha \rho \tilde{\nu} \tilde{S} - \frac{C_{w1} \alpha \rho f_w \tilde{\nu}^2}{y^2}$$

$$f_w = g \left(\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right)^{\frac{1}{6}}, g = r + C_{w2} (r^6 - r), \chi = \frac{\tilde{\nu}_t}{\nu}, \Omega = \sqrt{2} \left| \frac{\nabla \mathbf{U} - \nabla \mathbf{U}^T}{2} \right|, \nu_t = \tilde{\nu}_t f_{v1}$$

$$\tilde{S} = \max \left(\Omega + \frac{f_{v2} \tilde{\nu}_t}{\kappa^2 y^2}, C_s \Omega \right), f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}$$

$$r = \min \left(\frac{\tilde{\nu}_t}{\tilde{S} \kappa^2 y^2}, 10 \right), D_{\tilde{\nu}} = \frac{\tilde{\nu}_t + \nu}{\sigma_{\nu_t}}, \sigma_{\nu_t} = 0.666666, \kappa = 0.41, C_{b1} = 0.1355$$

$$C_{b2} = 0.622, C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{1 + C_{b2}}{\sigma_{\nu_t}}, C_{w2} = 0.3, C_{w3} = 2, C_{v1} = 7.1, C_s = 0.3$$

线性-可压缩-多相-低雷诺数 v2f 传统的 kEpsilon 模型认为壁面附近的特征速度尺度跟湍流动能 k 正相关。但研究工作表示这可能存在缺陷，v2f 湍流模型认为壁面附近的特征速度尺度与壁面法向应力 $\overline{\mathbf{U}'_y^2}$ （在下文的方程中用 v_2 表示）正相关。因此，相关的湍流模型若要更好的捕获近壁能力，需要建立 $k, \varepsilon, \overline{\mathbf{U}'_y^2}$ 三变量方程。因此 v2f 湍流模型的原始版本也被称之为 $k - \varepsilon - \overline{\mathbf{U}'_y^2}$ 湍流模型 [32]。同时，大部分其他类型的低雷诺数湍流模型都需要定义一个 f_{μ} 函数来修正湍流粘度，然而这种方法由于高度的非线性会导致数值刚性。v2f 湍流模型则不需要阻尼函数。针对 $\overline{\mathbf{U}'_y^2}$ 的求解可以实现类似的动理学意义上的阻断（kinematic blocking）[62]。该模型在 kEpsilon 模型的基础上，增加两个传输方程，一个是 $\overline{\mathbf{U}'_y^2}$ 传输方程，一个是 f 传输方程。壁面法向应力可以理解为流向垂直方向的速度波动。也可用于表征壁面附近的各向异性。其大小与壁面距离高度相关。因此与 $\overline{\mathbf{U}'_y^2}$ 相关的项可以理解考虑为壁面阻尼后的效果。 f 的定义为 $\overline{\mathbf{U}'_y^2}$ 传输方程的生成项与湍流动能 k 的比值。因此，若在 v2f 湍流模型为一个低雷诺数湍流模型，不需要调用壁面阻尼函数。在使用 v2f 湍流模型的过程中，需要给定边界条件。进口处的 $\overline{\mathbf{U}'_y^2}$ 可以设置为近似等于 $2/3k$ ，

进口处的 f 则设置为零法向梯度。出口处的 $\overline{\mathbf{U}'^2}$ 可以设置为零法向梯度，出口处的 f 则设置为一个较小的固定值。在壁面处， $\overline{\mathbf{U}'^2}$ 可以设置为 0， f 设置为一个非常小的值。

$$\frac{\partial \alpha \rho \varepsilon}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \varepsilon) - \nabla \cdot (\alpha \rho D_\varepsilon \nabla \varepsilon) = C_{\varepsilon,1} \alpha \rho \frac{G}{Ts} - \left(\left(\frac{2}{3} C_{\varepsilon,1} + C_{\varepsilon,3} \right) \alpha \rho (\nabla \cdot \mathbf{U}) \varepsilon \right) - C_{\varepsilon,2} \alpha \frac{\rho}{Ts} \varepsilon + S_\varepsilon$$

$$\frac{\partial \alpha \rho k}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} k) - \nabla \cdot (\alpha \rho D_k \nabla k) = \alpha \rho G - \frac{2}{3} \alpha \rho (\nabla \cdot \mathbf{U}) k - \alpha \rho \frac{\epsilon}{k} k + S_k$$

$$-\nabla \cdot (\nabla f) = -\frac{1}{L_2} f - \frac{1}{L_2 k} (v_\alpha - C_2 G)$$

$$\frac{\partial \alpha \rho v_2}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} v_2) - \nabla \cdot (\alpha \rho D_k \nabla v_2) = \alpha \rho \min(kf, C_2 G - v_\alpha) - N \alpha \rho \frac{\varepsilon}{k} v_2$$

$$C_{\varepsilon,1} = 1.4 \left(1 + 0.05 \min \left(\sqrt{\frac{k}{v_2}}, 100 \right) \right), L_2 = L_s^2, Ts = \max \left(\frac{k}{\varepsilon}, 6 \sqrt{\frac{\nu}{\varepsilon}} \right)$$

$$L_s = C_l \max \left(\frac{k^{1.5}}{\varepsilon}, C_\eta \left(\frac{\nu^3}{\varepsilon} \right)^{0.25} \right), S_2 = 2 \left| \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2} - \frac{1}{3} (\nabla \cdot \mathbf{U}) \mathbf{I} \right|^2$$

$$v_\alpha = \frac{1}{Ts} \left((C_1 - N) v_2 - \frac{2}{3} k (C_1 - 1) \right), \nu_t = \min \left(\frac{C_{\mu k \varepsilon} k^2}{\varepsilon}, C_\mu v_2 Ts \right)$$

$$G = \nu_t S_2, D_k = \frac{\nu_t}{\sigma_k} + \nu, D_\varepsilon = \frac{\nu_t}{\sigma_\varepsilon} + \nu, N = 6, C_\mu = 0.22, C_{\mu k \varepsilon} = 0.09, C_1 = 1.4, C_2 = 0.3, C_l = 0.23$$

$$C_\eta = 70, C_{\varepsilon,2} = 1.9, C_{\varepsilon,3} = -0.33, \sigma_k = 1, \sigma_\varepsilon = 1.3$$

线性-不可压缩-单相-低雷诺数 qZeta¹ qZeta 模型分别定义 q 与 ζ 变量（参考下面的方程），通过对 k 方程以及 ε 方程进行重整就可以得到 q 方程以及 ζ 方程。但是 qZeta 模型针对模型参数进行了调整，以满足低雷诺数的模型适用性。qZeta 湍流模型的边界条件可以参考 kEpsilon 模型，qZeta 模型目前并没有被广泛使用。

$$\frac{\partial \zeta}{\partial t} + \nabla \cdot (\mathbf{U} \zeta) - \nabla \cdot (D_\zeta \nabla \zeta) = (2C_1 - 1) G \frac{\zeta}{q} - \left((2C_2 f_2 - 1) \frac{\zeta}{q} \right) \zeta + E$$

$$\frac{\partial q}{\partial t} + \nabla \cdot (\mathbf{U} q) - \nabla \cdot (D_k \nabla q) = G - \frac{\zeta}{q} q$$

$$G = \frac{\nu_t 2 |(\nabla \mathbf{U} + \nabla \mathbf{U}^T)/2|^2}{2q}, E = \frac{\nu \nu_t}{q} |\nabla \nabla (\mathbf{U})|^2, k = q^2, \varepsilon = 2q\zeta$$

$$\nu_t = C_\mu f_\mu \frac{k^2}{\varepsilon}, f_2 = 1 - 0.3 \exp(-R_t^2), R_t = \frac{qk}{2\nu\zeta}, f_\mu = \exp \left(\frac{-6}{(1 + \frac{R_t}{50})^2} \right) \left(1 + 3 \exp \left(\frac{-R_t}{10} \right) \right)$$

$$C_\mu = 0.09, C_1 = 1.44, C_2 = 1.92, \sigma_\zeta = 1.3, D_q = \nu_t + \nu, D_\zeta = \frac{\nu_t}{\sigma_\zeta} + \nu$$

¹本模型方程中的 f_μ 假定各项异性。

线性-可压缩-多相 kOmega kOmega 湍流模型最早起源于 Kolmogorov 在 1942 年提出的两方程模型。随后 Wolcox、Saffman 等人基于前人的工作对 kOmega 模型进行了大量的改进。Kolmogorov 将 ω 认为是单位体积、单位时间下的能量耗散。Wilcox 简单的认为 ω 与 ε 以及 k 的比值正相关。OpenFOAM-8 中的 kOmega 模型原型为 Wilcox-1988 版²[95]。但其中的区别在于 Wilcox-1988 版原始模型的 $\beta = 0.075, \gamma = 0.55556$ 。在上世纪 70 年代左右，在 kEpsilon 成为最受欢迎的湍流模型的前提下，很多工作发现 kEpsilon 模型在处理逆压梯度、二次流以及一些曲率较高的计算域内的流动并不完美。这就导致研究学者将目光转移到 kOmega 模型。kOmega 模型相对于 kEpsilon 模型更适用于附加逆压梯度的壁面流动以及旋转机械。同时，kOmega 模型本质是一个普适性模型可以用来处理低雷诺数以及高雷诺数，在处理低雷诺数的时候，只要壁面网格足够细致，就不需要附加阻尼函数或低雷诺数壁面函数。但是 kOmega 模型在壁面附近处，即使是低雷诺数流动，通常预测的 ω 也会趋向于无穷大。同时，kOmega 模型对于自由来流的边界条件比较敏感。

$$\begin{aligned} \frac{\partial \alpha \rho k}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} k) - \nabla \cdot (\alpha \rho D_k \nabla k) &= \alpha \rho G - \frac{2}{3} \alpha \rho (\nabla \cdot \mathbf{U}) k - C_\mu \alpha \rho \omega k + S_k \\ \frac{\partial \alpha \rho \omega}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \omega) - \nabla \cdot (\alpha \rho D_\omega \nabla \omega) &= \alpha \rho G \frac{\gamma \omega}{k} - \frac{2}{3} \gamma \alpha \rho (\nabla \cdot \mathbf{U}) \omega - \beta \alpha \rho \omega^2 + S_\omega \\ G &= \nu_t \left(\nabla \mathbf{U} + \nabla \mathbf{U}^T - \frac{2}{3} (\nabla \cdot \mathbf{U}) \mathbf{I} \right) : \nabla \mathbf{U}, \nu_t = \frac{k}{\omega}, C_\mu = 0.09, \beta = 0.072, \alpha_k = 0.5 \\ \alpha_\omega &= 0.5, \gamma = 0.52, D_k = \alpha_k \nu_t + \nu, D_\varepsilon = \alpha_\omega \nu_t + \nu \end{aligned}$$

线性-可压缩-多相 kOmegaSST kOmegaSST 为一种混合 kEpsilon 与 kOmega 模型的混合模型。为了将 kEpsilon 模型与 kOmega 模型的优点相结合并去除相关的缺点（kEpsilon 对于逆压梯度区域的预测不足以及 kOmega 模型的进口条件过于敏感），kOmegaSST 通过混合函数可自动在 kOmega 模型以及 kEpsilon 模型间自动转换。在壁面附近区域，kOmegaSST 模型通过混合函数激活 kOmega 模型，在自由来流以及核心区，kOmegaSST 模型转变为 kEpsilon 模型。由于在壁面附近 kOmegaSST 模型转变为 kOmega 模型，因此 kOmegaSST 模型在低雷诺数流动时不需要像 kEpsilon 模型那样需要低雷诺数湍流模型或阻尼函数。在自由流处，kOmegaSST 模型也对进口条件不敏感。kOmegaSST 模型可以较好的预测逆压梯度以及分离流。模型中的 $B(a, b, c)$ 函数通过 a （也即 F_1 ）的值在 b 与 c 模型系数中进行切换。 F_1 为 \tanh 函数位于 0, 1 之间。kOmegaSST 模型的 $D_k, D_\omega, \beta, \gamma$ 均通过 $B(a, b, c)$ 函数进行系数混合转变。 F_3 函数默认关闭，若激活可以处理粗糙壁面问题 [43]。OpenFOAM-8 中目前植入的版本为 Menter-2003 版 [67]（因为其中的系数 c_1 为 10）³。注意下文公式中的 ω 方程右侧的湍流产生项经过推导后即为 $\alpha \rho \frac{\gamma}{\nu_t} P_k$ 。kOmegaSST 在提出的过程中，参考了二阶矩模型的思想，即将 Bradshaw 假定作为根基，认为边界

²也可参考 NASA 的内容。

³也可参考 NASA 的内容。

层的切应力与湍流动能呈线性关系: $\tau_{12} = \rho a_1 k$, 但在传统两方程模型中, 这个关系为 $\tau_{12} = \rho a_1 k (k_{production}/k_{dissipation})$ 。在逆压梯度附近, $k_{production}/k_{dissipation}$ 远大于 1。因此会导致剪切应力的高估。为了满足 Bradshaw 假定, kOmegaSST 将湍流粘度的定义除掉了 $\sqrt{S_2}$, 即 $\nu_t = a_1 k / \sqrt{S_2}$ 。这样可以保证剪切应力不会过快的变化。进一步的在 ν_t 的定义中的分母中施加 max 函数, 可以保证这个限制器仅仅在逆压梯度附近起作用。这也即为 SST 的来源 [66]。

$$\frac{\partial \alpha \rho k}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} k) - \nabla \cdot (\alpha \rho D_k \nabla k) = \alpha \rho P_k - \frac{2}{3} \alpha \rho k (\nabla \cdot \mathbf{U}) - \alpha \rho \beta^* \omega k + S_k$$

$$\begin{aligned} \frac{\partial \alpha \rho \omega}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \omega) - \nabla \cdot (\alpha \rho D_\omega \nabla \omega) &= \alpha \rho \gamma \min \left(\frac{G}{\nu_t}, \frac{c_1}{a_1} \beta^* \omega \max \left(a_1 \omega, b_1 F_2 \sqrt{S_2} \right) \right) \\ &\quad - \frac{2}{3} \alpha \rho \gamma (\nabla \cdot \mathbf{U}) \omega - \alpha \rho \beta \omega^2 - \alpha \rho (F_1 - 1) \frac{CD_{k\omega}}{\omega} \omega + S_\omega \\ P_k &= \min (G, c_1 \beta^* k \omega), CD_{k\omega} = \frac{2 \alpha_{\omega,2} (\nabla k \cdot \nabla \omega)}{\omega}, G = \nu_t \left(\nabla \mathbf{U} + \nabla \mathbf{U}^T - \frac{2}{3} (\nabla \cdot \mathbf{U}) \mathbf{I} \right) : \nabla \mathbf{U} \\ S_2 &= 2 \left| \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2} \right|^2, F_1 = \tanh \left(\min \left(\min \left(\max \left(\frac{\sqrt{k}}{\beta^* \omega y}, \frac{500 \nu}{\omega y^2} \right), \frac{4 \alpha_{\omega,2} k}{CD_{k\omega} y^2} \right), 10 \right) \right)^4 \right. \\ F_2 &= \tanh \left(\min \left(\max \left(\frac{2 \sqrt{k}}{\beta^* \omega y}, \frac{500 \nu}{\omega y^2} \right), 100 \right) \right)^2, F_3 = 1 - \tanh \left(\min \left(\frac{150 \nu}{\omega y^2}, 10 \right) \right)^4 \\ D_k &= B(F_1, \alpha_{k1}, \alpha_{k2}) \nu_t + \nu, D_\omega = B(F_1, \alpha_{\omega1}, \alpha_{\omega2}) \nu_t + \nu, \beta = B(F_1, \beta_1, \beta_2) \end{aligned}$$

$$\gamma = B(F_1, \gamma_1, \gamma_2), B(a, b, c) = ab + (1-a)c$$

$$\alpha_{k,1} = 0.85, \alpha_{k,2} = 1, \alpha_{\omega,1} = 0.5, \alpha_{\omega,2} = 0.856, \gamma_1 = \frac{5}{9}, \gamma_2 = 0.44, \beta_1 = 0.075, \beta_2 = 0.0828,$$

$$\beta^* = 0.09, a_1 = 0.31, b_1 = 1, c_1 = 10, \nu_t = \frac{a_1 k}{\max(a_1 \omega, b_1 F_2 \sqrt{S_2})}, \varepsilon = \beta^* k \omega$$

7.1.2 LES 湍流模型

可压缩-多相 Smagorinsky⁴ Smagorinsky 湍流模型是目前应用最广泛的 LES 模型。Smagorinsky 模型中的在大部分的文献中表示为 $\nu_t = (C_k \Delta)^2 \sqrt{2 \mathbf{D} : \mathbf{D}}$ 。很明显, 在高剪切流域 (\mathbf{D} 较大), Smagorinsky 模型会导致比较大的耗散。因此在这些区域, C_k 的值需要降低。尤其在壁面处, 虽然近似与层流, 由于速度梯度较大, Smagorinsky 模型偏向与高估湍流粘度。另一方面, Smagorinsky 假定湍流是均衡的, 因此在近壁面处略有不妥。同时, Smagorinsky 模型在壁面处的 Δ 并不为零 (由于梯度不为零), 这将导致壁面附近湍流粘度并不为零, 并不符合实际。这可以通过 van Driest 修正来实现: 将 Δ 建立与壁面距离的关系式并实现

⁴ Δ 表示截断尺度模型

一种阻尼效果 ($\Delta \rightarrow 0, y \rightarrow 0$)。van Driest 将 Δ 建立在 y^+ 基础之上，因此 van Driest 模型需要在壁面布置较多的网格点来获取精准的 y^+ 。另一方面，由于 Smagorinsky 模型高估湍流粘度，其求解也相对稳定。同时不需要附加任何的传输方程，计算快速。因此其在应用中较为广泛。

$$\nu_t = C_k \Delta \sqrt{k_{sgs}}, \mathbf{D} = \frac{\nabla \mathbf{U} + (\nabla \mathbf{U})^T}{2}, \varepsilon_{sgs} = \frac{C_e k_{sgs}^{3/2}}{\Delta}, C_k = 0.094, C_e = 1.048$$

$$k_{sgs} = \left(\frac{-b + \sqrt{b^2 + 4ac}}{2a} \right)^2, a = \frac{C_e}{\Delta}, b = \frac{2}{3} \text{tr}(\mathbf{D}), c = 2C_k \Delta \left(\left(\mathbf{D} - \frac{1}{3} \text{tr}(\mathbf{D}) \mathbf{I} \right) : \mathbf{D} \right)$$

可压缩-多相 WALE 虽然 Smagorinsky 可以结合 van Driest 函数来对壁面湍流粘度进行修正。但其需要在壁面布置较多的网格点形成边界层，因此对于复杂几何较难实施。WALE 模型将 ν_{sgs} 与旋转率联系起来，因此对于任意的表面，由于旋转率趋向于 0，因此 ν_{sgs} 也趋向于零。这使得 WALE 相对于 Smagorinsky 模型更加易于实施且更加具有稳定性。

$$\nu_t = C_k \Delta \sqrt{k_{sgs}}, \mathbf{S}_d = \frac{\nabla \mathbf{U} \cdot \nabla \mathbf{U} + (\nabla \mathbf{U} \cdot \nabla \mathbf{U})^T}{2} - \frac{1}{3} \text{tr} \left(\frac{\nabla \mathbf{U} \cdot \nabla \mathbf{U} + (\nabla \mathbf{U} \cdot \nabla \mathbf{U})^T}{2} \right) \mathbf{I}, C_e = 1.048$$

$$k_{sgs} = \left(C_w \frac{\Delta}{C_k} \right)^2 \frac{|\mathbf{S}_d|^5}{(|\nabla \mathbf{U} + \nabla \mathbf{U}^T|^5 + |\mathbf{S}_d|^{5/2})^2}, \varepsilon_{sgs} = \frac{C_e k_{sgs}^{3/2}}{\Delta}, C_k = 0.094, C_w = 0.325$$

可压缩-多相 kEqn

$$\frac{\partial \alpha \rho k_{sgs}}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} k_{sgs}) - \nabla \cdot (\alpha \rho D_k \nabla k_{sgs}) = \alpha \rho G - \frac{2}{3} \alpha \rho (\nabla \cdot \mathbf{U}) k_{sgs} - \alpha \rho \varepsilon_{sgs} + S_k$$

$$\nu_t = C_k \Delta \sqrt{k_{sgs}}, G = \nu_t \left(\nabla \mathbf{U} + \nabla \mathbf{U}^T - \frac{2}{3} (\nabla \cdot \mathbf{U}) \mathbf{I} \right) : \nabla \mathbf{U}, \varepsilon_{sgs} = C_e \frac{k_{sgs}^{3/2}}{\Delta}, C_e = 1.048, C_k = 0.094$$

可压缩-多相 dynamicLagrangian⁵

$$\frac{\partial \alpha \rho \jmath_{lm}}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \jmath_{lm}) = \frac{1}{T} (\mathbf{L} : \mathbf{M} - \jmath_{lm})$$

$$\frac{\partial \alpha \rho \jmath_{mm}}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \jmath_{mm}) = \frac{1}{T} (\mathbf{M} : \mathbf{M} - \jmath_{lm})$$

$$\frac{1}{T} = \alpha \rho \frac{1}{\theta \Delta} (\jmath_{lm} \jmath_{mm})^{1/8}, \mathbf{S} = \frac{\nabla \mathbf{U} + (\nabla \mathbf{U})^T}{2} - \frac{1}{3} (\nabla \cdot \mathbf{U}) \mathbf{I}, \mathbf{S}_{fi} = \frac{\nabla \mathbf{U}_{fi} + (\nabla \mathbf{U}_{fi})^T}{2} - \frac{1}{3} (\nabla \cdot \mathbf{U}_{fi}) \mathbf{I},$$

$$\mathbf{L} = (\mathbf{U}^2)_{fi} - (\mathbf{U}_{fi})^2 - \frac{1}{3} \text{tr} ((\mathbf{U}^2)_{fi} - (\mathbf{U}_{fi})^2) \mathbf{I}, \mathbf{M} = 2\Delta^2 ((|\mathbf{S}| \mathbf{S})_{fi} - 4 |\mathbf{S}_{fi}| \mathbf{S}_{fi})$$

$$\nu_t = \frac{\jmath_{lm}}{\jmath_{mm}} \Delta^2 \left| \frac{\nabla \mathbf{U} + (\nabla \mathbf{U})^T}{2} - \frac{1}{3} (\nabla \cdot \mathbf{U}) \mathbf{I} \right|, k_{sgs} = \left(\frac{2\jmath_{lm}}{\jmath_{mm}} \right)^{2/3} C_e^{-2/3} \Delta^2 \left| \frac{\nabla \mathbf{U} + (\nabla \mathbf{U})^T}{2} - \frac{1}{3} (\nabla \cdot \mathbf{U}) \mathbf{I} \right|^2$$

$$C_e = 1.048, \theta = 1.5,$$

⁵ $_{fi}$ 表示滤波函数模型。

可压缩-多相 dynamicKEqn⁶

$$\frac{\partial \alpha \rho k_{sgs}}{\partial t} + \alpha \rho k_{sgs} \nabla \cdot (\mathbf{U}) - \nabla \cdot (\alpha \rho D_k \nabla k_{sgs}) = \alpha \rho G - \frac{2}{3} \alpha \rho \nabla \cdot \mathbf{U} k_{sgs} - \alpha \rho \varepsilon_{sgs} + S_k$$

$$\nu_t = C_k \Delta \sqrt{k_{sgs}}, \mathbf{D} = \frac{\nabla \mathbf{U} + (\nabla \mathbf{U})^T}{2} - \frac{1}{3} (\nabla \cdot \mathbf{U}) \mathbf{I}, G = 2 \nu_t \nabla \mathbf{U} : \mathbf{D}, C'_e = \frac{[(\nu + \nu_t)((|\mathbf{D}|^2)_{fi} - |\mathbf{D}_{fi}|^2)]_{sfi}}{\left(\frac{KK^{1.5}}{2\Delta}\right)_{sfi}}$$

$$\mathbf{LL} = \left[(\mathbf{UU})_{fi} - \mathbf{U}_{fi} \mathbf{U}_{fi} - \frac{1}{3} \text{tr}((\mathbf{UU})_{fi} - \mathbf{U}_{fi} \mathbf{U}_{fi}) \mathbf{I} \right]_{sfi}, \mathbf{MM} = \left(-2 \Delta \sqrt{KK} \mathbf{D}_{fi} \right)_{sfi}$$

$$C'_k = \frac{(0.5 \mathbf{LL} : \mathbf{MM})_{sfi}}{(|\mathbf{MM}|^2)_{sfi}}$$

$$C_k = \frac{1}{2} (|C'_k| + C'_k), KK = \frac{1}{2} ((|\mathbf{U}|^2)_{fi} - |\mathbf{U}_{fi}|^2), C_e = \frac{1}{2} (|C'_e| + C'_e), D_k = \nu_t + \nu, \varepsilon_{sgs} = \frac{C_e k_{sgs}^{3/2}}{\Delta},$$

可压缩-多相 DeardorffDiffStress

$$\frac{\partial \alpha \rho \mathbf{R}}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \mathbf{R}) - \nabla \cdot \left(\left(\nu \mathbf{I} + C_s \frac{k_{sgs}}{\varepsilon} \mathbf{R} \right) \nabla \mathbf{R} \right) + C_m \frac{\alpha \rho \sqrt{k_{sgs}}}{\Delta} \mathbf{R} = \alpha \rho \mathbf{P} + \frac{4}{5} \alpha \rho k \mathbf{D} - \frac{2}{3} \left(1 - \frac{C_m}{C_e} \right) \alpha \rho \varepsilon \mathbf{I}$$

$$\nu_t = C_k \Delta \sqrt{k_{sgs}}, \mathbf{D} = \frac{\nabla \mathbf{U} + (\nabla \mathbf{U})^T}{2}, \mathbf{P} = \mathbf{R} \nabla \mathbf{U} + \mathbf{R} \nabla \mathbf{U}^T, \varepsilon_{sgs} = \frac{C_e k_{sgs}^{3/2}}{\Delta}, k_{sgs} = \frac{1}{2} (\mathbf{R}_{xx} + \mathbf{R}_{yy} + \mathbf{R}_{zz})$$

$$C_k = 0.094, C_m = 4.13, C_e = 1.05, C_s = 0.25$$

7.1.3 RANS-LES 混合模型

可压缩-多相 kOmegaSSTSAS 在 kOmegaSST 模型上添加 ω 方程源项:

$$S_\omega = \alpha \rho \omega \min \left(\max \left(\zeta_2 \kappa S_2 \left(\frac{L}{L_{vk}} \right)^2 - \frac{2C}{\sigma_\phi} k \max \left(\frac{|\nabla \omega|^2}{\omega^2}, \frac{|\nabla k|^2}{k^2} \right), 0 \right), \frac{\omega}{0.1 \Delta t} \right)$$

$$L_{vk} = \max \left(\frac{\kappa \sqrt{S_2}}{|\nabla \cdot (\nabla \mathbf{U})|}, C_s \sqrt{\frac{\kappa \zeta_2}{\beta/\beta^* - \gamma}} \Delta \right), L = \frac{\sqrt{k}}{\beta^{*0.25} \omega}$$

$$C_s = 0.11, \kappa = 0.41, \zeta_2 = 3.51, \sigma_\phi = \frac{2}{3}, C = 2, \Delta = (\Delta x \Delta y \Delta z)^{1/3}$$

可压缩-多相 SpalartAllmarasDES

$$\frac{\partial \alpha \rho \tilde{\nu}}{\partial t} + \nabla \cdot (\alpha \rho \mathbf{U} \tilde{\nu}) - \nabla \cdot (\alpha \rho D_{\tilde{\nu}} \nabla \tilde{\nu}) - \frac{C_{b2}}{\sigma_{\nu_t}} \alpha \rho |\nabla \tilde{\nu}|^2 = C_{b1} \alpha \rho \tilde{S} \tilde{\nu} - \frac{C_{w1} \alpha \rho f_w \tilde{\nu}}{\tilde{d}^2} \tilde{\nu}$$

$$f_w = g \left(\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right)^{\frac{1}{6}}, g = r + C_{w2} (r^6 - r), r = \min \left(\frac{\tilde{\nu}}{\tilde{S} \kappa^2 \tilde{d}^2}, 10 \right), \chi = \frac{\tilde{\nu}}{\nu}, f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}$$

⁶ s_{fi} 指定简单滤波函数模型。

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, S = \sqrt{2} \left| \frac{\nabla \mathbf{U} + (\nabla \mathbf{U})^T}{2} \right|, \Omega = \sqrt{2} \left| \frac{\nabla \mathbf{U} - \nabla \mathbf{U}^T}{2} \right|, \nu_t = \tilde{\nu} f_{v1}$$

$$\tilde{S} = \max \left(\Omega + \frac{f_{v2}\tilde{\nu}}{\kappa^2 \tilde{d}^2}, C_s \Omega \right), \tilde{d} = \min (C_{DES} \Delta, y), k_{sgs} = \left(\frac{\nu_t}{C_k \tilde{d}} \right)^2, \sigma_{\nu_t} = 0.66666, \kappa = 0.41, C_e = 1.048$$

$$C_{b1} = 0.1355, C_{b2} = 0.622, C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{1 + C_{b2}}{\sigma_{\nu_t}}, C_{w2} = 0.3, C_{w3} = 2$$

$$C_{v1} = 7.1, C_s = 0.3, C_{DES} = 0.65, C_k = 0.07$$

可压缩-多相 SpalartAllmarasDDES: 在 SpalartAllmarasDES 模型的基础上，更改

$$r = \min \left(\frac{\nu + \nu_t}{|\nabla \mathbf{U}| \kappa^2 \tilde{d}^2}, 10 \right), \tilde{d} = \max (y - (1 - \tanh ((8r)^3)) \max (y - C_{DES} \Delta, 0), 0)$$

可压缩-多相 kOmegaSSTDES: 将 kOmegaSST 模型的 k 方程的源项 $-\alpha \rho \beta^* \omega k$ 更改为

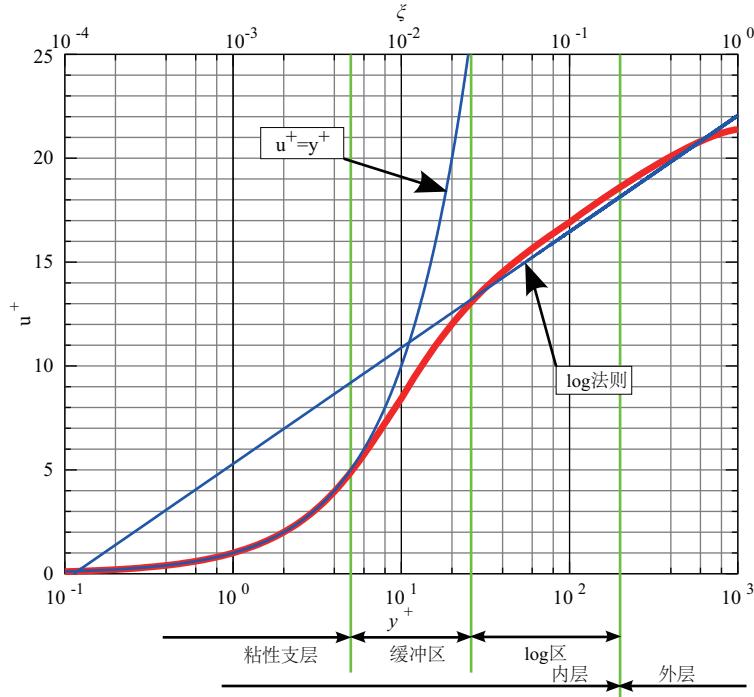
$$-\alpha \rho \beta^* \omega k \rightarrow -\alpha \rho \beta^* \omega k F_{DES}, F_{DES} = \max \left(\frac{L_t (1 - F_{DES})}{C_{DES} \Delta}, 1 \right), L_t = \frac{\sqrt{k}}{\beta^* \omega}, C_{DES} = 0.61$$

α	ρ	ν	ε	k	k_{sgs}	ω	\mathbf{R}	y	Δ
相分数	密度	运动粘度	耗散率	湍动能	亚格子湍动能	湍流频率	雷诺应力	壁面距离	截断尺度

表 7.1: 湍流模型符号表

7.2 OpenFOAM 中的壁面函数

随着工业 CFD 问题的尺度越来越大，虽然计算机的计算能力一直在提高，但对于湍流求解，壁面网格一直不够致密去直接解析。一些湍流模型要求壁面处的网格位于粘性支层内，这导致计算量大大增加。例如，压缩机叶片的研究需要全场的流动结构。如果叶片壁面网格位于粘性支层，会导致计算网格大量增加。壁面函数针对现存问题，可以通过较少的壁面网格结构获得精准的结果。如果应用壁面函数，壁面的网格点可放置于 log 区，大大减少网格数量并且毫不损失精度。例如，对于非均一分布（延伸率 1.15）的未使用壁面函数的壁面网格，对于基于壁面边界层厚度雷诺数为 5000 的的壁面流动，大约需要 40 个边界层网格。若使用壁面函数（第一层网格），只需要约 15 个壁面网格。大大节省了计算量。

图 7.1: 壁面处 $u^+ - y^+$ 分布图。

如图7.1所示，大量的实验表明壁面处附近的无量纲速度 u^+ 与无量纲距离 y^+ 满足一定的型线。在 19 世纪，当时的流体力学研究都是 2 维的，因此 y 方向表示壁面垂直的方向。图中的两个蓝线分别为 u^+ 与 y^+ 的线性分布与 log 分布，红线表示实验检测的普适性 $u^+ - y^+$ 分布。可以看出在粘性支层， $u^+ - y^+$ 符合线性关系：

$$u^+ = y^+ \quad (7.1)$$

在 log 区， $u^+ - y^+$ 符合 log 分布：

$$u^+ = \frac{\ln(Ey^+)}{\kappa} \quad (7.2)$$

其中 E 通常取值 9.8， κ 通常取值 0.4。

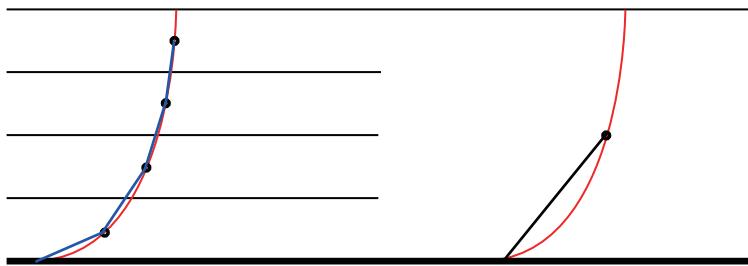


图 7.2: 壁面处速度型线分布图。

依据图7.2所示，红线为真实的速度分布。左侧在壁面网格比较细的情况下，蓝色的实线表示 CFD 计算的速度梯度，可以看出在网格比较密的情况下，分段的速度梯度可以比

较好的满足真实的速度梯度。右侧当网格比较粗的情况下，速度梯度与真实的速度梯度差异较大，因此并不能够合理的预测壁面剪切力。进一步的，预测的阻力系数升力系数必然不准确。壁面函数的意义就是在壁面网格比较稀疏的情况下，通过对粘度进行修正，获得比较精准的壁面剪切力。在调用湍流模型的情况下，壁面剪切力 τ_w 被定义为：

$$\tau_w \approx (\mu_l + \mu_t) \frac{u_p}{y_p} \quad (7.3)$$

其中 μ_l, μ_t 表示层流粘度以及湍流粘度， u_p, y_p 表示第一层网格速度以及第一层的网格壁面距离。在 CFD 代码中， u_p, y_p, μ_l 均为已知， μ_t 可以通过湍流变量计算而来。因此，壁面函数的调用过程，主要是对壁面附近第一层网格的 μ_t 进行修正。

在这里我们考虑第一层网格的 $y^+ > 11.6$ 的情况，也即定义在 log 区域内。首先，壁面附近处的湍流动能 k_p 不需要进行改动，同时 k_p 简单的取零法向梯度边界条件即可。在 k_p 已知的情况下，可以通过下式计算壁面第一层网格的湍流动能耗散率：

$$\varepsilon_p = \frac{C_\mu^{0.75} k_p^{1.5}}{\kappa y_p} \quad (7.4)$$

其中 C_μ 在 kEpsilon 湍流模型中通常取 0.09。同时，壁面附近第一层网格的湍流产生项 G_p 可以表示为

$$G_p = (\mu_l + \mu_t) \frac{u_p}{y_p} \frac{C_\mu^{0.25} k_p^{0.5}}{\kappa y_p} \quad (7.5)$$

随后，将湍流产生项 G_p 代入到湍流动能方程以及湍流动能耗散率方程进行求解获得 k, ε 。最终通过 ν_t 的定义（如 $\nu_t = C_\mu k^2 / \varepsilon$ ）获得湍流粘度场。

7.3 OpenFOAM 中的非结构网格

在第6.2节的讨论中可以看出，结构网格由于存在编号易于实施。然而非结构网格由于存在不同的网格单元类型，且不存在规则的编号体系，在植入的时候要复杂很多。OpenFOAM 中的非结构网格需要定义一系列的必要信息。主要有 points, faces, owner（主单元），neighbour（临单元）以及 boundary（边界）构成。因此，若使用第三方软件生成网格数据，必须存在的就是这五个文件。本节对其进行详述。

7.3.1 point

如图7.3所示，网格单元的 points⁷就是一系列的点数据，点数据由一系列的矢量数据构成。因此对于一套非结构网格，首先需要存储网格的所有的网格单元顶点的信息。OpenFOAM 中的 constant/polyMesh/points 文件包含了所有的点数据，点从上至下进行编号。

⁷ 英文常称之为 vertices

更底层的，OpenFOAM 中的 point 就是一个 vector（矢量）⁸。pointField 就是一个 vector-Field（矢量场）⁹。下面是一个真实的 points 文件示例：

```

1 6072
2 (
3 (0 0 0)
4 (0.009780111723 0 0)
5 (0 0.01 0)
6 (0.009780111723 0.01 0)
7 (0 0 0.00780385)
8 ...
9 (0.009780111723 0 0.007803849986)
10 )

```

其中 6072 表示网格存在 6072 个点。点从上至下进行编号。每一行对应的点的空间坐标。

7.3.2 line 和 edge

除了点之外，OpenFOAM 进一步定义 line，line 存在一个起始点、终止点、中心以及模长。line 的中心点的计算公式为：

$$\text{lineCenter} = 0.5(\text{lineStart} + \text{lineEnd}) \quad (7.6)$$

line 的模长即： $|\text{lineStart} - \text{lineEnd}|$ 。某点 \mathbf{P} 距离线（起始点 \mathbf{A} ，终止点 \mathbf{B} ）上最近的点的计算公式为¹⁰：

$$\mathbf{P} = \mathbf{A} + \left(\frac{(\mathbf{B} - \mathbf{A}) \cdot (\mathbf{P} - \mathbf{A})}{(\mathbf{B} - \mathbf{A}) \cdot (\mathbf{B} - \mathbf{A})} \right) \cdot (\mathbf{B} - \mathbf{A}) \quad (7.7)$$

除 line 外，OpenFOAM 进一步定义 edge，二者的作用类似。需要注意的是，line 与 edge 数据不需要进行读取。

7.3.3 face，面心，面积

在有了点数据之后，需要定义面。如图7.3所示，这些面可以通过将点文件进行连接而形成。OpenFOAM 中的 constant/polyMesh/faces 文件包含了所有的面数据。下面是一个 faces 文件实例：

```

1 5450 // 表示网格存在 5450 个面
2 (
3 4(4 5 7 6) // 每一行前面的数字 4 表示这个面包含 4 个点，后面括号内的数字表示
   这四个点连接成的一个网格面

```

⁸point.H

⁹pointField.H

¹⁰lineI.H

```

4 4(2 6 7 3)
5 4(1 3 7 5)
6 4(6 10 11 7)
7 ...
8 4(5886 5887 5979 5978)
9 4(5197 5978 6070 5243)
10 4(5978 5979 6071 6070)
11 )

```

其中 5450 表示网格存在 5450 个面。每一行前面的数字 4 表示这个面包含 4 个点，后面括号内的数字表示这四个点连接成的一个网格面。面从上至下进行编号。

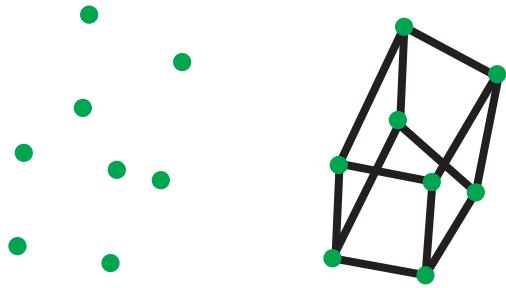


图 7.3: (左) 点数据。(右) 将点连接之后形成的非结构网格单元的面。

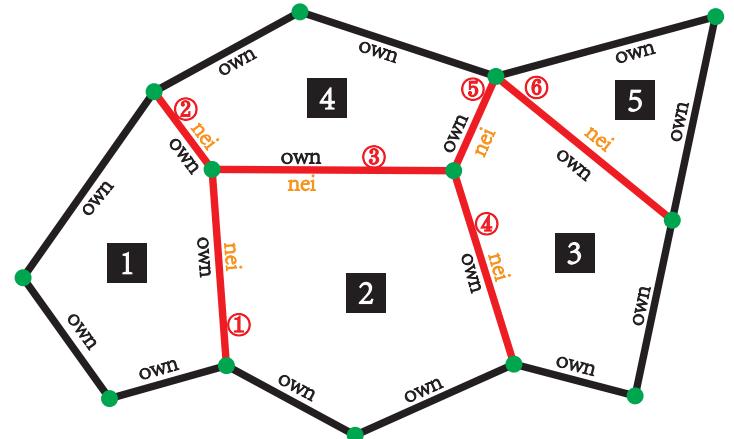


图 7.4: 二维非结构网格的主单元、临单元定义范例。

在有了面数据之后，需要定义主单元以及临单元。如图7.4所示，对于非结构网格的内部面，每个面必然与两个网格单元相连。这两个网格单元分别对应一个主单元，一个临单元。对于非结构网格的边界面，其与内部第一层网格相连，外部为壁面。因此边界面只存在主单元信息。因此，

- 非结构网格的临单元数量，必然等于内部面的数量。对临单元进行遍历，即对所有内部面进行遍历。

- 非结构网格的主单元面数量，必然等于网格内部面数量与边界面的数量之和。

网格面的质心（也被称之为面心）计算很重要¹¹。考虑图7.5中一个最简单的三角形，面的质心的计算非常简单，其计算公式为：

$$\mathbf{C}_f = \frac{1}{3}(\mathbf{p}_0 + \mathbf{p}_1 + \mathbf{p}_2) \quad (7.8)$$

其中 \mathbf{C}_f 表示三角形面的质心，其为一个矢量， $\mathbf{p}_i, i = 0, 1, 2$ 表示三角形的三个顶点。三角形的面矢量可以通过下式计算：

$$\mathbf{S}_f = 0.5(\mathbf{p}_1 - \mathbf{p}_0) \times (\mathbf{p}_2 - \mathbf{p}_0) \quad (7.9)$$

其中 \mathbf{S}_f 表示三角形的面矢量。三角形的面积则为面矢量 \mathbf{S}_f 的模。

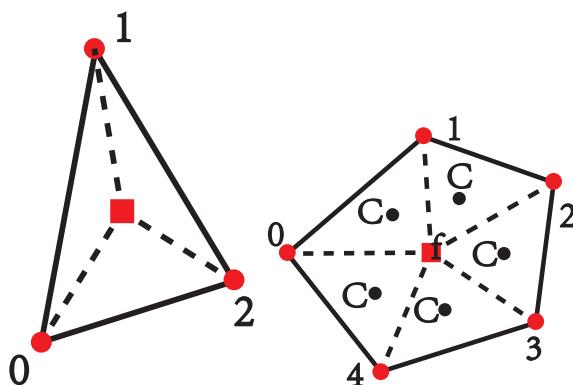


图 7.5: (左) 红色方框就是图中三角形面的质心。(右) 多边形分解。其中红色方框点的位置 f 通过公式(7.10)计算而来。

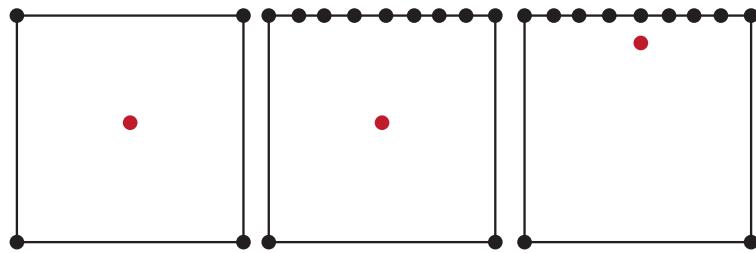


图 7.6: (左) 标准四边形面的质心。(中) 奇葩多边形通过公式(7.10)计算的面的质心。(右) 奇葩多边形通过公式(7.11)计算的面的质心。

对于图7.5中所示任意的多边形，要获得其面的质心，首先需要对图7.5中的多边形进行分解成若干三角形¹²。三角形共有的顶点通过下式进行计算：

$$\mathbf{f} = \frac{\mathbf{p}_0 + \mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_3 + \mathbf{p}_4}{5} \quad (7.10)$$

¹¹Centroid 在这里翻译为质心，更多的讨论请参考[链接](#)

¹²分解成其他形状也可以，但是三角形的质心最容易找。

分母中的 5 是表示多边形顶点的数量。这样，这个多边形就可以分解为 5 个三角形。对每个三角形求其质心可获得 $\mathbf{C}_0, \dots, \mathbf{C}_4$ ，同时可以获得每个三角形的面矢量 $\mathbf{S}_0, \dots, \mathbf{S}_4$ 。那么此多边形的面矢量，即为 $\mathbf{S}_0 + \dots + \mathbf{S}_4$ 。此多边形面的质心即为

$$\mathbf{C}_f = \frac{\sum_{i=1}^5 |\mathbf{S}_i| \mathbf{C}_i}{\sum_{i=1}^5 |\mathbf{S}_i|} \quad (7.11)$$

同时需要注意的是，对于多边形面的质心计算，公式(7.10)也是质心的一种计算方法。这种算法被通常称之为几何平均法 (arithmetic average)。但在 CFD 计算中，几何平均法计算的中心会有很大的误差。例如，对于图7.6中的奇葩多边形，使用公式(7.10)来计算质心可以保证其不受三角形分解的影响，然而使用公式(7.11)来计算则会导致偏移。最后我们总结面心的计算流程：

1. 通过几何平均法方程(7.10)估算一面心点；
2. 通过估算的面心点对面进行分解为三角形；
3. 求解三角形的质心；
4. 通过方程(7.11)计算面心；

7.3.4 虚拟的 cell，体心，体积

在这里首先需要强调的是。OpenFOAM 里面的 cell 并没有明确的对其定义，也可以理解为 OpenFOAM 中的非结构网格并不需要读取 cell 数据。OpenFOAM 将其定义为一系列的面包围的区域，并可以实现一系列的功能。更简化的，OpenFOAM 中的 cell 就可以理解为一系列的面¹³。因此，cell 可以通过 face 转换而来。

例如，对于对重要的 cell 的体积以及体心，可以从 face 的定义中求出。首先考虑几何平均法，虽然其不精准，但计算方法简单。在几何平均法中，网格单元的体心被定义为网格单元的面心加和除以网格单元的面数量¹⁴：

$$\mathbf{C}_c = \frac{\sum \mathbf{C}_{f,i}}{N_i} \quad (7.12)$$

其中 \mathbf{C}_c 表示网格单元体心， $\mathbf{C}_{f,i}$ 表示网格单元的面心， N_i 表示网格单元的面的数量。实施方程(7.12)的时候，一种方法是对网格单元遍历，寻找每个网格单元对应的面，对面心进行加和并除以网格面数。但是这种方法并不是最有效的。OpenFOAM 采用的遍历面的方式更具有有效性。具体的算法如图7.7所示。相应的代码如下：

¹³这在 primitiveMesh.H 中有明晰的体现。网格的基本信息如 point, face, owner, neighbour 都被定义为 Minimum mesh data，然而 cell 则被定义为 cell-faces 的连接性。

¹⁴primitiveMeshCellCentresAndVols.C

```

1 // Clear the fields for accumulation
2 cellCtrs = Zero;
3 cellVols = 0.0;
4
5 const labelList& own = faceOwner(); // 内部面与边界面
6 const labelList& nei = faceNeighbour(); // 边界面
7
8 // first estimate the approximate cell centre as the average of
9 // face centres
10
11 vectorField cEst(nCells(), Zero);
12 labelField nCellFaces(nCells(), 0);
13
14 forAll(own, facei) // 对网格所有的内部面以及边界面进行遍历
15 {
16     cEst[own[facei]] += fCtrs[facei]; // 对某个面对应的 own 网格单元的
17     // 贡献
18     nCellFaces[own[facei]] += 1; // 某网格单元标识为 own 面的数量
19 }
20
21 forAll(nei, facei) // 对网格所有的内部面进行遍历
22 {
23     cEst[nei[facei]] += fCtrs[facei]; // 对某个面对应的 nei 网格单元的
24     // 贡献
25     nCellFaces[nei[facei]] += 1; // 某网格单元标识为 nei 面的数量
26 }
27
28 forAll(cEst, celli)
29 {
30     cEst[celli] /= nCellFaces[celli];
}

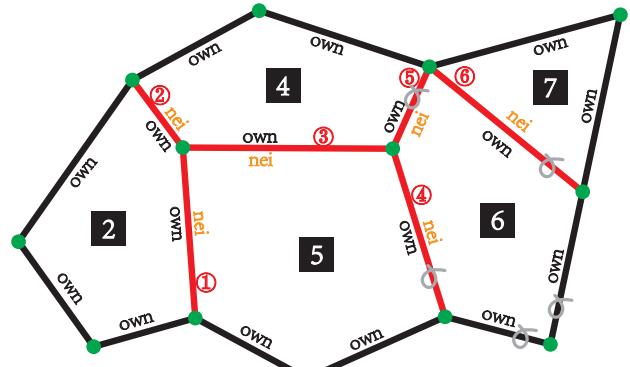
```

上述代码即可以获得几何平均法计算的体心。此方法计算的体心并不精准。因此还需要参考类似面的质心的方法来进行处理。既然有了预估的体心，那么对于这个网格单元，就可以将其分解为多个多面体。每个多面体的基为面，顶点为预估的体心。如图7.8所示，多面体的体心定义为多面体的顶点 B 与基底的面的质心 A 的连线的下方 1/4 处 (D 点)。可以看出 D 点的坐标为 $\mathbf{A}_f + 0.25\mathbf{C}$ ，其中 C 表示 A 点与 B 点之前的连线 $\mathbf{A} - \mathbf{B}$ ，因此 D 点的坐标为

$$\mathbf{D} = 0.75\mathbf{A}_f + 0.25\mathbf{B} \quad (7.13)$$

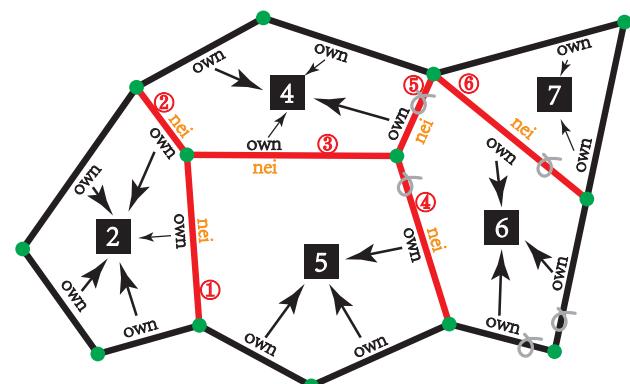
同时，多面体的体积被定义为

$$V = \frac{1}{3} \mathbf{S}_A \cdot (\mathbf{A}_f - \mathbf{B}) \quad (7.14)$$



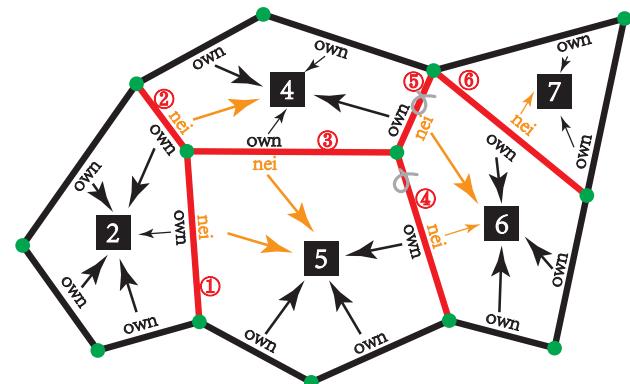
针对左图的2D网格系统，若要计算网格单元体心，首先要将所有的面心加和。例如，计算网格单元3的体心，为这些面的面心相加后除以面的数量。

这些面通过 \circlearrowleft 标记。



在实施的过程中，第一步，对所有的面进行遍历。在遍历的过程中，将某个面对应的own的贡献加和。

对于左图中标记 \circlearrowleft 的5个面，有三个面的贡献添加到了网格单元3。可以看出，此步骤缺失两个面的贡献。



下一步，对左图中的内部面进行遍历。在遍历的过程中，将某个内部面对应的nei的贡献加和。

对于左图中标记 \circlearrowleft 的2个内部面，其贡献添加到了网格单元3。这样，网格单元3的5个面的贡献全部被考虑。

图 7.7: 2D 非结构网格体心计算方法。

最后，有多面体体心以及质心的计算流程：

1. 通过几何平均法方程(7.12)估算多面体体心点；
2. 通过估算的体心点对多面体进行分解为多个多面体；
3. 通过方程(7.13)计算分解后多面体的体心；

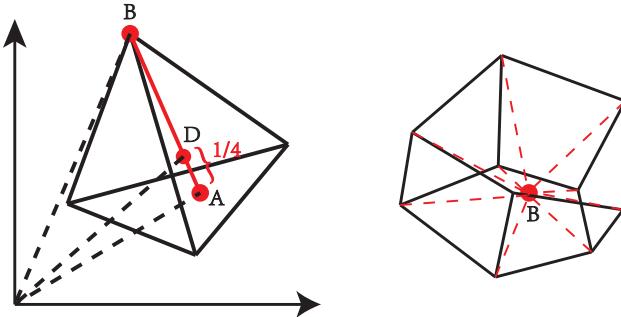


图 7.8: (左) 多边形体心计算。(右) 多面体分解。

4. 通过方程(7.14)计算分解后多面体的体积;
5. 分解前的多面体体积为分解后多面体体心的体积加和:

$$V_c = \sum V_i \quad (7.15)$$

6. 分解前的多面体体心为分解后多面体体心的体积平均:

$$\mathbf{C}_c = \frac{\sum V_i \mathbf{C}_{c,i}}{\sum V_i} \quad (7.16)$$

如图7.9所示，现在我们考虑 cell 与 face 连接性。再次重申，OpenFOAM 中的 cell 并不是我们想象的简单的网格单元标识。其最重要的功能是实现 cell 与 face 的连接性¹⁵。因此，OpenFOAM 中的 cell 标识可以看为一个二维数组 `[]`。数组第一个元素可以认为是 cell 的标识，例如如果有 30 个网格，那么 `cell[0]`, `cell[1]` 就标识第 0 个、第 1 个网格单元。数组的第二个元素（也是一个数组）标识这个 cell 包围的 face，比如 `cell[0][0]`, `cell[0][1]` 标识第 0 个网格里面的第 0 个面以及第 1 个面。相关的代码如下：

```

1   forAll (cellFaceAddr, cellI)//cellFaceAddr可以认为是一个近似的二维
2     数组，首先对数组的第一个元素进行遍历
3   {
4     cellFaceAddr[cellI].setSize(ncf[cellI]);//对数组的第二个元素
5       (数组)设置大小（并不设置值）。比如设置第0个网格单元具有3个值（数组
6       大小为3），第1个网格单元具有4个值（数组大小为4）
7   }
8   ncf = 0;
9
10  forAll (own, faceI)//对网格所有内部面以及边界面进行遍历，此处own为
11    faceOwner()
12  {
13    label cellI = own[faceI];//对某个面，其owner对应一个网格编号，
14    因此这个网格必然与这个face相连

```

¹⁵ primitiveMeshCells.C

```

10         cellFaceAddr[cellI][ncf[cellI]++] = faceI;//将这个face的编号赋
11         值给当前网格，也就是二维数组的第二个元素，成为第二个元素（数组）的一个元素
12     }
13     //通过这一次遍历，所有cell的对应的own面都被定义为相连接的face
14
15     forAll (nei, faceI)//对网格所有内部面进行遍历，此处own为
16     faceNeighbour()
17     {
18         label cellI = nei[faceI];//对某个面，其neighbour对应一个网格编
19         号，因此这个网格必然与这个face相连
20
21         if (cellI >= 0)
22         {
23             cellFaceAddr[cellI][ncf[cellI]++] = faceI;//将这个face的编
24             号赋值给当前网格，也就是二维数组的第二个元素，成为第二个元素（数组）
               的一个元素
25         }
26     }
27     //通过这一次遍历，所有cell的对应的nei面都被定义为相连接的face。因
28     此，cell对应的nei面以及own面都已考虑

```

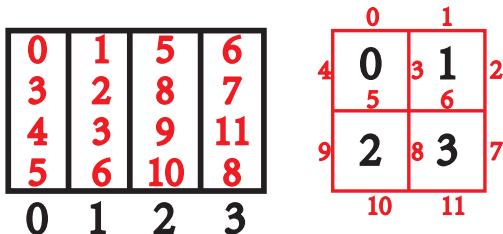


图 7.9: 2D 非结构网格 cell 与 face 的连接性。红色数字表示 face 编号，黑色数字表示 cell 编号。

总结一下，OpenFOAM 的非结构网格必备数据为 point, face, owner, neighbour 以及 boundary。若通过第三方软件生成网格，在进行转换的时候最少提供这 5 个文件。有了这些文件之后，可以计算面心、面矢量等数据。同时不需要显性的定义 cell，就可以获得网格的体心、体积等数据。

7.3.5 偏斜度

偏斜度存在各种不同的定义。如图7.10所示，公开的资料表示偏斜度可以通过一个几何内部的最大角以及最小角来计算。

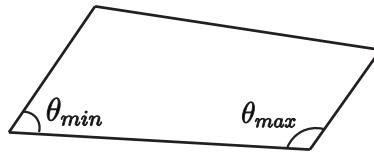


图 7.10: 一种计算偏斜度的方式。

OpenFOAM 并没有采用这种方式。OpenFOAM 中对网格的偏斜度定义在每个网格面上，针对图7.12所示的网格面，首先要求得 D 点的位置。OpenFOAM 中通过下式进行计算：

$$\mathbf{D} = \mathbf{A} \frac{d_{nei}}{d_{nei} + d_{own}} + \mathbf{B} \frac{d_{own}}{d_{nei} + d_{own}} \quad (7.17)$$

其中 d_{nei} 表示 B 点与 C 点的距离， d_{own} 表示 A 点与 C 点的距离。其在图7.12中的局部坐标系下可以更好的理解。D 点位于 A 与 B 点的连线上，其距离 A 点的位置为 $\frac{d_{own}}{d_{nei}+d_{own}}$ 。在获得 D 点的坐标后，最终偏斜度的定义为：

$$\text{skewness} = \frac{|\mathbf{C} - \mathbf{D}|}{|\mathbf{A} - \mathbf{B}|} \quad (7.18)$$

相应的代码如下¹⁶：

```

1   scalar dOwn = mag(faceCtrs[faceI] - cellCtrs[own[faceI]]);
2   scalar dNei = mag(faceCtrs[faceI] - cellCtrs[nei[faceI]]);
3
4   point faceIntersection =
5     cellCtrs[own[faceI]]*dNei/(dOwn+dNei)
6     + cellCtrs[nei[faceI]]*dOwn/(dOwn+dNei);
7
8   scalar skewness =
9     mag(faceCtrs[faceI] - faceIntersection)
10    /(mag(cellCtrs[nei[faceI]] - cellCtrs[own[faceI]]) + VSMALL);

```

图7.11所示的为一个真实的网格，其中的红色的网格单元面的偏斜度很差。偏斜度主要影响的是插值的精度问题。很明显，图7.11中两个点的插值，更倾向于表示红面之上的面的插值，而不是红面上的插值。

7.3.6 非结构网格 TVD 格式

参考图7.13，在结构网格中，TVD 格式通过定义变量 r 来计算通量限制器。在结构网格中， r 定义为

$$r = \frac{\psi_C - \psi_U}{\psi_D - \psi_C} \quad (7.19)$$

¹⁶ primitiveMeshCheck.C

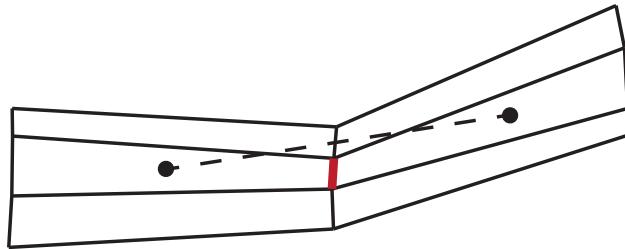


图 7.11: 一个比较差的 skewness 面 (其中的红面)。

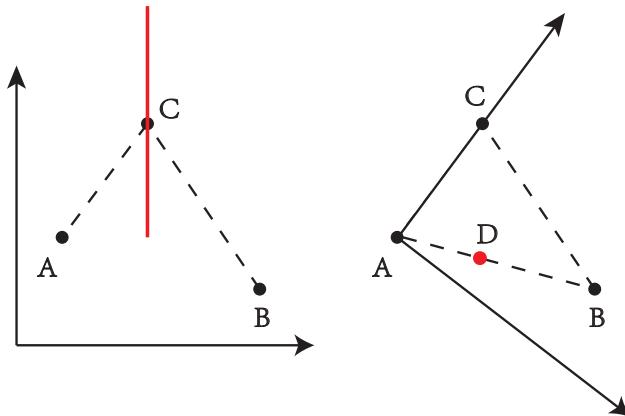


图 7.12: OpenFOAM 中计算网格偏斜度示意图。红线表示网格单元的某个面。A 点表示 own 网格单元体心, B 点表示 nei 网格单元体心, C 点表示面心。(左) 全局坐标系示意图。(右) 局部坐标系示意图, D 点表示通过公式(7.17)求得的矢量点。

r 可以用来衡量网格面处的光滑度。如果在此处是光滑的，则 $r \approx 1$ 。如果此处存在间断，则 r 远远不等于 1。在非结构网格中植入 TVD 格式最大的问题在于：在非结构网格，针对网格单元 C ，并不存在 U 网格点的定义。在 90 年代，OpenFOAM 的前身 foam 就植入了非结构网格的 TVD 格式。相关工作也在 Jasak 的博士论文中被讨论 [46]。在这里简述这种非结构网格 TVD 格式，读者也可参考 2003 年 Darwish 和 Moukalled 的相关工作 [27]。

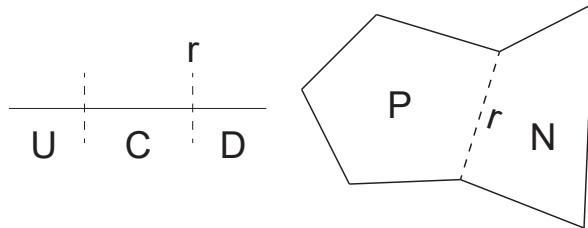


图 7.13: 结构网格（左侧）、非结构网格（右侧）示意图。虚线表示控制体面。其中 r 为 TVD 格式需要定义的变量

方程(7.19)中可以拓展为：

$$r = \frac{\psi_C - \psi_U}{\psi_D - \psi_C} = \frac{(\psi_D - \psi_U) - (\psi_D - \psi_C)}{\psi_D - \psi_C} \quad (7.20)$$

在非结构网格中， $\psi_D - \psi_U$ 被认为是网格 C 点的梯度乘以网格 D 与 U 的方向矢量：

$$\psi_D - \psi_U = \nabla \psi_C \cdot \mathbf{d}_{UD} \quad (7.21)$$

其中 \mathbf{d}_{UD} 表示非结构网格上连接虚拟的 U 点与 D 点的矢量：

$$\mathbf{d}_{UD} = \mathbf{x}_D - \mathbf{x}_U \quad (7.22)$$

一旦 \mathbf{d}_{UD} 可求，那么 $\psi_D - \psi_U$ 可求。目前，Darwish 和 Moukalled [27] 认为这个虚拟节点 U 的位置可以这样给定：

$$\mathbf{d}_{UD} = 2\mathbf{d}_{CD} \quad (7.23)$$

其本质上是假定 C 点在 U 与 D 点的中心，这是一种最简单的二阶假定。这样，有：

$$\psi_D - \psi_U = 2\nabla \psi_C \mathbf{d}_{CD} \quad (7.24)$$

因此，非结构网格中 TVD 格式的 r 定义为：

$$r = \frac{2\nabla \psi_C \mathbf{d}_{CD}}{\psi_D - \psi_C} - 1 \quad (7.25)$$

在图7.13右侧显示的非结构网格中， r 即为：

$$r = \frac{2\nabla \psi_P \mathbf{d}_{PN}}{\psi_N - \psi_P} - 1 \quad (7.26)$$

在获得 r 值之后，有变量 W 在网格面上的值为：

$$\psi_f = \psi_P + \frac{1}{2}\psi(r)(\psi_N - \psi_P) \quad (7.27)$$

其中 $\psi(r)$ 对应不同的高阶格式，如 vanLeer、Minmod 等。

7.4 OpenFOAM 中的有限体积离散

7.4.1 显性对流项 $\nabla \cdot (\mathbf{U}T)$ 、通量 $\sum \mathbf{U}_f \cdot \mathbf{S}_f$ 计算

OpenFOAM 中变量 T 的对流项，若显性离散，可以表示为¹⁷

$$\text{fvc :: div(phi, T)} = \frac{\int_V \int_t \nabla \cdot (TU) dV dt}{\Delta V \Delta t} = \text{surfaceIntegrate}(\mathbf{U}_f \cdot \mathbf{S}_f T_f) \quad (7.28)$$

其中的变量均为已知量（因为为显性离散）。方程(7.28)中使用 `surfaceIntegrate` 来代替 \sum ，这是因为 `surfaceIntegrate` 并不是简单的加和操作¹⁸。

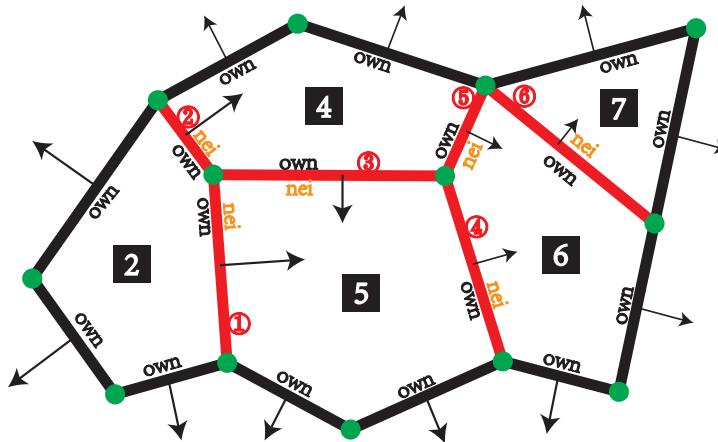


图 7.14: 2D 非结构网格示意图。黑线表示面矢量

下面对 `surfaceIntegrate` 进行解释，考虑整个网格系统的通量计算。一种方式是，对每个网格单元的面进行遍历操作，然后加和，获得当前网格单元的值。但由于每个面链接 2 个网格单元。这种操作实际上会遍历两次面单元。会导致效率损失。在开源 CFD 软件 OpenFOAM 中，仅仅需要遍历一次网格单元，即可求出面通量。具体的，考虑图7.14，针对图中的 2D 网格系统，黑色的箭头表示面矢量，以网格单元 5 为例，如果某个面速度矢量与面矢量同向，则表示流体流入网格 5，反之则表示流出网格 5。在实施的过程中，第一步，对所有的内部面进行遍历，以面 3 为例。假定此面的通量为 A （不管正负），那么 4 单元（own）的值要加上 3 面的通量。同时，5 网格单元将减去 A 贡献。这正是 `surfaceIntegrate`

¹⁷ 注意下方公式除掉了网格单元体积以及时间步长 $\Delta V \Delta t$ 。

¹⁸ OpenFOAM 中的 `surfaceSum` 函数与数学的 \sum 函数作用相同。

函数的意义（依据 own 与 nei 的作用进行正负调节）。随后，对内部面遍历之后，则需要对边界面进行遍历，原理相同。计算通量的代码如下¹⁹：

```

1   const fvMesh& mesh = ssf.mesh();
2
3   const unallocLabelList& owner = mesh.owner(); // 内部面的 own 序号
4   const unallocLabelList& neighbour = mesh.neighbour(); // 内部面的 nei
5   序号
6
7   const Field<Type>& issf = ssf;
8
9   forAll(owner, facei)
10  {
11      ivf[owner[facei]] += issf[facei]; // 针对某个面，own 网格增加贡献
12      ivf[neighbour[facei]] -= issf[facei]; // 针对某个面，nei 网格减少
13      贡献
14  }
15  // 至此，所有内部面都已经考虑，内部面毗连的网格单元通量已经计算。但如果网格存在边界，其贡献还没考虑
16
17  forAll(mesh.boundary(), patchi)
18  {
19      const unallocLabelList& pFaceCells =
20      mesh.boundary()[patchi].faceCells();
21
22      const fvPatchField<Type>& pssf = ssf.boundaryField()[patchi];
23
24      forAll(mesh.boundary()[patchi], facei)
25      {
26          ivf[pFaceCells[facei]] += pssf[facei];
27      }
28  }
29  // 上述代码考虑边界 face 对网格的贡献
30
31  ivf /= mesh.V(); // 除掉网格体积

```

在上面的代码中，注意这里的 own 与之前的 own 的定义并不相同。这里的 own 定义为

```
1   const unallocLabelList& owner = mesh.owner(); // 内部面的 own 序号
```

在 fvMesh.H 中，其表示内部面的 own 网格单元。前述代码的 own 定义为：

```
1   const labelList& own = faceOwner(); // 内部面与边界面 own 序号
```

¹⁹fvcSurfaceIntegrate.C

在 primitiveMesh.H 中，其表示内部面与边界面的网格单元。nei 的定义以此类推。

7.4.2 显性拉普拉斯项 $\nabla \cdot (\nabla T)$ 计算

OpenFOAM 中的显性拉普拉斯项可以定义为²⁰

$$\text{fvc} :: \text{laplacian}(T) = \frac{\int_V \int_t \nabla \cdot (\nabla T) dV dt}{\Delta V \Delta t} = \text{surfaceIntegrate} \left((\nabla T)_f \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|} \right) |\mathbf{S}_f| \quad (7.29)$$

其中 $(\nabla T)_f \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|}$ 当做面法向梯度进行离散即可（参考第7.4.3节）。相关代码如下²¹：

```

1   tmp<GeometricField<Type, fvPatchField, volMesh>> tLaplacian
2   (
3     fvc::div(gamma*this->tsnGradScheme_.snGrad(vf)*mesh.magSF())
4   );
5
6   tLaplacian().rename("laplacian(" + gamma.name() + ',' + vf.name() +
7   ')');
8
9   return tLaplacian;

```

7.4.3 显性面法向梯度项 $(\nabla T)_f \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|}$ 计算

OpenFOAM 中变量 T 的面法向梯度（面场）可以定义为：

$$\text{fvc} :: \text{snGrad}(T) = (\nabla T)_f \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|} \quad (7.30)$$

其中 $(\nabla T)_f$ 表示定义在面上的梯度。方程(7.30)在计算的过程中分为不同的情况，在最简单的情况下（正交网格），其可以表示为：

$$\text{fvc} :: \text{snGrad}(T) = (\nabla T)_f \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|} = \frac{T_{nei} - T_{own}}{|\mathbf{d}|} \quad (7.31)$$

其中的 \mathbf{d} 表示从 own 单元指向 nei 单元的矢量，相对应的代码如下：

```

1   const scalarField& deltaCoeffs = tdeltaCoeffs().internalField(); // 网格单元之间的距离的倒数
2
3   // owner/neighbour addressing
4   const unallocLabelList& owner = mesh.owner(); // 内部面的own标识
5   const unallocLabelList& neighbour = mesh.neighbour(); // 内部面的nei标识

```

²⁰注意下方公式中除掉了网格单元体积与时间步长 $\Delta V \Delta t$ 。

²¹fvcLaplacian.C

```

6
7     forAll(owner, faceI)//对内部面进行遍历
8     {
9         ssf[faceI] =
10            deltaCoeffs[faceI]*(vf[neighbour[faceI]] - vf[owner[faceI
11           ]]);//某个面上的面梯度，在这里假定是正交的
12     }
13
14     forAll(vf.boundaryField(), patchI)
15     {
16         ssf.boundaryField()[patchI] = vf.boundaryField()[patchI].snGrad
17           ()//对于边界面，进一步调用边界面的计算函数（思想相同）
18     }

```

然而在很多情况下，网格并不是正交的。在这里要引入非正交修正。在实现的过程中，首先按照正交网格通过方程(7.31)计算面法向梯度。然后加上非正交修正即可。OpenFOAM 中植入的为超松弛修正，即下面的方程形式：

$$fvc::snGrad(T) = \frac{T_{nei} - T_{own}}{|\mathbf{d}|} + \text{corr}(T) \quad (7.32)$$

其中 $\text{corr}(T)$ 表示非正交修正，可以表示为

$$\text{corr}(T) = \left(\frac{\mathbf{S}_f}{|\mathbf{S}_f|} - \frac{\mathbf{d}}{|\mathbf{d}|} \right) \cdot (\nabla T)_f \quad (7.33)$$

相应的代码如下：

```

1   for (direction cmpt = 0; cmpt < pTraits<Type>::nComponents; cmpt++)
2   {
3       ssf.replace
4       (
5           cmpt,
6           mesh.correctionVectors()
7           & linear
8           <
9               typename
10              outerProduct<vector, typename pTraits<Type>::cmptType
11            >::type
12            >(mesh).interpolate
13            (
14                gradScheme<typename pTraits<Type>::cmptType>::New
15                (
16                    mesh,
17                    mesh.gradScheme(ssf.name())

```

```

17     )()
18     //gaussGrad<typename pTraits<Type>::cmptType>(mesh)
19     .grad(vf.component(cmpt))
20   )
21   );
22 }
23 ...
24 forAll(owner, faceI)
25 {
26   vector unitArea = areas[faceI]/magAreas[faceI];
27   vector delta =
28     cellCentres[neighbour[faceI]] - cellCentres[owner[faceI]];
29
30   CorrVecs[faceI] = unitArea - delta*DeltaCoeffs[faceI];
31 }
```

其中 $(\nabla T)_f$ 仅仅表示 ∇T 向网格面上的插值，将在梯度格式中介绍具体的求解方法。

7.4.4 显性梯度项 ∇T 计算

OpenFOAM 中变量 T 的梯度项可以表示为

$$\text{fvc} :: \text{grad}(T) = \frac{\int_V \int_t \nabla T dV dt}{\Delta V \Delta t} = \frac{1}{\Delta V} \sum T_f \cdot \mathbf{S}_f \quad (7.34)$$

其中 T_f 为用户指定的插值格式。相应的代码如下²²:

```

1  forAll(owner, facei)
2  {
3    GradType Sfssf = Sf[facei]*issf[facei];
4
5    igGrad[owner[facei]] += Sfssf;
6    igGrad[neighbour[facei]] -= Sfssf;
7  }
8  ...
9  igGrad /= mesh.V();
```

OpenFOAM 中梯度项还可以使用最小二乘法计算，在这里不做详细介绍。

7.4.5 显性时间项计算

OpenFOAM 中变量 T 的时间项可以表示为

$$\text{fvc} :: \text{ddt}(T) = \frac{\int_V \int_t \frac{\partial T}{\partial t} dV dt}{\Delta V \Delta t} \quad (7.35)$$

²²gaussGrad.C

在这里要区分具体的时间格式，若采用 Euler 格式，则其可以进一步的写为：

$$\text{fvc} :: \text{ddt}(T) = \frac{\int_V \int_t \frac{\partial T}{\partial t} dV dt}{\Delta V \Delta t} = \frac{T^t - T^{t-\Delta t}}{\Delta t} \quad (7.36)$$

相应的代码如下²³：

```

1 template<class Type>
2 tmp<GeometricField<Type, fvPatchField, volMesh>>
3 EulerDdtScheme<Type>::fvcDdt
4 (
5     const GeometricField<Type, fvPatchField, volMesh>& vf
6 )
7 {
8     ...
9     else
10    {
11        return tmp<GeometricField<Type, fvPatchField, volMesh>>
12        (
13            new GeometricField<Type, fvPatchField, volMesh>
14            (
15                ddtI0object,
16                rDeltaT*(vf - vf.oldTime())// 函数定义
17            )
18        );
19    }
20 }
```

7.5 OpenFOAM 中的稀疏线性系统求解器

7.5.1 稀疏线性系统、`lduMatrix`, `fvMatrix` 等

CFD 求解器在离散之后，会形成一个稀疏线性系统，其可以用方程表示为：

$$\mathbf{A}\psi = \mathbf{b} \quad (7.37)$$

其中 **A** 表示有限体积法离散后的矩阵 `fvMatrix`, ψ 表示待求变量 `fvMatrix.psi`, **b** 表示已知量（源项）`fvMatrix.source`。整个方程(7.37)都通过 `fvMatrix` 来进行存储以及相关的求解操作。具体的，**A** 还要分为两部分进行存储，一个是元素位置，一个是元素的值。进一步的，**A** 要还要区分为对角系数`diagPtr_`，上对角系数`upperPtr_` 以及下对角系数 \times `lowerPtr_` 进行存储²⁴。这些元素的位置信息通过 `lowerAddr_` 以及 `upperAddr_` 存储。现在以下面的代码举例说明矩阵构造过程：

²³EulerDdtScheme.C

²⁴lduMatrix.C

```

1 ...
2 #include createMesh.H
3 ...
4 fvScalarMatrix TEqn(fvm::ddt(T));
5 ...

```

1. 求解器首先通过 `createMesh.H` 来构建 `fvMesh` 类型，构建 `fvMesh` 类型之前，要构建 `polyMesh` 类型。在求解器运行时 `fvMesh`、`polyMesh` 会读取网格文件中的 `owner`、`neighbour` 等信息，然后将这些信息在 `polyMesh` 中进行存储。因此，在求解器运行时，一旦创建网格，`owner`、`neighbour` 等信息既可以随时调用；
2. `fvm::ddt(T)` 首先构建一个空的 `fvMatrix`²⁵，并将这个 `fvMatrix` 的对角线系数、以及源项进行赋值；
3. 将上一步 `fvm::ddt(T)` 构建的 `fvMatrix`，赋值给 `TEqn`，因为上面的代码只有时间项，因此两个 `fvMatrix` 是相同的；

上述代码将初始化一个稀疏线性系统，其 **A** 为一个对角阵，**b** 存在值。边界条件对矩阵不产生影响，同时不存在非对角阵系数。下面看一个稍微复杂一点的代码：

```

1 ...
2 #include createMesh.H
3 ...
4 fvScalarMatrix TEqn
(
    fvm::ddt(T)
    + fvm::laplacian(nu, T)
);
5 ...

```

这两个代码段的区别在于，后面的代码段构造的矩阵 **A** 存在非对角元素：

1. `fvm::laplacian(nu, T)` 首先构建一个空的 `fvMatrix`²⁶，并将这个 `fvMatrix` 的 `upper()` 进行赋值。在赋值的过程中，会调用 `lDUMatrix` 中的 `upper()` 函数，`upper()` 函数将值赋予 `lDUMatrix` 中的 `upperPtr_`，即矩阵的上对角元素。需要强调的是，`upperPtr_` 仅仅是一系列值，不具备位置信息；
2. 为了存储非对角元素的位置信息。必要需要调用某些函数。我们发现在对 `upperPtr_` 赋值时，会自动调用 `lDUAddr().lowerAddr().size()` 函数来获取上三角元素的场元素数量。同时在这一步，也确定了非对角线元素的位置信息。具体的：在调用这个函数

²⁵ EulerDdtScheme.C

²⁶ gaussLaplacianScheme.C

的时候，`lduAddr()` 函数实际调用的是 `lduMesh_.lduAddr()`²⁷。其中的 `lduMesh_` 在创建 `fvMatrix` 以及 `lduMatrix` 的时候，就已经将 `mesh` 赋值给了 `lduMesh` 的具体类型 `fvMesh`。因此，`lduMesh_.lduAddr()` 调用的是 `fvMesh` 中的 `lduAddr()` 函数，这个函数进一步返回 `lduAddressing` 类型，其值通过 `fvMeshLduAddressing` 初始化构造函数来实现²⁸。这个函数在进行初始化的时候，会对 `lowerAddr_` 以及 `upperAddr_` 进行赋值（读取网格的 `owner` 以及 `neighbour` 信息）²⁹。至此，非对角阵的元素位置信息被确立。

3. 更新边界条件信息。
4. 将上一步 `fvm::ddt(T)` 构建的 `fvMatrix`，与拉普拉斯项构建的矩阵，进行加和，赋值给 `TEqn`。此时的矩阵存在对角线系数，非对角线系数，以及源项；

下述即为拉普拉斯项的计算过程。在第7.5.2节有更详细的信息。

```

1 {
2     tmp<surfaceScalarField> tdeltaCoeffs =
3         this->tsnGradScheme_().deltaCoeffs(vf);
4     const surfaceScalarField& deltaCoeffs = tdeltaCoeffs();
5
6     tmp<fvMatrix<Type> > tfvm
7     (
8         new fvMatrix<Type>
9         (
10            vf,
11            deltaCoeffs.dimensions()*gammaMagSf.dimensions()*vf.
12            dimensions()
13        )
14    ); //一个空 fvMatrix
15    fvMatrix<Type>& fvm = tfvm();
16
17    //更新对称的非对角线系数
18    fvm.upper() = deltaCoeffs.internalField()*gammaMagSf.internalField
19    ();
20
21    //更新对角线系数
22    fvm.negSumDiag();
23
24    //更新边界条件对对角线系数的影响
25    forAll(fvm.psi().boundaryField(), patchI)

```

²⁷ lduMatrix.H

²⁸ fvMeshLduAddressing.C

²⁹ fvMeshLduAddressing.H

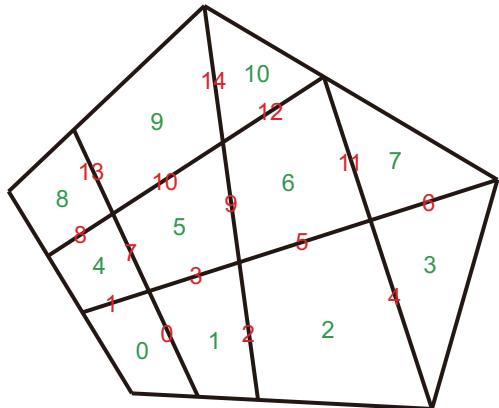
```

23 {
24     const fvPatchField<Type>& psf = fvm.psi().boundaryField()[patchI];
25     const fvPatchScalarField& patchGamma =
26         gammaMagSf.boundaryField()[patchI];
27
28     fvm.internalCoeffs()[patchI] = patchGamma*psf.
29     gradientInternalCoeffs();
30     fvm.boundaryCoeffs()[patchI] = -patchGamma*psf.
31     gradientBoundaryCoeffs();
32 }
33
34     return tfvm;
35 }
```

对流项的离散过程类似，在此不再赘述。需要强调的是，矩阵的非对角阵元素系数的位置，与具体的项有关（例如时间项离散就没有非对角阵元素，扩散项离散则有非对角阵元素）。具体的项的离散后的非对角阵元素位置，由网格直接确定。图7.15则解释了如何通过网格来确定矩阵非对角元素的位置信息。

同时，还可以调用下面的语句来获取相应的信息：

- TEqn.D(): 矩阵的对角线系数。注意，矩阵在离散的过程中，边界条件也会对对角线系数产生影响。因此，矩阵的对角线系数在组建的过程中，首先对内部场组建对角线系数 TEqn.diag()，然后将边界离散产生对对角线系数的影响通过 internalCoeffs_ 进行赋值。TEqn.D() 的值为 diag() 和 internalCoeffs_ 的值加和；
- TEqn.source(): 未考虑边界条件离散的矩阵源项；
- TEqn.internalCoeffs_: 边界条件对矩阵对角线系数的影响。参考第7.5.2节；
- TEqn.boundaryCoeffs_: 边界条件对矩阵源项的影响。**b** 为 source() 与 boundaryCoeffs_ 的和。参考第7.5.2节；
- TEqn.A(): TEqn.D() 除以网格单元体积；
- TEqn.H()、TEqn.H1(): 参考第7.5.9节；
- TEqn.upper(): 矩阵的上对角线系数；
- TEqn.lower(): 矩阵的下对角线系数；
- TEqn.upperAddr(): 矩阵的上对角线系数存储位置；
- TEqn.lowerAddr(): 矩阵的下对角线系数存储位置；



	0	1	2	3	4	5	6	7	8	9	10
0	●										
1		●									
2			●								
3				●							
4					●						
5						●					
6							●				
7								●			
8									●		
9										●	
10											●

A的非对角元素的存储需要两部分信息，一部分是存储位置，一部分是存储的值。存储位置通过upperAddr()与lowerAddr()进行（下文简称u()与l()）。u()本质上即为每个内部面的nei网格标识。l()即为每个内部面的own网格标识。左侧2D网格的u()与l()分别为左下角所示。

以u()举例，其表示：

第0个内部面的nei网格编号为1

第1个内部面的nei网格编号为4

第2个内部面的nei网格编号为2

第3个内部面的nei网格编号为5

...

因此，在网格建立之后，即可以获得u()与l()。在有了u()与l()之后，可以构建A的非对角元素。OpenFOAM中按照网格连接性来构建对称矩阵（非对称矩阵存储元素值为0）。例如：

取l()的第一个元素0, u()的第一个元素1, 构成A01

取l()的第二个元素0, u()的第二个元素4, 构成A04

取l()的第三个元素1, u()的第三个元素2, 构成A12

这些值即矩阵A的非零上三角矩阵，需要存储数据。同理，下三角矩阵为一种反向操作：

取u()的第一个元素1, u()的第一个元素0, 构成A10

取u()的第二个元素4, u()的第二个元素0, 构成A40

取u()的第三个元素2, u()的第三个元素1, 构成A21

...

这样，稀疏非对称矩阵需要进行存储的元素位置即可被定义

。

图 7.15: 网格与矩阵元素位置信息的直接关系。

7.5.2 隐性扩散项离散

首先定义一个单网格系统。考虑图7.16中的网格单元 5，如果其为内部网格：

- 网格面 3 的 own 网格单元为 1, nei 网格单元为 5;
- 网格面 7 的 own 网格单元为 5, nei 网格单元为 5;
- 网格面 9 的 own 网格单元为 5, nei 网格单元为 6;
- 网格面 10 的 own 网格单元为 5, nei 网格单元为 9;

如果其存在边界面，那么边界面与网格毗连的网格单元为边界的 own 网格，且面矢量永远指向边界向外。

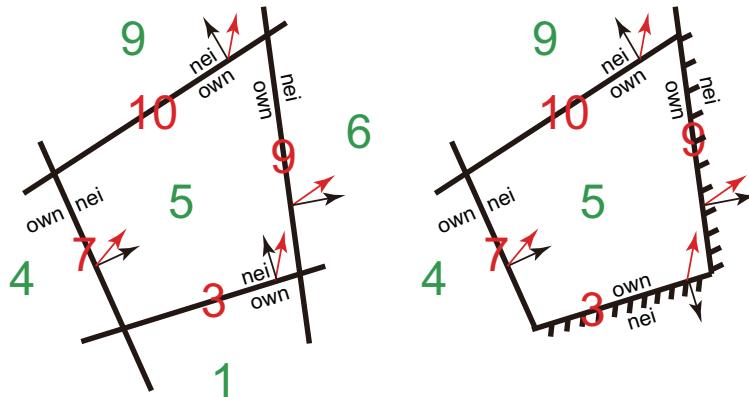


图 7.16: 左侧: 内部网格单元 5 的面连接性。红箭头表示速度矢量, 黑箭头表示面矢量。右侧: 网格单元 5 存在两个边界面 3 与 6。

现在考虑稳态离散后的扩散方程³⁰:

$$\text{surfaceIntegrate} \left(\Gamma_f (\nabla T)_f \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|} |\mathbf{S}_f| \right) = 0 \quad (7.38)$$

现假定网格为正交的, 则方程(7.38)可以进一步写为

$$\text{surfaceIntegrate} \left(\Gamma_f (\nabla T)_f \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|} |\mathbf{S}_f| \right) = \text{surfaceIntegrate} \left(\frac{T_{nei} - T_{own}}{|\mathbf{d}|} \Gamma_f |\mathbf{S}_f| \right) = 0 \quad (7.39)$$

继续考虑图7.16中的网格单元 5, 假定其所有网格面均为内部面, 实施方程(7.39)有³¹:

$$-\frac{T_5 - T_1}{|\mathbf{d}_{51}|} \Gamma_3 |\mathbf{S}_3| - \frac{T_5 - T_4}{|\mathbf{d}_{54}|} \Gamma_7 |\mathbf{S}_7| + \frac{T_6 - T_5}{|\mathbf{d}_{65}|} \Gamma_9 |\mathbf{S}_9| + \frac{T_9 - T_5}{|\mathbf{d}_{95}|} \Gamma_{10} |\mathbf{S}_{10}| = 0 \quad (7.40)$$

³⁰真正的扩散方程需要添加负号。若不添加负号, 离散后的对角线系数为负, 非对角线系数为正。

³¹注意方程(7.40)中前两项添加的负号。这是因为第一项对应的是面 3, 网格单元 5 是面 3 的 nei, 则需要减去相应的贡献。第二项对应的是面 7, 网格单元 5 是面 7 的 nei, 同样需要减去相应的贡献。这也是 surfaceIntegrate 函数不同于 surfaceSum 函数的意义。

整理后为：

$$\underbrace{\frac{\Gamma_3|\mathbf{S}_3|}{|\mathbf{d}_{51}|}T_1 + \frac{\Gamma_7|\mathbf{S}_7|}{|\mathbf{d}_{54}|}T_4 + \frac{\Gamma_9|\mathbf{S}_9|}{|\mathbf{d}_{65}|}T_6 + \frac{\Gamma_{10}|\mathbf{S}_{10}|}{|\mathbf{d}_{95}|}T_9}_{lower} - \underbrace{\left(\frac{\Gamma_3|\mathbf{S}_3|}{|\mathbf{d}_{51}|} + \frac{\Gamma_7|\mathbf{S}_7|}{|\mathbf{d}_{54}|} + \frac{\Gamma_9|\mathbf{S}_9|}{|\mathbf{d}_{65}|} + \frac{\Gamma_{10}|\mathbf{S}_{10}|}{|\mathbf{d}_{95}|} \right)}_{diag} T_5 = 0 \quad (7.41)$$

很明显，若针对所有网格点进行离散，类似的所有网格点的方程(7.41)会构成一个完全对称的矩阵。对角线元素，即为该行所有元素的加和的负值³²。相应的代码如下³³：

```

1   fvm.upper() = deltaCoeffs.internalField()*gammaMagSf.internalField()
2   () ;
3   //更新上三角矩阵，deltaCoeffs.internalField()即为面相邻的两个网格体心向量的模，gammaMagSf.internalField()为面扩散系数与网格面单元的模的乘积，在这里仅仅对upper进行初始化，如果不对lower进行初始化，默认
4   lower=upper
5   fvm.negSumDiag(); //按列进行非对角线元素的加和并取负值

```

现在考虑网格单元 5 存在边界面 3 与 9。在离散的过程中，网格面 7 与 10 按照上述流程正常处理。对于边界面 3，假定给定的为固定值边界条件 $T_f^{ref,3}$ ，在对边界面 3 离散的过程中，其离散形式为：

$$\frac{T_f^{ref,3} - T_5}{|\mathbf{d}|_{53}} \Gamma_3 |\mathbf{S}_3| \quad (7.42)$$

对于网格边界面 9，给定零法向梯度边界条件，其离散形式为：

$$\frac{T_f^9 - T_5}{|\mathbf{d}|_{59}} \Gamma_9 |\mathbf{S}_9| = 0 \quad (7.43)$$

参考方程(7.40)的形式，如果网格单元 5 存在边界面，离散后的方程为：

$$\frac{T_f^{ref,3} - T_5}{|\mathbf{d}|_{53}} \Gamma_3 |\mathbf{S}_3| + \frac{T_f^9 - T_5}{|\mathbf{d}|_{59}} \Gamma_9 |\mathbf{S}_9| - \frac{T_5 - T_4}{|\mathbf{d}|_{65}} \Gamma_7 |\mathbf{S}_7| - \frac{T_5 - T_9}{|\mathbf{d}|_{95}} \Gamma_{10} |\mathbf{S}_{10}| = 0 \quad (7.44)$$

注意其中第二项为 0。可以看出，边界条件可以对边界面毗连的网格单元的矩阵产生影响。对于固定值边界条件，例如方程(7.42)，其影响主要有：

- 离散后的矩阵系数增量为 $\Gamma_d |\mathbf{S}_d| / |\mathbf{d}|_{53}$ ；
- 矩阵源项增量为 $\Gamma_d |\mathbf{S}_d| / |\mathbf{d}|_{53} T_f^{ref,3}$ ；

在代码中，边界条件带来对角线的系数增量用 `internalCoeffs_` 来表示，源项增量用 `boundaryCoeffs_` 来表示。因此，在对拉普拉斯项离散的时候，下列代码用来处理相应的增量：

³²由于为对称阵，也可以取该列所有元素，OpenFOAM 中通过 `negSumDiag()` 函数来实现。

³³gaussLaplacianScheme.C

```

1 // 边界导致的对角线系数增量，一个负值
2 // 由于未考虑负号的拉普拉斯项本身对角线系数为负
3 // 因此考虑固定值边界条件之后，对角线系数的绝对值增加
4 // 注意其中 psf.gradientInternalCoeffs() 为负值
5 fvm.internalCoeffs()[patchI]
6   = patchGamma*psf.gradientInternalCoeffs();
7
8 // 边界导致的源项增量，一个负值
9 // 由于未考虑负号的拉普拉斯项本身无源项
10 // 因此考虑固定值边界条件之后，源项变成一个负值
11 // 注意其中 psf.gradientBoundaryCoeffs() 为正值
12 fvm.boundaryCoeffs()[patchI]
13   = -patchGamma*psf.gradientBoundaryCoeffs();

```

7.5.3 拉普拉斯项非正交修正

在方程(7.39)中，假定网格是正交的。通常情况下，网格非正交的情况比较普遍。因此要考虑非正交修正。非正交修正的思想就是在对拉普拉斯项，假定正交隐性离散的基础上，添加一定量的非正交修正量。如图7.17所示，网格单元面法向矢量为 \mathbf{V} ，其可以分解为正交矢量 \mathbf{V}_{ortho} 以及非正交矢量 $\mathbf{V}_{unortho}$ 之和。用公式表示，即为：

$$(\nabla\phi)_f \cdot \mathbf{V} = (\nabla\phi)_f \cdot \mathbf{V}_{ortho} + (\nabla\phi)_f \cdot \mathbf{V}_{unortho} \quad (7.45)$$

其中的 $(\nabla\phi)_f \cdot \mathbf{V}_{ortho}$ 可隐性离散， $(\nabla\phi)_f \cdot \mathbf{V}_{unortho}$ 可显性离散。在这里存在若干的方法来

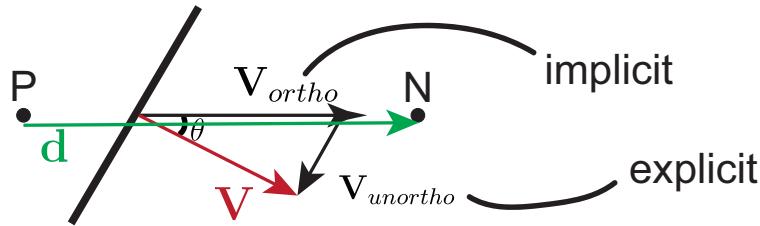


图 7.17: 非正交修正示意图。

处理 \mathbf{V}_{ortho} 以及 $\mathbf{V}_{unortho}$ 的相对大小。需要注意的是，由于方程(7.45)中的 $(\nabla\phi)_f \cdot \mathbf{V}_{ortho}$ 的隐性处理，然而 $(\nabla\phi)_f \cdot \mathbf{V}_{unortho}$ 却为显性处理，其从一个已知量来进行计算，因此存在一个滞后。实操过程中，其需要进行多次迭代来进行。

7.5.4 隐性对流项离散

现在考虑稳态离散后的对流方程：

$$\text{surfaceIntegrate}(\phi_f T_f) = 0 \quad (7.46)$$

其中 $\phi_f = \mathbf{U}_f \cdot \mathbf{S}_f$ 为定义在面上的通量。在考虑隐性离散之前，假定所有网格单元面的通量 ϕ_f 均为 10（注意这里的通量是正号）， T_f 均为 1，也即为已知的，首先我们尝试理解某个网格单元的通量守恒限定性条件的实施过程。在实施的过程中（参考7.28节的介绍），需要对面 3、7、9、10 上的通量进行处理：

- 网格单元 5 为网格单元 3 的 nei，因此要减去面 3 上 $\phi_f T_f$ 的值，即 -10；
- 网格单元 5 为网格单元 7 的 nei，因此要减去面 7 上 $\phi_f T_f$ 的值，即 -10；
- 网格单元 5 为网格单元 9 的 own，因此要加上面 9 上 $\phi_f T_f$ 的值，即 +10；
- 网格单元 5 为网格单元 10 的 own，因此要加上面 10 上 $\phi_f T_f$ 的值，即 +10；

因此， $(-10) + (-10) + (10) + (10) = 0$ ，即表示流入网格单元 5 的 T 以及流出网格单元 5 的 T 是守恒的。

在隐性离散的情况下，面上的 T_f 未知，因此需要从网格单元体心插值而来。即：

$$\begin{aligned} T_{3,f} &= w_3 T_1 + (1 - w_3) T_5 \\ T_{7,f} &= w_7 T_4 + (1 - w_7) T_5 \\ T_{9,f} &= w_9 T_5 + (1 - w_9) T_6 \\ T_{10,f} &= w_{10} T_5 + (1 - w_{10}) T_9 \end{aligned} \quad (7.47)$$

其中 w 表示面上的权重。考虑最简单的情况，比如等距网格中心格式，那么 $w = 1/2$ 。若采用不同的对流项格式， w 具有不同的值。将方程(7.47)代入到方程(7.46)中有：

$$\begin{aligned} & - (w_3 T_1 + (1 - w_3) T_5) \phi_3 - (w_7 T_4 + (1 - w_7) T_5) \phi_7 \\ & + (w_9 T_5 + (1 - w_9) T_6) \phi_9 + (w_{10} T_5 + (1 - w_{10}) T_9) \phi_{10} = 0 \end{aligned} \quad (7.48)$$

整理有：

$$\begin{aligned} & - w_3 \phi_3 T_1 - w_7 \phi_7 T_4 + (1 - w_9) \phi_9 T_6 + (1 - w_{10}) \phi_{10} T_9 \\ & + (-(1 - w_3) \phi_3 - (1 - w_7) \phi_7 + w_9 \phi_9 + w_{10} \phi_{10}) T_5 = 0 \end{aligned} \quad (7.49)$$

参考图7.16，在针对网格单元 5 进行离散的情况下，其网格单元面分别为 3、7、9、10，因此要对图7.18中的网格单元 5 的横行处（浅灰色）红色点 3、7、9、10 进行填充（存在值）。具体的：

- 红点 3 填充的值为 $-w_3 \phi_3$ ；
- 红点 7 填充的值为 $-w_7 \phi_7$ ；
- 红点 9 填充的值为 $(1 - w_9) \phi_9$ ；

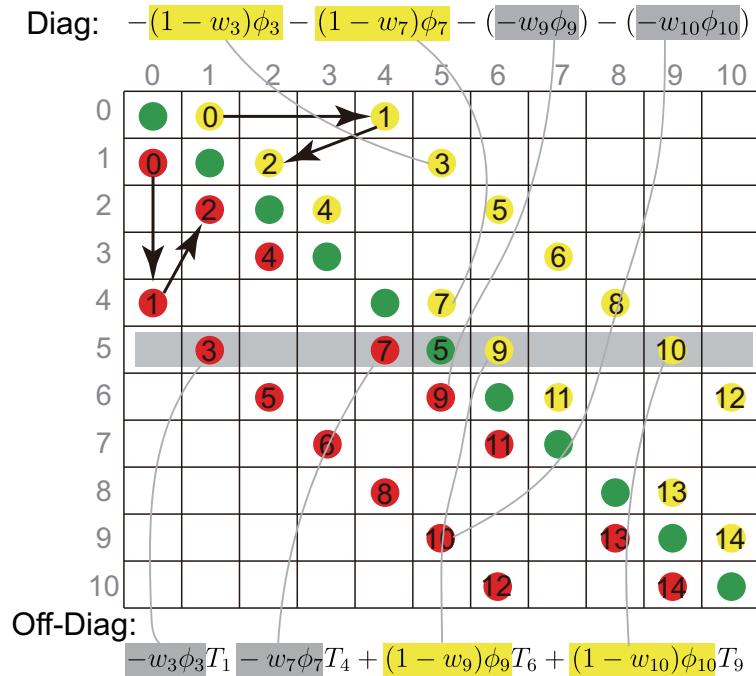


图 7.18: 网格单元 5 离散后方程与矩阵的对应性。红点上的数字为面编号。下三角矩阵从上至下排序。上三角矩阵从上至右排序。底部的方程为方程(7.49)中的非对角线项。顶部的方程为对角线项。

- 红点 10 填充的值为 $(1 - w_{10})\phi_{10}$;

以此类推，在对对流项进行隐形离散的情况下，下三角以及上三角的矩阵可以通过下面的方式进行组建³⁴:

- 在下三角矩阵中，某点处的值均为 $-w_i\phi_i$ (例如下三角矩阵中的红点 5 的值为 $-w_5\phi_5$);
- 在上三角矩阵中，某点处的值均为 $(1 - w_i)\phi_i$ (例如上三角矩阵中的红点 5 的值为 $-(1 - w_5)\phi_5$);

相应的代码如下：

```
1 fvm.lower() = -weights.internalField()*faceFlux.internalField();
2 fvm.upper() = fvm.lower() + faceFlux.internalField();
```

至此，矩阵的非对角元素被组建。

现在看对角元素的组建，从方程(7.49)中可以看出对角线元素的值为：

$$-(1 - w_3)\phi_3 - (1 - w_7)\phi_7 - (-w_9\phi_9) - (-w_{10}\phi_{10}) \quad (7.50)$$

参考图7.18中可以看出，其为网格单元 5 所在列的元素的加和。OpenFOAM 正是通过这种方式来组建隐性对流项离散的对角线系数。相关的代码如下³⁵:

³⁴ gaussConvectionScheme.C

³⁵ gaussConvectionScheme.C, lduMatrixOperations.C

```

1   fvm.negSumDiag();
2   ...
3   for (register label face=0; face<l.size(); face++)
4   {
5       Diag[l[face]] -= Lower[face];
6       Diag[u[face]] -= Upper[face];
7   }

```

参考图7.18中顶部的公式，代码中之所以为负号是因为对角元素为相应的非对角元素的负值的加和。

7.5.5 稀疏线性系统带宽减小算法

给定一个稀疏线性系统，其具有一个特征量叫做带宽。例如对于下面的两个矩阵，他们的带宽可以理解为元素延拓的幅度。很明显，左边的矩阵带宽较大，右边的矩阵带宽较小。

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

对于大型稀疏线性系统，计算人员通常诉求带宽尽可能的小，来进一步的形成条带状稀疏线性系统。同时有资料表示 [5]，带宽减小算法并不能够改变 Krylov 子空间迭代求解器的效率，但可以增加 LU 分解类求解器的效率，但后者在 CFD 中用的较少。对于 Krylov 子空间迭代求解器，减少带宽最重要的作用是做 CPU 的缓存优化。即减少带宽可以增加缓存命中率。这种效率的提升更倾向于硬件效率的改变，而非算法效率类提升。

OpenFOAM 中可以调用不同的算法来减小稀疏线性系统的带宽。比如 Cuthill and McKee 方法 [26]，Sloan 方法 [80] 等。在这里简述 Reverse Cuthill-McKee 方法的实施思想 [11]。对于图7.19中的网格单元，可以依据网格连接性自动填充矩阵元素。例如，对于网格单元 1，其与网格单元 4 相连，因此对于离散出来的 6×6 矩阵，其第一行的第 1、4 个元素存在值。对于网格单元 2，其与网格单元 5、6 相连，因此对于离散出来的 6×6 矩阵，其第二行的第 2、5、6 个元素存在值。以此类推。其最终离散后形成的稀疏线性系统如图7.19所示。

Reverse Cuthill-McKee 方法首先选取其中网格连接性较少的网格单元，如网格单元 1（因为其仅仅与 4 相连接）。选取了网格单元 1 之后，构造一个数列，其中含有 6 个元素，并进行下述操作：

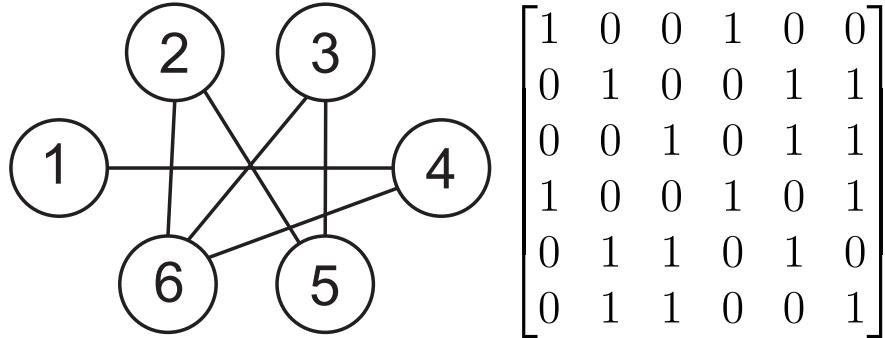


图 7.19: 一个 6 网格单元范例，以及离散后的稀疏线性系统实例。

1. 首先，填充数列第一个元素为 1，数列形成为 $(1, -, -, -, -, -)$ ；
2. 网格单元 1 与网格单元 4 相连，且 4 未填充到数列中，那么数列形成为 $(1, 4, -, -, -, -)$ ；
3. 网格单元 4 与网格单元 1、6 相连，且 6 未填充到数列中，那么数列形成为 $(1, 4, 6, -, -, -)$ ；
4. 网格单元 6 与网格单元 2、3、4 相连，且 2、3 未填充到数列中，取其中的小数 2，那么数列形成为 $(1, 4, 6, 2, -, -)$ ；
5. 网格单元 2 与网格单元 5、6 相连，且 5 未填充到数列中，那么数列形成为 $(1, 4, 6, 2, 5, -)$ ；
6. 网格单元 5 与网格单元 2、3 相连，且 3 未填充到数列中，那么数列形成为 $(1, 4, 6, 2, 5, 3)$ ；
7. 将数列 $(1, 4, 6, 2, 5, 3)$ 转置为 $(3, 5, 2, 6, 4, 1)$ ；

最后，Reverse Cuthill-McKee 方法重置的标号即为 $(3, 5, 2, 6, 4, 1)$ ，参考原数列 $(1, 2, 3, 4, 5, 6)$ ，即原始元素 3 的位置填充为 1，原始元素 5 的位置填充为 2，以此类推。如图7.20所示，很明显，其带宽要少，且呈现一种带状结构。

7.5.6 将网格单元设为固定值

通过对 OpenFOAM 中的矩阵操作，可以将某个网格单元的值设置为固定值。即 OpenFOAM 中矩阵的 `setValues()` 函数要实现的功能。结合图7.16与7.18，现假定要将网格单元 5 的值设置为一个给定的参数。如果形成的矩阵为一个对角阵，那么不需要考虑图7.18中的对角阵，只需要将矩阵的源项设定为矩阵的对角线系数乘以需要设定的值即可。比如网格单元 5 要给定的值为 10，其离散后的矩阵元素值为 3，那么将源项设定为 30 即可。对

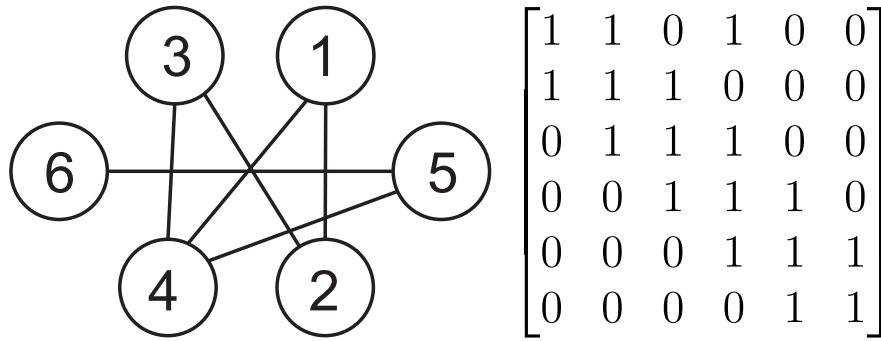


图 7.20: 采用 Reverse Cuthill-McKee 方法调整网格编号后的稀疏线性系统实例。

于非对角矩阵，OpenFOAM 通过一些列的操作可以实现将某个网格单元的值设置为固定值的功能。具体的实现原理可参考图7.21³⁶。

7.5.7 fvMatrix 之 Gauss-Seidel 求解器

在构建整个矩阵系统 $\mathbf{A}\psi = \mathbf{b}$ 之后，需要对其进行求解。OpenFOAM 中提供了各种不同的方法进行求解，本节讨论其中的 smoother 类之 Gauss-Seidel 迭代求解方法。假定矩阵 \mathbf{A} 具有如下形式：

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,0} & a_{n,1} & \cdots & a_{n,n} \end{bmatrix} \quad (7.51)$$

那么 Gauss-Seidel 迭代方法的解 ψ 可以表示为：

$$\psi_i^{(k+1)} = \frac{1}{a_{ii}} b_i - \frac{1}{a_{ii}} \sum_{j=i+1}^n a_{ij} \psi_j^{(k)} - \frac{1}{a_{ii}} \sum_{j=0}^{i-1} a_{ij} \psi_j^{(k+1)}, \quad i = 1, 2, \dots, n \quad (7.52)$$

也可以用文字直观地描述如下：

1. 首先给定初始值 $\psi_0^{(k)}, \dots, \psi_n^{(k)}$ ；
2. 通过 $\psi_1^{(k)}, \dots, \psi_0^{(k)}$ ，来计算 $\psi_0^{(k+1)}$ ；
3. 通过 $\psi_2^{(k+1)}, \psi_1^{(k)}, \dots, \psi_n^{(k)}$ ，来计算 $\psi_1^{(k+1)}$ ；
4. 以此类推...；
5. 通过 $\psi_0^{(k+1)}, \dots, \psi_{n-2}^{(k+1)}, \psi_n^{(k)}$ ，来计算 $\psi_{n-1}^{(k+1)}$ ；
6. 通过 $\psi_0^{(k+1)}, \dots, \psi_{n-1}^{(k+1)}$ ，来计算 $\psi_n^{(k+1)}$ ；

³⁶fvMatrix.C

7. 回到第一步，进行下一次迭代；

如果以网格单元 5 为范例，有：

$$\psi_5^{(k+1)} = \frac{1}{a_{55}}b_5 - \frac{1}{a_{55}}a_{56}\psi_6^{(k)} - \frac{1}{a_{55}}a_{59}\psi_9^{(k)} - \frac{1}{a_{55}}a_{51}\psi_1^{(k+1)} - \frac{1}{a_{55}}a_{54}\psi_4^{(k+1)} \quad (7.53)$$

仔细观察，会发现矩阵的上三角元素，都是乘以已知的值，例如 a_{56} 是上三角元素，其乘以的 $\psi_6^{(k)}$ 为当前迭代步 k 的已知值。 a_{59} 是上三角元素，其乘以的 $\psi_9^{(k)}$ 为当前迭代步 k 的已知值。

更具体地， ψ_0^{k+1} 可以通过下式计算³⁷：

$$\psi_0^{k+1} = \frac{1}{a_{00}} \left(b_0 - \underbrace{a_{01}\psi_1^{(k)} - \dots - a_{0n}\psi_n^{(k)}}_{code1} - \underbrace{0}_{code2} \right) \quad (7.54)$$

其中 a_{01}, \dots, a_{0n} 为上三角元素（对应upperPtr）。有了 ψ_0^{k+1} 后，可以获得 ψ_1^{k+1} ³⁸：

$$\psi_1^{k+1} = \frac{1}{a_{11}} \left(b_1 - \underbrace{a_{02}\psi_2^{(k)} - \dots - a_{0n}\psi_n^{(k)}}_{code1} - \underbrace{a_{10}\psi_0^{(k+1)}}_{code2} \right) \quad (7.55)$$

其中 a_{02}, \dots, a_{0n} 为上三角元素（对应upperPtr）， a_{10} 为下三角元素（对应lowerPtr）。有了 $\psi_0^{k+1}, \psi_1^{k+1}$ 后，可以获得 ψ_2^{k+1} ³⁹：

$$\psi_2^{k+1} = \frac{1}{a_{22}} \left(b_2 - \underbrace{a_{03}\psi_3^{(k)} - \dots - a_{0n}\psi_n^{(k)}}_{code1} - \underbrace{a_{10}\psi_0^{(k+1)} - a_{11}\psi_1^{(k+1)}}_{code2} \right) \quad (7.56)$$

随后以此类推，获得 $k + 1$ 次的迭代解。OpenFOAM 中的代码片段如下

```

1   bPrime = source;
2
3   // code2
4   psii = bPrimePtr[celli];
5
6   // code1
7   for (label facei=fStart; facei<fEnd; facei++)

```

³⁷

$$\psi_0^{k+1} = \frac{1}{a_{00}}b_0 - \frac{1}{a_{00}} \sum_{j=1}^n a_{0j}\psi_j^{(k)}, \quad i = 1, 2, \dots, n$$

³⁸

$$\psi_1^{k+1} = \frac{1}{a_{11}}b_1 - \frac{1}{a_{11}} \sum_{j=2}^n a_{0j}\psi_j^{(k)} - \frac{1}{a_{11}} \sum_{j=0}^1 a_{j0}\psi_j^{(k+1)}, \quad i = 1, 2, \dots, n$$

³⁹

$$\psi_2^{k+1} = \frac{1}{a_{22}}b_2 - \frac{1}{a_{22}} \sum_{j=3}^n a_{0j}\psi_j^{(k)} - \frac{1}{a_{22}} \sum_{j=0}^2 a_{j0}\psi_j^{(k+1)}, \quad i = 1, 2, \dots, n$$

```

8     {
9         psii -= upperPtr[facei]*psiPtr[uPtr[facei]];
10    }
11
12    psii /= diagPtr[celli];
13
14    // code2
15    for (label facei=fStart; facei<fEnd; facei++)
16    {
17        bPrimePtr[uPtr[facei]] -= lowerPtr[facei]*psii;
18    }
19
20    psiPtr[celli] = psii;

```

7.5.8 fvMatrix 的单位

OpenFOAM 中的 fvMatrix 是需要存储单位的。其单位为待求变量的单位乘以体积单位再除以时间单位。现以下面的代码举例：

```

1 fvVectorMatrix UEqn
2 (
3     fvm::ddt()
4 );

```

上述代码通过 fvm::ddt() 来构建 UEqn 的单位⁴⁰。在 fvm::ddt() 函数中，会调用不同的离散格式，若采用欧拉方法，其单位被定义为速度的单位乘以体积单位再除以时间单位⁴¹：

```

1     tmp<fvMatrix<Type> > tfvm
2     (
3         new fvMatrix<Type>
4         (
5             vf,
6             vf.dimensions()*dimVol/dimTime//速度的单位乘以体积单位再除
7             以时间单位
8         )
9     );

```

若考虑下面的代码：

```

1     fvVectorMatrix UEqn
2     (
3         fvm::ddt() + fvm::div(phi,U)

```

⁴⁰fvmDdt.H

⁴¹EulerDdtScheme.C

```
4 );
```

其单位同样是速度的单位乘以体积单位再除以时间单位，因为对流项与时间项的单位必然是一致的。另一方面，在对流项离散的代码中：

```
1 tmp<fvMatrix<Type>> tfvm
2 (
3     new fvMatrix<Type>
4     (
5         vf,
6         faceFlux.dimensions()*vf.dimensions() //速度的单位乘以面积的
7         //单位是faceFlux.dimensions()
8         //再乘以速度的单位
9     )
9 );
```

速度的单位乘以面积的单位再乘以速度的单位，与时间项离散的单位是一致的，均为 $m^4 s^{-2}$ 。

7.5.9 fvMatrix 的 H()、H1() 函数

OpenFOAM 中 fvMatrix 的 H() 函数均涉及到一些矩阵的操作⁴²。fvMatrix 的 H() 函数返回 lduMatrix 的 H() 函数的值，同时加上 fvMatrix 的源项。在这里看 lduMatrix 的 H() 函数。H() 函数与 OpenFOAM 中计算通量的代码非常相似，原理上是将临点的贡献进行加和并取负。例如，若有下述离散方程：

$$a_P \psi_P + \sum a_N \psi_N = S \quad (7.57)$$

lduMatrix 中 H() 操作符的数学表达形式即为：

$$H() = - \sum a_N \psi_N \quad (7.58)$$

fvMatrix 中 H() 操作符的数学表达形式即为：

$$H() = - \sum a_N \psi_N + S \quad (7.59)$$

lduMatrix 中 H1() 操作符用于将矩阵的非对角线元素加和并取负。其数学表达形式即为：

$$H1() = - \sum a_N \quad (7.60)$$

下面即 lduMatrix 中 H() 操作符的代码：

⁴² fvMatrix.C

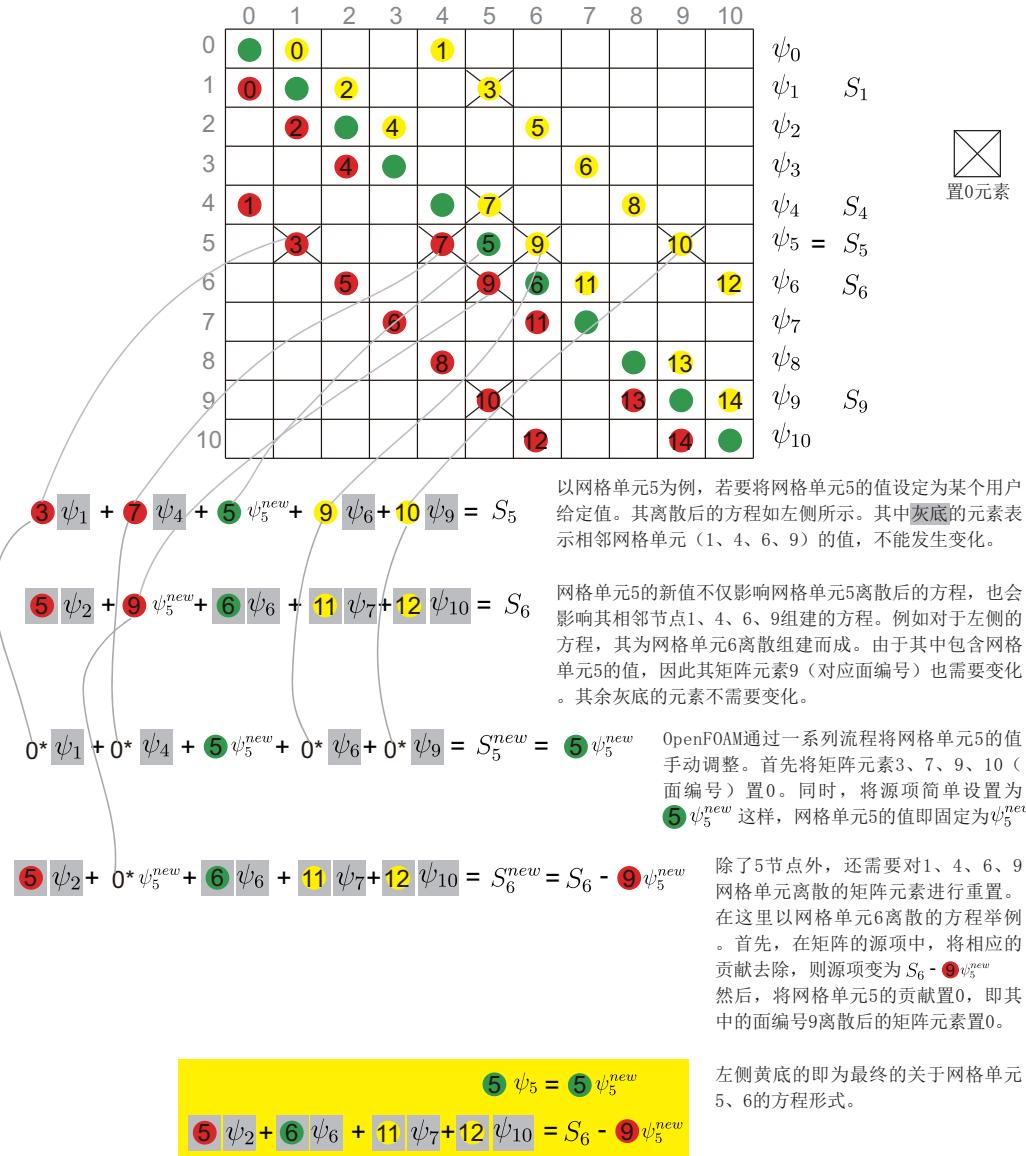


图 7.21: 将网格单元 5 的值设置为固定值的流程。

```

1  if (lowerPtr_ || upperPtr_)
2  {
3      Field<Type> & Hpsi = tHpsi();
4
5      Type* __restrict__ HpsiPtr = Hpsi.begin();
6
7      const Type* __restrict__ psiPtr = psi.begin();
8
9      const label* __restrict__ uPtr = lduAddr().upperAddr().begin();
10     const label* __restrict__ lPtr = lduAddr().lowerAddr().begin();
11
12    const scalar* __restrict__ lowerPtr = lower().begin();
13    const scalar* __restrict__ upperPtr = upper().begin();
14
15    const label nFaces = upper().size();
16
17    for (label face=0; face<nFaces; face++)
18    {
19        HpsiPtr[uPtr[face]] -= lowerPtr[face]*psiPtr[lPtr[face]];
20        HpsiPtr[lPtr[face]] -= upperPtr[face]*psiPtr[uPtr[face]];
21    }
22 }
```

下面即 lduMatrix 中 H1() 操作符的代码，可见其与 H() 函数的代码非常类似。

```

1 if (lowerPtr_ || upperPtr_)
2 {
3     scalarField& H1_ = tH1();
4     ...
5     for (label face=0; face<nFaces; face++)
6     {
7         H1Ptr[uPtr[face]] -= lowerPtr[face];
8         H1Ptr[lPtr[face]] -= upperPtr[face];
9     }
10 }
```

7.5.10 fvMatrix 的“减法”操作符

fvMatrix 对很多操作符进行了重载，本节讨论“减法”操作符。以 fvScalarMatrix 举例，Tbyt 为体标量场 T/t ，单位为 T 的单位除以时间 t 的单位。求解器中可以定义下面的函数：

```

1 fvScalarMatrix TEqn(fvm::ddt(T) - Tbyt);
```

以三网格系统为例, 若考虑全部为零法向梯度边界条件, 上述代码首先通过`fvm::ddt(T)`构建矩阵:

$$\begin{bmatrix} \frac{\Delta V}{\Delta t} & 0 & 0 \\ 0 & \frac{\Delta V}{\Delta t} & 0 \\ 0 & 0 & \frac{\Delta V}{\Delta t} \end{bmatrix} \begin{bmatrix} T_0^{t+\Delta t} \\ T_1^{t+\Delta t} \\ T_2^{t+\Delta t} \end{bmatrix} = \begin{bmatrix} \frac{\Delta V}{\Delta t} T_0^t \\ \frac{\Delta V}{\Delta t} T_1^t \\ \frac{\Delta V}{\Delta t} T_2^t \end{bmatrix} \quad (7.61)$$

随后, `-Tbyt`将其影响添加至方程 Eq. (7.61) 右侧的源项中:

$$\begin{bmatrix} \frac{\Delta V}{\Delta t} & 0 & 0 \\ 0 & \frac{\Delta V}{\Delta t} & 0 \\ 0 & 0 & \frac{\Delta V}{\Delta t} \end{bmatrix} \begin{bmatrix} T_0^{t+\Delta t} \\ T_1^{t+\Delta t} \\ T_2^{t+\Delta t} \end{bmatrix} = \begin{bmatrix} \frac{\Delta V}{\Delta t} T_0^t + \Delta VT/t \\ \frac{\Delta V}{\Delta t} T_1^t + \Delta VT/t \\ \frac{\Delta V}{\Delta t} T_2^t + \Delta VT/t \end{bmatrix} \quad (7.62)$$

因此, 其数学形式即为:

$$\frac{\partial T}{\partial t} - T/t = 0 \quad (7.63)$$

“减法”操作符对应的源代码如下⁴³:

```

1 template<class Type>
2 Foam::tmp<Foam::fvMatrix<Type>> Foam::operator-
3 (
4     const fvMatrix<Type>& A,
5     const dimensioned<Type>& su
6 )
7 {
8     checkMethod(A, su, "-");
9     tmp<fvMatrix<Type>> tC(new fvMatrix<Type>(A));
10    tC().source() += su.value()*tC().psi().mesh().V();
11    return tC;
12 }
```

7.5.11 fvMatrix 的 correction() 函数

OpenFOAM 在处理瞬态可压缩流动的情况下, 经常会调用下面的函数来组建矩阵:

```

1 fvScalarMatrix pDDtEqn
2 (
3     fvc::ddt(rho) + psi*correction(fvm::ddt(p))
4     + fvc::div(phiHbyA) + fvm::div(phid, p)
5 );
```

本节主要讨论其中的 `correction()` 函数。依据 `correction()` 函数的定义:

⁴³ fvMatrix.C

```

1 tmp<Foam::fvMatrix<Type>> tAcorr = tA - (tA() & tA().psi());
2
3 return tAcorr;

```

可以看出计算流程为:1)通过 `fvm:ddt(p)` 来构建一个压力离散的对角矩阵,2)`correction()` 函数将对这个离散矩阵的系数进行修正, 返回一个离散对角矩阵。

以三网格系统为例, `fvm:ddt(p)` 离散后的矩阵为:

$$\begin{bmatrix} \frac{\Delta V}{\Delta t} & 0 & 0 \\ 0 & \frac{\Delta V}{\Delta t} & 0 \\ 0 & 0 & \frac{\Delta V}{\Delta t} \end{bmatrix} \begin{bmatrix} p_0^* \\ p_1^* \\ p_2^* \end{bmatrix} = \begin{bmatrix} \frac{\Delta V}{\Delta t} p_0^t \\ \frac{\Delta V}{\Delta t} p_1^t \\ \frac{\Delta V}{\Delta t} p_2^t \end{bmatrix} \quad (7.64)$$

其中 p^* 表示待求变量, `fvm:ddt(p)` 的 `source` 为方程(7.64)右侧的变量。方程(7.64)即为 `fvm:ddt(p)` 离散后形成的矩阵, 也为传入 `correction()` 函数代码中的 `tA`。

下面细分 `tA() & tA().psi()` 操作。首先, `tA().psi()` 表示下面的源项:

$$\begin{bmatrix} p_0^t \\ p_1^t \\ p_2^t \end{bmatrix} \quad (7.65)$$

`tA().psi()` 表示当前的压力。`tA() & tA().psi()` 是将离散矩阵与 `tA().psi()` 进行一个乘积操作, 其通过在 `fvMatrix.C` 文件中定义一个 `&` 操作符来进行。`tA() & tA().psi()` 分为两步, 第一步的简要代码如下:

```

1 Mphi.primitiveFieldRef().replace(cmpt, -boundaryDiagCmpt*psiCmpt);

```

其将创造一个体标量场, 值为:

$$\begin{bmatrix} -\frac{\Delta V}{\Delta t} p_0^t \\ -\frac{\Delta V}{\Delta t} p_1^t \\ -\frac{\Delta V}{\Delta t} p_2^t \end{bmatrix} \quad (7.66)$$

第二步的简要代码如下:

```

1 Mphi.primitiveFieldRef() += M.lduMatrix::H(psi.field()) + M.source();

```

其在第一步的基础上将, 将第一步创建的体标量场更改为⁴⁴:

$$\begin{bmatrix} -\frac{\Delta V}{\Delta t} p_0^t + \frac{\Delta V}{\Delta t} p_0^t \\ -\frac{\Delta V}{\Delta t} p_1^t + \frac{\Delta V}{\Delta t} p_1^t \\ -\frac{\Delta V}{\Delta t} p_2^t + \frac{\Delta V}{\Delta t} p_2^t \end{bmatrix} \quad (7.67)$$

第三部的的简要代码如下:

⁴⁴`H()` 操作符的元素为空, 因为时间项不构造非对角线系数。

```
1 Mphi.primitiveFieldRef() /= -psi.mesh().V();
```

其将构造的体标量场更改为:

$$\begin{bmatrix} \frac{1}{\Delta t} p_0^t - \frac{1}{\Delta t} p_0^t \\ \frac{1}{\Delta t} p_1^t - \frac{1}{\Delta t} p_1^t \\ \frac{1}{\Delta t} p_2^t - \frac{1}{\Delta t} p_2^t \end{bmatrix} \quad (7.68)$$

然后, 继续看 `tA - (tA() & tA().psi())` 操作。其为一个 `fvMatrix` 与一个体标量场相减的操作, 其中减法操作符可以参考第7.5.10节。其会返回如下矩阵:

$$\begin{bmatrix} \frac{\Delta V}{\Delta t} & 0 & 0 \\ 0 & \frac{\Delta V}{\Delta t} & 0 \\ 0 & 0 & \frac{\Delta V}{\Delta t} \end{bmatrix} \begin{bmatrix} p_0^* \\ p_1^* \\ p_2^* \end{bmatrix} = \begin{bmatrix} \frac{\Delta V}{\Delta t} p_0^t + (\frac{\Delta V}{\Delta t} p_0^t - \frac{\Delta V}{\Delta t} p_0^t) \\ \frac{\Delta V}{\Delta t} p_1^t + (\frac{\Delta V}{\Delta t} p_1^t - \frac{\Delta V}{\Delta t} p_1^t) \\ \frac{\Delta V}{\Delta t} p_2^t + (\frac{\Delta V}{\Delta t} p_2^t - \frac{\Delta V}{\Delta t} p_2^t) \end{bmatrix} = \begin{bmatrix} \frac{\Delta V}{\Delta t} p_0^t \\ \frac{\Delta V}{\Delta t} p_1^t \\ \frac{\Delta V}{\Delta t} p_2^t \end{bmatrix} \quad (7.69)$$

可见, 在当前时间步下, 第一次调用 `correction(fvm::ddt(p))` 与 `fvm::ddt(p)` 是相同的。

然而, 在 PISO 算法中, 需要对压力方程进行多次求解。从第二步开始, `correction(fvm::ddt(p))` 与 `fvm::ddt(p)` 将是有区别的。如果进行了压力求解, 然后继续进行 `correction(fvm::ddt(p))` 的植入, 那么在方程(7.66)中创立的则是

$$\begin{bmatrix} -\frac{\Delta V}{\Delta t} p_0^* \\ -\frac{\Delta V}{\Delta t} p_1^* \\ -\frac{\Delta V}{\Delta t} p_2^* \end{bmatrix} \quad (7.70)$$

方程(7.67)中创立的则变为:

$$\begin{bmatrix} \frac{1}{\Delta t} p_0^* + \frac{1}{\Delta t} p_0^t \\ \frac{1}{\Delta t} p_1^* + \frac{1}{\Delta t} p_1^t \\ \frac{1}{\Delta t} p_2^* + \frac{1}{\Delta t} p_2^t \end{bmatrix} \quad (7.71)$$

最终, `correction(fvm::ddt(p))` 将会返回如下矩阵:

$$\begin{bmatrix} \frac{\Delta V}{\Delta t} & 0 & 0 \\ 0 & \frac{\Delta V}{\Delta t} & 0 \\ 0 & 0 & \frac{\Delta V}{\Delta t} \end{bmatrix} \begin{bmatrix} p_0^{**} \\ p_1^{**} \\ p_2^{**} \end{bmatrix} = \begin{bmatrix} \frac{\Delta V}{\Delta t} p_0^t + (\frac{1}{\Delta t} p_0^* - \frac{1}{\Delta t} p_0^t) \\ \frac{\Delta V}{\Delta t} p_1^t + (\frac{1}{\Delta t} p_1^* - \frac{1}{\Delta t} p_1^t) \\ \frac{\Delta V}{\Delta t} p_2^t + (\frac{1}{\Delta t} p_2^* - \frac{1}{\Delta t} p_2^t) \end{bmatrix} = \begin{bmatrix} \frac{\Delta V}{\Delta t} p_0^* \\ \frac{\Delta V}{\Delta t} p_1^* \\ \frac{\Delta V}{\Delta t} p_2^* \end{bmatrix} \quad (7.72)$$

以此类推, 在同一个时间步下, 如果再次进行了压力方程求解, 则 `correction(fvm::ddt(p))` 变为

$$\begin{bmatrix} \frac{\Delta V}{\Delta t} & 0 & 0 \\ 0 & \frac{\Delta V}{\Delta t} & 0 \\ 0 & 0 & \frac{\Delta V}{\Delta t} \end{bmatrix} \begin{bmatrix} p_0^{***} \\ p_1^{***} \\ p_2^{***} \end{bmatrix} = \begin{bmatrix} \frac{\Delta V}{\Delta t} p_0^{**} \\ \frac{\Delta V}{\Delta t} p_1^{**} \\ \frac{\Delta V}{\Delta t} p_2^{**} \end{bmatrix} \quad (7.73)$$

可见, `correction()` 函数类似于 `fvm::ddt()` 函数, 不同的是 `fvm::ddt()` 在同一个时间步之内, 一直表示的为 $(p^* - p^t)/\Delta t$, 其中 p^* 表示待求变量。若进行一次压力求解, `fvm::ddt()` 变为 $(p^{**} - p^t)/\Delta t$ 。继续进行压力求解, 则变为 $(p^{***} - p^t)/\Delta t$ 。`correction(fvm::ddt())` 则在最初表示的为 $(p^* - p^t)/\Delta t$ 。进行一次压力求解, 变成 $(p^{**} - p^*)/\Delta t$ 。再进行一次压力求解, 变成 $(p^{***} - p^{**})/\Delta t$ 。同时也可以看出, 在最终一个时间步收敛的情况下, p^{***} 趋向于 p^{**} , 因此 `correction(fvm::ddt())` 作用趋向于 0。

7.6 OpenFOAM 中的多重参考系

CFD 计算中经常会涉及到旋转部件。如图7.22所示, 如果存在单一的旋转部件, 可以使用单一参考系 (Single Reference Frame, SRF) 方法。如果计算域内存在旋转结构以及静止结构, 则需要使用多重参考系 (Multiple Reference Frame, MRF) 方法。在定义旋转域的时候, 需要指定旋转点 \mathbf{o} 与旋转轴 \mathbf{a} 。二者均为位置矢量。同时需要给定旋转角速度 Ω 。这样, 旋转域的绕轴方向 Ω 即为:

$$\Omega = \Omega \mathbf{a} \quad (7.74)$$

给定任意的网格单元, 其距离旋转中心点 \mathbf{o} 的距离矢量表示为 \mathbf{r} , 在这种情况下, 绝对速度 \mathbf{U} 为 [9]:

$$\mathbf{U} = \mathbf{U}_r + \Omega \times \mathbf{r} \quad (7.75)$$

其中 \mathbf{U}_r 为相对速度。

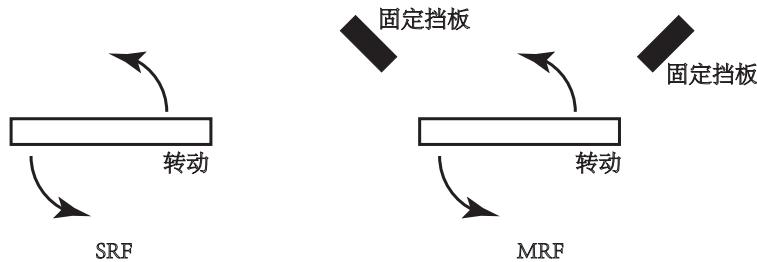


图 7.22: 通过开源 CFD 软件 OpenFOAM 模拟的粘弹性流体两相流。

在 CFD 计算中, 为了调用旋转相关变量, 绝对速度方程需要转变为相对速度方程, 在 SRF 方法中, 相对速度控制方程为:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}_r) &= 0 \\ \frac{\partial \rho \mathbf{U}_r}{\partial t} + \nabla \cdot (\rho \mathbf{U}_r \mathbf{U}_r) + \Omega \times \Omega \times \mathbf{r} + 2\Omega \times \mathbf{U}_r &= \nabla p + \nabla \cdot (\mu \nabla \mathbf{U}_r) \end{aligned} \quad (7.76)$$

其中 $\Omega \times \Omega \times \mathbf{r}$ 表示离心力, $2\Omega \times \mathbf{U}_r$ 表示科氏力。很明显, 在对 SRF 控制方程进行求解的时候, 可以将 \mathbf{U}_r 当做求解变量, 并在方程添加离心力以及科氏力源项即可。

在 MRF 方法中，进一步考虑不可压缩控制方程为：

$$\begin{aligned} \nabla \cdot \mathbf{U}_r &= \nabla \cdot \mathbf{U} = 0 \\ \frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U} \mathbf{U}) - \nabla \cdot ((\boldsymbol{\Omega} \times \mathbf{r}) \mathbf{U}) + \boldsymbol{\Omega} \times \mathbf{U} / \rho &= \frac{\nabla p}{\rho} + \nabla \cdot (\nu \nabla \mathbf{U}) \end{aligned} \quad (7.77)$$

其中连续性方程可用来组建压力泊松方程。动量方程可用于求解绝对速度 \mathbf{U} 。

在 OpenFOAM 中，以 pimpleFoam 为例，速度方程代码如下：

```

1 tmp<fvVectorMatrix> tUEqn
2 (
3     fvm::ddt(U) + fvm::div(phi, U)
4     + MRF.DDt(U)
5     + turbulence->divDevSigma(U)
6     ==
7     fvOptions(U)
8 );

```

其中的 `MRF.DDt(U)` 即为方程(7.77)中的 $\boldsymbol{\Omega} \times \mathbf{U} / \rho$ 。上述代码首先求解下述方程

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U} \mathbf{U}) + \boldsymbol{\Omega} \times \mathbf{U} / \rho = \nabla \cdot (\nu \nabla \mathbf{U}) \quad (7.78)$$

获得预测速度 **HbyA**。然后通过预测速度 **HbyA** 组建通量 `phiHbyA` 以及压力方程⁴⁵。同时，因为在方程(7.78)中并没有考虑 $-\nabla \cdot ((\boldsymbol{\Omega} \times \mathbf{r}) \mathbf{U})$ ，因此在组建 `phiHbyA` 后，需要减去其影响。这在 OpenFOAM 中通过下述代码实现：

```
1 MRF.makeRelative(phiHbyA);
```

7.7 OpenFOAM 中的多孔介质模型

多孔介质可以通过在动量方程中添加一个沉降项来实现。考虑不可压缩流体，有添加多孔介质的动量方程为：

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U} \mathbf{U}) = -\nabla \frac{p}{\rho} + \nabla \cdot (\nu \nabla \mathbf{U}) + \mathbf{S}, \quad (7.79)$$

其中 \mathbf{S} 表示多孔介质源项。其应为一个阻力源项，数值上为一种沉降项，可写为：

$$\mathbf{S} = -\left(\nu \mathbf{D} + \frac{1}{2} \text{tr}(\mathbf{U} \cdot \mathbf{I}) \mathbf{F}\right) \mathbf{U} \quad (7.80)$$

其中 **D**（达西阻力系数）⁴⁶，**F**（Forchheimer 阻力系数）分别表示粘性损失以及惯性损失。各为一个二阶张量。在流体速度比较低的情况下，通常只需要考虑粘性损失，即 $\mathbf{F} = 0$ 。在

⁴⁵更详细的内容请参考 [icoFoam 解析](#)

⁴⁶其分量可能为负，但一定是正定且对称。

这种情况下，如果忽略对流项以及扩散项，在稳态的情况下，有：

$$\nabla p = \mathbf{S} = -\nu \mathbf{D} \mathbf{U} \quad (7.81)$$

若进一步考虑各向同性的粘性损失（ \mathbf{D} 非对角线为 0，且对角线元素相同），则 \mathbf{D} 可以写成一个标量 d 。在这种情况下，方程(7.82)可以简化为达西定律：

$$\nabla p = -\nu d \mathbf{U} \quad (7.82)$$

在将多孔介质源项进行植入的时候需要进行离散，因此就涉及到显性隐性操作。若仅仅考虑时间项，方程(7.79)可以写为：

$$\frac{\partial \mathbf{U}}{\partial t} = -\nu \mathbf{D} \mathbf{U} \quad (7.83)$$

将其通过有限体积法离散，若采用纯隐性方法有⁴⁷：

$$\frac{\mathbf{U}^{t+\Delta t} - \mathbf{U}^t}{\Delta t} \Delta V = -\nu \mathbf{D} \mathbf{U}^{t+\Delta t} \Delta V \quad (7.84)$$

若采用纯显性方法有：

$$\frac{\mathbf{U}^{t+\Delta t} - \mathbf{U}^t}{\Delta t} \Delta V = -\nu \mathbf{D} \mathbf{U}^t \Delta V \quad (7.85)$$

OpenFOAM 里采用一种显性隐性结合的方法。首先对方程(7.83)右侧进行一种数值操作：

$$\frac{\partial \mathbf{U}}{\partial t} = -\nu (\mathbf{D} + \text{tr}(\mathbf{D}) \mathbf{I} - \text{tr}(\mathbf{D}) \mathbf{I}) \mathbf{U} \quad (7.86)$$

分别对右侧进行部分显性离散，部分隐性离散有：

$$\begin{aligned} \frac{\mathbf{U}^{t+\Delta t} - \mathbf{U}^t}{\Delta t} \Delta V &= -\nu (\mathbf{D} - \text{tr}(\mathbf{D}) \mathbf{I}) \mathbf{U}^t \Delta V - \nu (\text{tr}(\mathbf{D}) \mathbf{I}) \mathbf{U}^{t+\Delta t} \Delta V \\ \frac{\mathbf{U}^{t+\Delta t} - \mathbf{U}^t}{\Delta t} \Delta V + \nu (\text{tr}(\mathbf{D}) \mathbf{I}) \mathbf{U}^{t+\Delta t} \Delta V &= -\nu (\mathbf{D} - \text{tr}(\mathbf{D}) \mathbf{I}) \mathbf{U}^t \Delta V \end{aligned} \quad (7.87)$$

方程(7.87)左侧的各向同性部分 $\nu (\text{tr}(\mathbf{D})) \mathbf{U}^{t+\Delta t} \Delta V$ 进入到了离散后的矩阵对角线（类似正应力），右侧的各向异性部分 $-\nu (\mathbf{D} - \text{tr}(\mathbf{D})) \mathbf{U}^t \Delta V$ 进入到了矩阵源项（类似偏应力）⁴⁸。

在考虑各向异性的情况下， \mathbf{D} 为一个二阶张量，既然为各向异性，必然各个方向的达西系数各不相同。因此，在使用的过程中，需要给定流通方向以及各个方向的达西系数。同时，在笛卡尔坐标系下，流动方向可能并不是完全平行于坐标轴（例如一块矩形蜂窝煤放置在一个倾斜的平板上）。因此会涉及到一个坐标转换过程。OpenFOAM 中的渗透率 \mathbf{D} 通过一个矢量 \mathbf{d} 与坐标系统来表示。首先考虑 \mathbf{d} ，如果矢量 \mathbf{d} 某个分量为负（并无物理意义），则需要将其乘以最大的矢量分量的负值，来变成一个正值。例如，如果指定 \mathbf{d} 为：

⁴⁷注意，这种方式 OpenFOAM 的分离式求解器并不支持，因为如果 \mathbf{D} 为非对角阵，这需要使用耦合求解器。

⁴⁸相关代码请参考 DarcyForchheimerTemplates.C。

```
1 d (5000 -1000 -1000);
```

其最大的分量为 5000，则其等于

```
1 d (5000 -1000*5000*(-1) -1000*5000*(-1));
```

即

```
1 d (5000 5000000 5000000);
```

在笛卡尔坐标系下，这表示该多孔介质在 x 方向有一定的渗透率。y、z 方向渗漏率很低。采用 (5000 -1000 -1000) 这种设置只是比较简便且清晰。

同时，在将 d 转换为二阶张量 \mathbf{D} 的时候，需要考虑坐标系统。OpenFOAM 里存在不同的模型，在这里介绍非常基本的 axesRotation 方法。在调用 axesRotation 方法时，需要给定 e_1 以及 e_2 的方向。假设如果考虑流动方向为 x 、 y 平面，那么 e_1 存在 x 、 y 分量， e_2 可以简单设置为 $(0\ 0\ 1)$ 。类似的，如果考虑流动方向为 x 、 z 平面，那么 e_1 存在 x 、 z 分量， e_2 可以简单设置为 $(0\ 1\ 0)$ 。进一步的，其中 e_1 的分量与角度有关。考虑图7.23中的多种渗透方向，其中与 x 方向的夹角为 θ ，那么 e_1 即为 $(\cos \theta, \sin \theta, 0)$ ⁴⁹。在这里附上一种

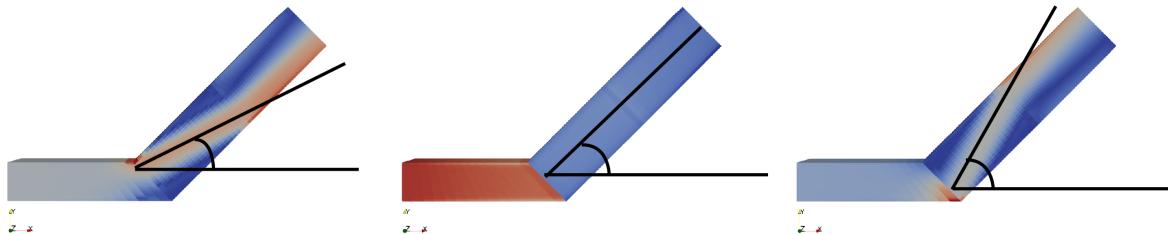


图 7.23: 设置多孔介质不同角度的模拟结果。左中右：30 度、45 度、60 度。

比较方便的设置代码：

```
1 coordinateSystem
2 {
3     type      cartesian;
4     origin   (0 0 0);
5     coordinateRotation
6     {
7         angle    30.0;
8         x       #calc "Foam::cos($angle*0.01745)";
9         y       #calc "Foam::sin($angle*0.01745)";
10        type   axesRotation;
11        e1    ($x $y 0);
12        e2    (0 0 1);
```

⁴⁹ 其中 θ 取弧度值。

13 }

14 }

7.8 OpenFOAM 中的 IATE 模型

Interfacial Area Transport Equation (IATE) 模型为一种可以预测气泡粒径分布的模型。IATE 模型从普适性群体平衡模型出发并对其简化，得到界面浓度传输方程并求解来获得粒径分布。在 IATE 模型中，数量密度函数 $n(V)$ (单位 $1/m^6$) 的控制方程可以写为⁵⁰:

$$\frac{\partial n}{\partial t} + \nabla \cdot (n\mathbf{U}) = S_{bre} + S_{coa} \quad (7.88)$$

其中 S_{bre}, S_{coa} 为聚并破碎源项。定义 $n(V)$ 的 k 阶矩为:

$$m_k = \int V^k n(V) dV \quad (7.89)$$

有 m_0 表示每单位体积的粒子数量 (单位 $1/m^3$)， m_1 表示每单位体积的粒子体积 (无单位)，也即相分数 α 。

对方程(7.88)求一阶矩，有关于 m_1 的体积传输方程:

$$\frac{\partial m_1}{\partial t} + \nabla \cdot (m_1 \mathbf{U}) = \int V S_{bre} dV + \int V S_{coa} dV \quad (7.90)$$

由于体积守恒 (聚并破碎并不会改变总体积)，因此方程右侧积分为 0，即

$$\int V S_{bre} dV + \int V S_{coa} dV = 0 \quad (7.91)$$

在下文中遵循传统写法，用 α 表示一阶矩 m_1 ，因此方程(7.90)可以简单的写为:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{U}) = 0 \quad (7.92)$$

即无传质的相方程。定义 $A(V)$ 为体积 V 粒子的表面积，其和直径以及体积的关系为:

$$A(V) = \pi d^2, V = \frac{\pi d^3}{6} \quad (7.93)$$

进而有 $A(V)$ 和体积 V 的关系:

$$A(V) = 6^{2/3} \pi^{1/3} V^{2/3} \quad (7.94)$$

IATE 模型定义界面浓度 a 为

$$a = \int A(V) n(V) dV \quad (7.95)$$

⁵⁰注意此处的数量密度函数 $n(V)$ 的变量为体积 V 。

依据矩关系有：

$$A(V) = \frac{\int A(V)n(V)dV}{\int n(V)dV} = \frac{a}{m_0} \quad (7.96)$$

$$V = \frac{\int Vn(V)dV}{\int n(V)dV} = \frac{\alpha}{m_0} \quad (7.97)$$

同时，对方程(7.88)左右两边乘以 $A(V)$ 并取矩有：

$$\frac{\partial a}{\partial t} + \nabla \cdot (a\mathbf{U}) = \int A(V)S_{bre}dV + \int A(V)S_{coa}dV \quad (7.98)$$

由于粒子界面浓度并不具有守恒性（同体积情况下粒子数量越多，表面积越大），因此

$$\int A(V)S_{bre}dV + \int A(V)S_{coa}dV \neq 0 \quad (7.99)$$

同时依据方程(7.96)和(7.97)的关系有：

$$A(V) = \frac{a}{m_0} = 6^{2/3}\pi^{1/3} \left(\frac{\alpha}{m_0}\right)^{2/3} \rightarrow m_0 = \psi \frac{a^3}{\alpha^2}, A(V) = \frac{1}{\psi} \left(\frac{\alpha}{a}\right)^2 \quad (7.100)$$

其中

$$\psi = \frac{1}{36\pi} \quad (7.101)$$

依据积分关系：

$$\begin{aligned} \int A(V)S_{bre}dV &\approx \Delta A \int S_{bre}dV, \\ \int A(V)S_{coa}dV &\approx \Delta A \int S_{coa}dV \end{aligned} \quad (7.102)$$

Ishii and Kim 假定 [45]：

$$\Delta A = \frac{1}{3}A, \quad (7.103)$$

这是因为考虑聚并和破碎的情况下，粒子界面变化分别为：

$$\Delta A = -0.413A, \Delta A = 0.26A \quad (7.104)$$

其中 $(|0.413| + |0.26|)/2 \approx 1/3$ 。因此，有：

$$\begin{aligned} \Delta A \int S_{bre}dV &= \frac{1}{3}A \int S_{bre}dV \\ \Delta A \int S_{coa}dV &= \frac{1}{3}A \int S_{coa}dV \end{aligned} \quad (7.105)$$

在 IATE 算法中，通常将方程(7.105)中的积分表示为

$$\int S_{bre}dV = R_{bre}, \int S_{coa}dV = R_{coa} \quad (7.106)$$

将方程(7.103)、(7.106)、(7.101)带入到(7.98)中有：

$$\frac{\partial a}{\partial t} + \nabla \cdot (a \mathbf{U}) = \frac{1}{3} \frac{1}{\psi} \left(\frac{\alpha}{a} \right)^2 (R_{bre} + R_{coa}) \quad (7.107)$$

其中 R_{bre} , B_{coa} 需要进一步的模化。目前 OpenFOAM 中考虑三种机理：湍流引致破碎⁵¹，随机碰撞聚并⁵²以及尾流夹带聚并⁵³。聚并会导致界面浓度的降低，因此为沉降项（负号）。破碎会导致界面浓度的升高，因此为生成项（正号）。

7.9 OpenFOAM 中的常用代码

7.9.1 $\text{laplacian}(\phi) = \text{div}(\text{grad}(\phi))$?

在 OpenFOAM 中，扩散项 $\nabla \cdot (\nabla \phi)$ 的植人形式为

```
1 fvc::laplacian(psi)
```

从数学形式来讲， $\nabla \cdot (\nabla \phi)$ 似乎也可以写为

```
1 fvc::div(fvc::grad(psi))
```

然而，方程的连续形式和离散形式有时是不同的。OpenFOAM 创始人 Henry Weller 也表示：

“The consistency issue arises because laplacian uses a compact molecule and grad(div has to use an extended molecule. The only numerically consistent approach is to use the extended molecule for all three terms but then you loose the advantage of the compact molecule for laplacian.”

下面我们就证明二者的差异。

若采用拉普拉斯项形式对 $\nabla \cdot (\nabla \phi)$ 进行离散有⁵⁴：

$$\int \nabla \cdot (\nabla \phi) dV = \int \nabla \phi dS = \sum (\nabla \phi)_f \mathbf{S}_f = \sum \underbrace{\left((\nabla \phi)_f \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|} \right)}_{\text{snGrad}} \cdot |\mathbf{S}_f| \quad (7.108)$$

其中 snGrad 项可以调用紧致基架点（Compact Stencil）进行离散。

若采用散体梯度的形式对 $\nabla \cdot (\nabla \phi)$ 进行离散有：

$$\int \nabla \cdot (\nabla \phi) dV = \int \nabla \cdot \left(\frac{1}{\Delta V} \sum \phi_f \mathbf{S}_f \right) dV = \sum \left(\frac{1}{\Delta V} \sum \phi_f \mathbf{S}_f \right)_f \cdot \mathbf{S}_f \quad (7.109)$$

⁵¹在 OpenFOAM 中为 turbulentBreakUp。

⁵²在 OpenFOAM 中为 randomCoalescence。

⁵³在 OpenFOAM 中为 wakeEntrainmentCoalescence。

⁵⁴/src/finiteVolume/finiteVolume/laplacianSchemes/gaussLaplacianScheme /gaussLaplacianScheme.C

其中 $\nabla\phi$ 调用 grad (梯度) 离散格式, $\nabla\cdot$ 调用 div (散度) 离散格式, 整体上形成非紧致基架点 (Non-compact Stencil)。

俩种形式的区别在一维形式下更好理解, 考虑 $\frac{\partial}{\partial x} \frac{\partial \phi}{\partial x}$, 采用中心格式拉普拉斯形式的离散为

$$\int \frac{\partial}{\partial x} \frac{\partial \phi}{\partial x} dV = \frac{(\frac{\partial \phi}{\partial x})_w - (\frac{\partial \phi}{\partial x})_e}{\Delta x} \Delta x \Delta y \Delta z = \frac{\phi_W + \phi_E - 2\phi_P}{\Delta x} \Delta y \Delta z \quad (7.110)$$

若采用中心格式散体梯度的形式进行离散有:

$$\begin{aligned} \int \frac{\partial}{\partial x} \frac{\partial \phi}{\partial x} dV &= \int \frac{\partial}{\partial x} \left(\frac{(\frac{\phi_W + \phi_P}{2} - \frac{\phi_P + \phi_E}{2}) \Delta y \Delta z}{\Delta x \Delta y \Delta z} \right) dV = \int \frac{\partial}{\partial x} \left(\frac{\phi_W - \phi_E}{2 \Delta x} \right) dV \\ &= \frac{\phi_{WW} + \phi_{EE} - 2\phi_P}{4 \Delta x} \Delta y \Delta z \end{aligned} \quad (7.111)$$

很明显, 方程(7.110)调用的为相邻网格点, 方程(7.111)调用了更远的网格点, 有可能会引起震荡。

同时, 正如 Henry Weller 所说, 在 N-S 方程中, 压力方程存在压力 p 的拉普拉斯项, 只有其采用拉普拉斯形式进行离散的时候, 才能保证调用紧致基架点防止震荡。

7.9.2 -fvc::flux(-phir,..) = fvc::flux(phir,..)?

OpenFOAM 中可以使用 `flux()` 函数计算通量。举例, 对于 $\phi_f \alpha_f = \mathbf{U}_f \cdot \mathbf{S}_f \alpha_f$, 可通过下述代码计算通量

```
1 phi = fvc::interpolate(U) & mesh.Sf();
2 phiAlpha = fvc::flux(phi, alpha);
```

在 interFoam 中, 存在压面压缩项 $\nabla \cdot \left(\alpha \beta c |\mathbf{U}| \frac{\nabla \alpha}{|\nabla \alpha|} \right)$ 。注意其中 $\beta = 1 - \alpha$ 。现在将其改写为 $\nabla \cdot (\mathbf{U}_r \alpha \beta)$, 其中 \mathbf{U}_r 表示压缩速度。界面压缩项的离散形式为

```
1 fvc::flux(-fvc::flux(-phir, beta, upwind), alpha, upwind)
```

在这里需要讨论的是这些负号, 在调用合适的离散格式的时候, 可以保证有界性。例如, 考虑一维的压缩项, 其可以简写为 $\partial u_r \alpha \beta / \partial x$ 。考虑网格单元面, 按照上述代码, 对这一项中的分子部分进行通量计算可以写为 $u_{r,f} \alpha_{\text{upwind}} \beta_{\text{downwind}}$, 其原因下述, 考虑下段代码:

```
1 -fvc::flux(-phir, beta, upwind)
```

`flux()` 函数内的负号表示对 β 依据负的通量的方向进行迎风离散, 即下风格式。由于 `flux()` 函数内添加的负号不仅改变了离散格式的选取方向, 还改变了通量的正负, 因此在 `flux()` 函数前应该添加负号将其进行还原。这段代码在数学上可以理解为 $-(-u_{r,f} \beta_{\text{downwind}})$ 。然后, α 依据 $-(-u_{r,f} \beta_{\text{downwind}})$ 进行离散。由于 $-(-u_{r,f} \beta_{\text{downwind}})$ 并没有改变方向, 因此 α 迎风离散。

考虑下面三个网格单元，界面压缩项速度的方向源自于 α 的梯度。对于图示的 α 值，相应的速度方向倾向于将中间网格单元的值“挤压”到相邻网格，这也是界面压缩相存在的意义（使得界面更加尖锐）。然而，对于图示的 α 场，如果进行压缩，同时对 α 和 β 调

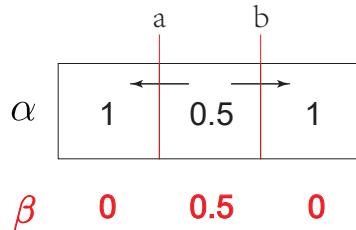


图 7.24: 界面压缩项网格示意图。 a, b 表示网格单元面。

用迎风格式：即 b 网格面的 α, β 均为 0.5，会导致 b 面存在相应的界面压缩通量，导致中心处网格单元的值小于 0.5（合理的），同时周围网格单元的 α 值大于 1。后者会导致越界，并不符合物理。 a 面同理。

相反，如果对 α 调用迎风格式， β 调用下风格式，则 b 网格面的 α, β 分别为 0.5 和 0，进而界面压缩通量为 0，越界行为可以避免。

7.9.3 pEqn.flux() 函数

通常来讲，对于给定变量在网格面上的通量，可以通过速度与变量以及网格面积的乘积来获得。例如考虑变量 T 的对流项 $\nabla \cdot (\mathbf{U}T)$ 对应的对流通量，其可以被定义为 $\mathbf{U}_f \cdot \mathbf{S}_f T_f$ 。考虑扩散通量 $\nabla \cdot (D \nabla T)$ ，其可以被定义为 $D_f (\nabla T)_f \cdot \mathbf{S}_f$ ，进一步可以定义为面法向梯度相关量 $D_f ((\nabla T)_f \cdot \mathbf{S}_f / |\mathbf{S}_f|) |\mathbf{S}_f|$ 。以后者扩散通量举例，其可以通过下面的代码进行计算：

```
1   fvc::snGrad(T)*mesh.magSf()*fvc::interpolate(D);
```

这是传统的通量计算方法。还有另外一种通量计算方法，其更加普适性且连贯。例如，如果对 T 方程组建稀疏线性系统，OpenFOAM 中的 `flux()` 可以直接调用稀疏线性系统的矩阵系数，反过来计算定义在网格面上的通量。这也即 `flux()` 函数的意义。例如对于下述代码

```
1   fvScalarMatrix pEqn
2   (
3       fvm::laplacian(D, p) == fvc::div(phiHbyA)
4   );
```

其为不可压缩单相流的压力泊松方程。对于其中的 `pEqn`，其中的 `fvc` 部分进入源项，不参与 `flux()` 函数的计算。其左侧的拉普拉斯项，即表示压力的扩散项，参考前述 $\nabla \cdot (D \nabla T)$ 扩散通量的含义，有压力方程的扩散通量的数学意义：

$$\text{fvm :: laplacian(rAU, p)} = D_f \left((\nabla p)_f \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|} \right) |\mathbf{S}_f| \quad (7.112)$$

方程(7.112)的右侧数学公式用代码可以表示为:

```
1 fvc::interpolate(D)*fvc::snGrad(p)*mesh.magSf();
```

上述代码获得的通量场与pEqn.flux()是一致的。

需要注意的是，对于时间项以及源项，由于其并不参与网格间的流动，因此flux()函数并不考虑他们的贡献。同时，显性项离散不进入flux()函数的贡献，即使是对流项或者扩散项。

下面再进行举例，如果有如下代码：

```
1 fvScalarMatrix pEqn
2 {
3     fvm::div(phiHbyA, p)
4     - fvm::laplacian(D, p)
5 }
```

pEqn.flux()即为：

$$\text{flux}() = p_f \mathbf{H} \cdot \mathbf{A}_f \cdot \mathbf{S}_f - D_f \left((\nabla p)_f \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|} \right) |\mathbf{S}_f| \quad (7.113)$$

7.9.4 fvc::div(phiHbyA) 函数

fvc::div(phiHbyA)在OpenFOAM中大量的出现，其主要用于计算通量的散度 $\nabla \cdot \mathbf{U}$ 。容易引起困惑的是，div为散度操作符，但是phiHbyA却是一个标量，对一个标量做散度操作这并不符合数学逻辑。接下来我们对这一问题进行解答⁵⁵。

从数学形式来看，散度项 $\nabla \cdot \mathbf{U}$ 在有限体积法下可以离散为

$$\int \nabla \cdot \mathbf{U} dV = \int \mathbf{U}_f \cdot d\mathbf{S}_f = \sum \mathbf{U}_f \cdot \mathbf{S}_f \quad (7.114)$$

OpenFOAM中将 $\sum \mathbf{U}_f \cdot \mathbf{S}_f$ 称之为phiHbyA。其为一个surfaceScalarField面标量场。fvc::div(phiHbyA)离散时，首先调用：

```
1 return GeometricField<Type, fvPatchField, volMesh>::New
2 (
3     "div(" + ssf.name() + ')',
4     fvc::surfaceIntegrate(ssf)
5 );
```

然后进一步调用surfaceIntegrate()函数，其为一个面积分函数。举例，如果进行surfaceIntegrate(phiHbyA)操作，其数学含义即为：

$$\text{fvc :: div(phiHbyA)} = \text{surfaceIntegrate(phiHbyA)} = \sum \text{phiHbyA} = \sum \mathbf{U}_f \cdot \mathbf{S}_f \quad (7.115)$$

⁵⁵ 问题讨论参考[本链接](#)。

因此, `fvc::div(phiHbyA)` 中的 `div` 并不是散度操作, 而是面积分的操作。同时需要注意, 在组建 `phiHbyA` 的时候, 可以通过 `fvc::flux(HbyA)` 函数调用不同的离散格式。

7.9.5 fvc::reconstruct() 函数

`fvc::reconstruct()` 函数主要用于处理动量方程中源项。例如表面张力项、压力项、浮力项、界面动量交换项等。此函数返回的场的变量要比输入的场变量高一阶。例如, 若对标量进行 `reconstruct`, 则得到矢量场, 若对矢量做 `reconstruct`, 则得到二阶张量场。现存一些讨论 `fvc::reconstruct()` 的文献很有意义, 但并不完善 [1, 2]。本小节详述为什么需要 `fvc::reconstruct()` 以及其数学形式。

`fvc::reconstruct()` 实现了将定义在网格单元面上的量重组到体心的过程, 即 `reconstruct`。结合代码, 例如对于速度, 这种重组的数学过程为⁵⁶:

$$\mathbf{U}_P = \frac{\sum \frac{\mathbf{S}_f}{|\mathbf{S}_f|} (\underbrace{\mathbf{S}_f \cdot \mathbf{U}_f}_{input})}{\sum \frac{\mathbf{S}_f}{|\mathbf{S}_f|} \otimes \mathbf{S}_f} \quad (7.116)$$

即:

$$\left(\sum \mathbf{n}_f \otimes \mathbf{S}_f \right) \cdot \mathbf{U}_P = \sum (\mathbf{n}_f \otimes \mathbf{S}_f) \cdot \mathbf{U}_P = \sum \mathbf{n}_f \left(\underbrace{\mathbf{S}_f \cdot \mathbf{U}_f}_{input} \right) \quad (7.117)$$

其中 \mathbf{n}_f 表示网格单元面单位矢量, \mathbf{U}_P 表示网格单元体心的速度, \mathbf{U}_f 表示网格单元面心的速度。*input* 表示输入的场, 在方程(7.116)中则输入通量场 $\phi_f = \mathbf{S}_f \cdot \mathbf{U}_f$ 。需要注意的是, $\mathbf{n}_f \otimes \mathbf{S}_f$ 为一个二阶张量场。方程(7.117)可以继续整理为⁵⁷:

$$\sum (\mathbf{n}_f \otimes \mathbf{S}_f) \cdot \mathbf{U}_P = \sum \mathbf{n}_f (\mathbf{S}_f \cdot \mathbf{U}_P) = \sum \mathbf{n}_f \phi_f \quad (7.118)$$

略去其中的 \mathbf{n}_f 有下列等式:

$$\sum \mathbf{U}_P \cdot \mathbf{S}_f = \sum \phi_f \quad (7.119)$$

方程(7.119)可以更清晰的理解为 `fvc::reconstruct()` 函数在给定面通量 (右侧的 ϕ_f) 的情况下, 重组的体心速度场 (左侧的 \mathbf{U}_P)。由于矢量并不能求逆, 方程(7.119)虽然更易理解但难以植入, 因此代码中植入的为方程(7.116)。

之所以进行 `fvc::reconstruct()` 函数的原因在于可以防止数值震荡。举例说明: 对于压力梯度项 ∇p , 一种离散方法是将其写为 `fvc::grad(p)`, 其数学形式为⁵⁸:

$$\frac{1}{\Delta V} \int \nabla p dV = \frac{1}{\Delta V} \int p d\mathbf{S} = \frac{1}{\Delta V} \sum p_f \mathbf{S}_f \quad (7.120)$$

⁵⁶ `src/finiteVolume/finiteVolume/fvc/fvcReconstruct.C`

⁵⁷ 基本操作: $(\mathbf{a} \otimes \mathbf{b}) \cdot \mathbf{c} = \mathbf{a}(\mathbf{b} \cdot \mathbf{c})$

⁵⁸ 在 OpenFOAM 中, 虽然要对动量方程做体积分, 但除了时间离散, 诸如对流、散度、梯度等在离散时均除掉了网格体积。

在第7.9.1节的讨论中可以看出，方程(7.120)的离散形式调用非紧致基架点，跨越两个网格点的值，因此可能会产生数值震荡。

∇p 另外一种离散形式即通过fvc::reconstruct()的形式组建。对应的代码为

```

1 fvc::reconstruct
2 (
3     fvc::snGrad(p)*mesh.magSf()
4 )

```

相应的数学形式为

$$\sum \nabla p_P \cdot \mathbf{S}_f = \sum (\nabla p)_f \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|} |\mathbf{S}_f| \quad (7.121)$$

方程(7.121)左侧的 ∇p_P 为fvc::reconstruct()函数返回的变量。参考第7.9.1节的讨论，可以看出方程(7.121)右侧的离散调用紧致基架点，可以防止数值震荡。总体来说，fvc::reconstruct()函数结合面法向梯度，将普通的梯度离散导致的数值震荡抹平，即为fvc::reconstruct()函数存在的意义。

7.9.6 setReference() 函数

setReference() 函数在 OpenFOAM 中被大量的应用于求解压力方程中。这是因为在某些条件下（比如封闭的空腔中），压力方程不需要给定固定值边界条件，即所有边界条件都为固定梯度边界条件。这会导致构建的矩阵系统出现相容性问题。从数学上来讲，对于方程 $-\nabla \cdot (\nabla p) = 0$ ，其一定不能给定固定梯度边界条件。但若其存在其他项如时间项，如 $\frac{\partial p}{\partial t} - \nabla \cdot (\nabla p) = 0$ （在这里我们假定量纲是一致的）则没有问题。

现以一个横向二维网格为例：|0|1|2|，其网格编号从左至右为 0、1、2，同时左右分别给定零法向梯度边界条件，相对应的压力方程为

$$-\frac{\partial}{\partial x} \left(\frac{\partial p}{\partial x} \right) = S \quad (7.122)$$

对其在 |0|1|2| 网格系统上进行离散之后，形成的方程为⁵⁹：

$$\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} p_0 \\ p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} S\Delta x^2 \\ S\Delta x^2 \\ S\Delta x^2 \end{pmatrix} \quad (7.123)$$

可以看出，方程(7.123)中的矩阵的秩为 2，这会导致 p 存在无穷多个解。也即方程奇异。但若对于附加时间项的压力方程：

$$\frac{\partial p}{\partial t} - \frac{\partial}{\partial x} \left(\frac{\partial p}{\partial x} \right) = S \quad (7.124)$$

⁵⁹注意其中矩阵对角线元素为负值，这是因为方程(7.122)左侧的拉普拉斯项为正值，其等于一种负的扩散项（扩散项本身为负）。

时间项的离散会改变离散矩阵，如：

$$\begin{pmatrix} 1 + 1/\Delta t & -1 & 0 \\ -1 & 2 + 1/\Delta t & -1 \\ 0 & -1 & 1 + 1/\Delta t \end{pmatrix} \cdot \begin{pmatrix} p_0 \\ p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} S\Delta x^2 \\ S\Delta x^2 \\ S\Delta x^2 \end{pmatrix} \quad (7.125)$$

其矩阵的阶数为 3，可以求解。

在 OpenFOAM 中，通过 `setReference()` 函数可以将某个网格单元施加一个参考量，并且使得方程增阶，进一步获得解。相应的代码如下⁶⁰：

```
1 source()[cell] += diag()[cell]*value;
2 diag()[cell] += diag()[cell];
```

在实施的过程中，假定要对网格单元 2 给定某参考值 0，可以看出，方程(7.123)的源项不需要变化。但是方程(7.123)的矩阵最中间的元素变成了-4，即：

$$\begin{pmatrix} -1 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} p_0 \\ p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} S\Delta x^2 \\ S\Delta x^2 \\ S\Delta x^2 \end{pmatrix} \quad (7.126)$$

在这种情况下，方程(7.123)有解，且基于 0 上下浮动。本质上，针对离散的方程

$$a_P\psi_P + \sum a_N\psi_N = S \quad (7.127)$$

我们要假定某个网格单元的值在某个值上下浮动，在无任何因素影响的情况下，可以假定 $a_P\psi_P = a_P\psi_{ref}$ ，将其加到方程(7.128)中有：

$$a_P(\psi_P + \psi_P) + \sum a_N\psi_N = S + a_P\psi_{ref} \quad (7.128)$$

整理有：

$$(a_P + a_P)\psi_P + \sum a_N\psi_N = S + a_P\psi_{ref} \quad (7.129)$$

其与代码的操作是相符的。从方程(7.126)中还可以看出，其增加了矩阵的对角占优特性。

7.9.7 `bound()` 函数

在 OpenFOAM 的湍流模型求解时，经常会提示：

```
1 bounding k.water, min: -3.56016e-05 max: 0.09656 average: 0.000370
```

上述代码提示 water 相的湍流动能 k 的最小值为-3.56016e-05，最大值为 0.09656，平均值为 0.000370。bound 函数主要用于处理越界问题。依据湍流动能的定义，其为一个不为负值的函数。因此在出现负的湍流动能的情况下需要进行处理。bound 函数通过一种平均的

⁶⁰fvMatrix.C

操作对场进行操作使其有界。对于图7.25中的 3 个网格单元，其中间的网格单元出现负值，在进行 bound 操作的时候，首先从网格体心对网格面心的值进行线性插值操作求得面心的 k 值，然后通过下述公式三个面 S1, S2, S3 计算面平均值：

$$k_{vol} = \frac{S_1 k_{1,surf} + S_2 k_{2,surf} + S_3 k_{3,surf}}{S_1 + S_2 + S_3} \quad (7.130)$$

其中 $k_{i,surf}$ 是第 i 个网格面的 k 值， S_i 表示第 i 个网格面。

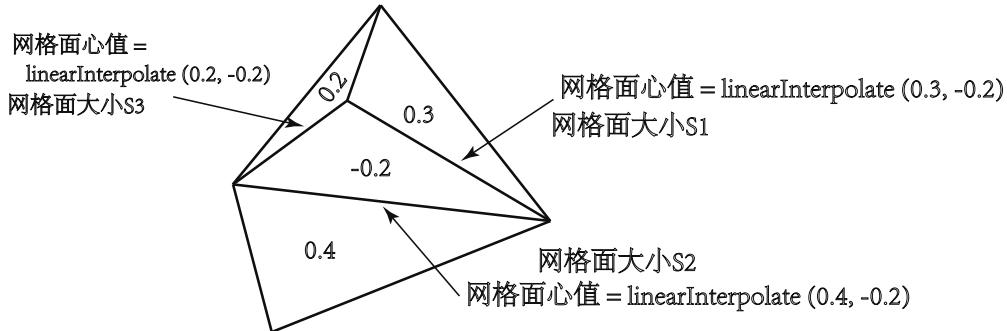


图 7.25: bound 函数示意图。

7.9.8 adjustPhi() 函数

OpenFOAM 中的adjustPhi函数主要用于那些不存在压力固定值边界条件的算例。这是因为压力方程求解的实际就是连续性方程。连续性方程的根本是质量守恒。如果质量不守恒，则压力方程的求解必然会出现问题。因此在求解压力方程之前，要保证质量守恒。另一方面，对于计算域内所有的进出口，必然存在通量（除非速度为 0）。这些流入的通量必然要等于流出的通量，否则结果必然不符合物理。adjustPhi函数将边界条件分为三种类型：固定进口 ϕ_{in} 、固定出口 ϕ_{out} 以及可调整的出口 ϕ_{adjust} 。 ϕ_{in} 即速度固定的流入的边界， ϕ_{out} 即速度固定的流出的边界， ϕ_{adjust} 即为速度可变的边界。adjustPhi函数对所有边界进行遍历，求出所有的 ϕ_{in} , ϕ_{out} , 以及 ϕ_{adjust} 。

在执行的过程中，如果计算过程中 $\phi_{adjust} \neq 0$ ，表示存在可以调整的通量，则定义一个质量修正因子 Corr，其值为：

$$\text{Corr} = \frac{|\phi_{in}| - |\phi_{out}|}{|\phi_{adjust}|} \quad (7.131)$$

同时对于可调整的边界，将其与质量修正因子相乘：

$$\phi_{adjust}^{new} = \text{Corr} \cdot \phi_{adjust} \quad (7.132)$$

在这种情况下，有：

$$|\phi_{adjust}^{new}| = |\phi_{in}| - |\phi_{out}| \quad (7.133)$$

即可保证守恒。如果计算过程中 $\phi_{adjust} = 0$, 则判断 $|\phi_{in}| - |\phi_{out}|$ 的大小。如果在机器精度以下, 说明边界条件是守恒的。如果其值大于机器精度, 则说明进出口通量不守恒, 程序终止并抛出下列错误:

```

1 --> FOAM FATAL ERROR:
2 Continuity error cannot be removed by adjusting the outflow.
3 Please check the velocity boundary conditions and/or run potentialFoam
   to initialise the outflow.
```

在实际使用中, 如果边界条件设置正确, 通常不会抛出该错误。

7.9.9 CourantNo 三维库朗数

OpenFOAM 中的库朗数主要依据可压缩以及不可压缩有不同的计算方式。进一步的, 某一些特殊的求解器可能还存在特殊的库朗数计算方法, 但大体思想一致。在不可压缩流中, OpenFOAM 中通过下式定义库朗数⁶¹:

$$Co = \frac{1}{2} \frac{\sum |\phi_f|}{\Delta V} \Delta t \quad (7.134)$$

其中 ϕ_f 为体积通量。其中要除以 2 是因为对于一个网格单元, 流入的通量与流出的通量必然相等, 在做 $\sum |\phi_f|$ 操作的时候, 其流量为流入的通量或流出的通量的二倍。对于可压缩流动, ϕ_f 为质量通量, 因此库朗数的定义略有区别:

$$Co = \frac{1}{2} \frac{\sum |\phi_f|}{\rho \Delta V} \Delta t \quad (7.135)$$

其中 ρ 表示当前网格单元的密度。在双流体模型中, 还存在相对库朗数的概念, 其定义为:

$$Co = \frac{1}{2} \frac{\sum |\phi_1 - \phi_2|}{\Delta V} \Delta t \quad (7.136)$$

其中 ϕ_1 、 ϕ_2 表示多相流中不同的相通量。在 VOF 模型中, 还存在界面库朗数的概念, 其定义为:

$$Co = \frac{1}{2} \frac{\sum |\phi|}{\Delta V} \Delta t, 0.01 < \alpha < 0.99 \quad (7.137)$$

可见, 界面库朗数仅仅在 $0.01 < \alpha < 0.99$ (也即界面处) 存在值。其可以用于判定界面处速度的大小。

7.9.10 deltaCoeffs() 函数

OpenFOAM 中的 `deltaCoeffs` 返回一个矢量的模的倒数。如图7.26所示, 这个矢量对于内部面, 即为这个内部面相对应的网格单元以及相邻网格单元的距离矢量 \mathbf{d} 。对于边

⁶¹每个网格单元都存在一个库朗数。

界面，由于不存在相邻网格单元，因此这个距离矢量可以表示为边界面对应的网格单元距离边界的矢量。边界处的距离矢量 \mathbf{d} 可以表示为：

$$\mathbf{d} = \mathbf{n}(\mathbf{n} \cdot (\mathbf{V}_{faceCenter} - \mathbf{V}_{owner})) \quad (7.138)$$

那么 `deltaCoeffs` 函数，就是返回的 $1/|\mathbf{d}|$ 。

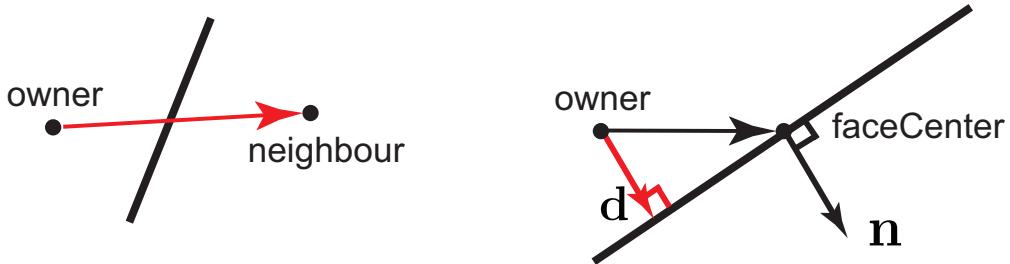


图 7.26: `deltaCoeffs` 函数中使用的 \mathbf{d} 矢量 (红色剪头)。

7.10 OpenFOAM 中的曳力模型

7.10.1 欧拉方法

曳力模型可以表示为

$$\mathbf{M}_{drag} = \frac{3}{4} \alpha_d \rho_c \frac{C_D}{d_d} |\mathbf{U}_c - \mathbf{U}_d| (\mathbf{U}_c - \mathbf{U}_d)$$

注意其中 \mathbf{M}_{drag} 的单位为 $\text{kg}\cdot\text{m}^{-2}\text{s}^{-2}$ ，其为力的单位除以体积单位。在 OpenFOAM 中， C_D 通常与 Re 结合一起来使用形成 $C_D Re = C_D Re$ ，其中 Re 为

$$Re = \frac{d_d |\mathbf{U}_d - \mathbf{U}_c|}{\nu_c}$$

在这种情况下，曳力可以写为：

$$\mathbf{M}_{drag} = \frac{3}{4} \frac{\alpha_d \mu_c}{d_d^2} CdRe (\mathbf{U}_c - \mathbf{U}_d) \quad (7.139)$$

SchillerNaumann

$$CdRe = \begin{cases} 24(1 + 0.15Re^{0.687}), & Re < 1000 \\ 0.44Re, & Re \geq 1000 \end{cases}$$

SyamlalOBrien

$$CdRe = \alpha_c \left(0.63\sqrt{Re} + 4.8\sqrt{V_r} \right) / V_r^2$$

$$V_r = 0.5 \left(A - 0.06Re + \sqrt{(0.06Re)^2 + 0.12Re(2B - A) + A^2} \right)$$

$$A = (\alpha_c)^{4.14}, B = \begin{cases} 0.8\alpha_c^{1.28}, & \alpha_c < 0.85 \\ \alpha_c^{2.65}, & \alpha_c \geq 0.85 \end{cases}$$

Tenneti

$$\begin{aligned} \text{CdRe} &= \text{CdRe}_{iso} + 24\alpha_c^2(F0 + F1) \\ F0 &= 5.81 \frac{\alpha_d}{\alpha_r c^3} + 0.48 \frac{\alpha_d^{1/3}}{\alpha_c^4}, F1 = \alpha_d^3 \text{Re}_s \left(0.95 + 0.61 \frac{\alpha_d^3}{\alpha_c^2} \right) \\ \text{CdRe}_{iso} &= \begin{cases} 24 (1 + 0.15 \text{Re}_s^{0.687}), & \text{Re}_s < 1000 \\ 0.44 \text{Re}_s, & \text{Re}_s \geq 1000 \end{cases} \\ \text{Re}_s &= \alpha_c \text{Re} \end{aligned}$$

TomiyamaKataokaZunSakaguchi

$$\begin{aligned} \text{CdRe} &= \max \left(\frac{24 (1 + 0.15 \text{Re}^{0.687})}{\text{Re}}, \frac{8}{3} \frac{Eo}{3Eo + 4} \right) \text{Re} \\ Eo &= \frac{|\rho_d - \rho_c| |\mathbf{g}| d_d^2}{\sigma} \end{aligned}$$

TomiyamaAnalytic

$$\begin{aligned} \text{CdRe} &= \frac{8}{3} \frac{Eo}{16E^{4/3} + Eo \frac{E^{2/3}}{1-E^2}} \frac{1}{F^2} \text{Re} \\ F &= \frac{\sin(\sqrt{1-E^2}) - E\sqrt{1-E^2}}{1-E^2} \end{aligned}$$

TomiyamaCorrelated

$$\text{CdRe} = \max \left(A \min (1 + 0.15 \text{Re}^{0.687}, 3), \frac{8}{3} \frac{Eo \text{Re}}{3Eo + 4} \right)$$

WenYu

$$\text{CdRe} = (1 - \alpha_d)^{-3.65} \begin{cases} (1 - \alpha_d) 24 (1 + 0.15 \text{Re}^{0.687}), & (1 - \alpha_d) \text{Re} < 1000 \\ (1 - \alpha_d) 0.44 \text{Re}, & (1 - \alpha_d) \text{Re} \geq 1000 \end{cases}$$

Lain

$$\text{CdRe} = \begin{cases} 16 \text{Re}, & \text{Re} < 1.5 \\ 14.9 \text{Re}^{0.22}, & 1.5 \leq \text{Re} < 80 \\ 48 (1 - \frac{2.21}{\text{Re}}), & 80 \leq \text{Re} < 1500 \\ 2.51 \text{Re}, & \text{Re} \geq 1500 \end{cases}$$

IshiZuber

$$\text{CdRe} = \begin{cases} \min (\text{CdRe}_{ell}, 2.66667 \text{Re} (1 - \alpha_d)^2), & \text{CdRe}_{ell} \geq \text{CdRe} \\ \text{CdRe}, & \text{CdRe}_{ell} < \text{CdRe} \end{cases}$$

$$\text{CdRe}_{ell} = 0.66666 E_\alpha \text{Re} \sqrt{E_o}, E_\alpha = \frac{1 + 17.67 F^{0.8571428}}{18.67 F}$$

$$F = \max \left(\frac{\mu_c}{\mu_m} \sqrt{1 - \alpha_d}, 0.001 \right), \text{Re}_m = \frac{\mu_c}{\mu_m} \text{Re}$$

$$\text{CdRe} = \begin{cases} 24 (1 + 0.15 \text{Re}_m^{0.687}), & \text{Re}_m < 1000 \\ 0.44 \text{Re}_m, & \text{Re}_m \geq 1000 \end{cases}$$

$$\mu_m = \mu_c (1 - \alpha_d, 0.001)^{-2.5 \frac{\mu_d + 0.4 \mu_c}{\mu_d + \mu_c}}$$

GidaspowSchillerNaumann

$$\text{CdRe} = \alpha_c (1 - \alpha_d)^{-2.65} \begin{cases} \frac{24}{1 - \alpha_d} (1 + 0.15 \text{Re}^{0.687}), & (1 - \alpha_d) \text{Re} < 1000 \\ (1 - \alpha_d) 0.44 \text{Re}, & (1 - \alpha_d) \text{Re} \geq 1000 \end{cases}$$

GidaspowWenYu

$$\text{CdRe} = \begin{cases} \text{CdRe}_{\text{WenYu}} & \alpha_c \geq 0.8 \\ \text{CdRe}_{\text{Ergun}} & \alpha_c < 0.8 \end{cases}$$

Gibilaro

$$\text{CdRe} = \frac{4}{3} \alpha_c \left(\frac{17.3}{1 - \alpha_d} + 0.336 \text{Re} \right) (1 - \alpha_d)^{-2.8}$$

Ergun

$$\text{CdRe} = \frac{4}{3} \left(150 \frac{1 - \alpha_c}{\alpha_c} + 1.75 \text{Re} \right)$$

Beetstra

$$\text{CdRe} = 24 (1 - \alpha_d) (F0 + F1)$$

$$F0 = 10 \frac{\alpha_d}{(1 - \alpha_d)^2} + (1 - \alpha_d)^2 (1 + 1.5 \sqrt{\alpha_d})$$

$$F1 = 0.413 \frac{\text{Re}_s}{24(1 - \alpha_d)^2} \frac{\frac{1}{1 - \alpha_d} + 3\alpha_d(1 - \alpha_d) + 8.4 \text{Re}_L^{-0.343}}{1 + 10^{3\alpha_d} \text{Re}_L^{-(1+4\alpha_d)/2}}$$

$$\text{Re}_L = (1 - \alpha_d) \text{Re}$$

7.11 OpenFOAM 中的状态方程

rhoConst

$$\rho = \rho_{\text{const}}$$

perfectGas

$$\rho = \frac{p}{RT}$$

perfectFluid

$$\rho = \rho_0 + \frac{p}{RT}$$

linear

$$\rho = \rho_0 + \psi p$$

incompressiblePefectGas

$$\rho = \frac{p_0}{RT}$$

icoPolynomial

$$\rho = a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4 + a_5 T^5 + a_6 T^6 + a_7 T^7 + a_8 T^8$$

Boussinesq

$$\rho = \rho_0(1 - \beta(T - T_0))$$

7.12 OpenFOAM 中的边界条件

7.12.1 边界条件计算方法

有限体积法中边界条件的作用通过改变稀疏矩阵的系数与源项来进行。考虑图6.13所示的一维网格单元，结合典型的无量纲化一维传输方程（各项系数均为一）：

$$\frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} - \frac{\partial}{\partial x} \left(\frac{\partial \phi}{\partial x} \right) = S \quad (7.140)$$

对 A 网格点进行离散有⁶²：

$$\frac{\phi_A^{t+\Delta t} - \phi_A}{\Delta t} + \frac{\phi_A - \phi_0}{0.5 \Delta x} - \frac{\frac{\phi_B - \phi_A}{\Delta x} - \frac{\partial \phi_0}{\partial x}}{\Delta x} = S \quad (7.141)$$

可以看出边界值 ϕ_0 （边界条件）对时间项与源项 S 并无影响。大部分情况下，边界条件可以表示为固定值边界条件与零法向梯度边界条件的组合形式。因此，分析固定值边界条件与零法向梯度边界条件是理解边界条件植入的必要条件。

在分析各项之前，在这里要说明的是在 OpenFOAM 中，对流项边界条件对矩阵对角线系数的影响通过 `valueInternalCoeffs` 进行更新⁶³，对矩阵源项的影响通过 `valueBoundaryCoeffs` 进行更新。扩散项边界条件对矩阵对角线系数的影响通过 `gradientInternalCoeffs` 进行更新，对矩阵源项的影响通过 `gradientBoundaryCoeffs` 进行更新。

对于对流项，固定值边界条件值 ϕ_0 为固定值，因此，其并不改变 ϕ_A 的对角系数 `valueInternalCoeffs`。同时， $\frac{\phi_0}{\Delta x}$ 将进入离散矩阵的源项部分，改变 `valueBoundaryCoeffs`。若其为零法向梯度

⁶²假定正向迎风格式： $\phi_1 = \phi_A$

⁶³注意区分控制方程的离散形式和连续形式，方程7.140为连续形式，对其进行有限体积法离散之后，每个网格点均具备方程7.141的形式，由于存在大量网格点，若干个方程7.141组成稀疏矩阵系统，通常写为 $a_P \phi_P + \sum a_N \phi_N = S$ ，这里的 S 即为矩阵源项，其不同于方程7.140中的 S 。 a_P 为矩阵对角线系数。

边界条件，则 $\phi_0 = \phi_A$ ，在离散方程中，对流项变为 $\frac{\phi_1 - \phi_0}{\Delta x}$ ，因此，其改变 ϕ_A 的对角系数 `valueInternalCoeffs`，但并没有项进入矩阵源项部分，因此矩阵源项 `valueBoundaryCoeffs` 不变。

类似的，对于扩散项，若其为固定值边界条件，有：

$$\frac{\partial \phi_0}{\partial x} = \frac{\phi_A - \phi_0}{d\Delta x} \quad (7.142)$$

因此，固定值边界条件将改变矩阵对角线系数 `gradientInternalCoeffs`（乘以 $\frac{1}{d}$ ⁶⁴），同时改变矩阵源项 `gradientBoundaryCoeffs` ($\frac{\phi_0}{d\Delta x}$)。若其为零法向梯度边界条件，则 $\frac{\partial \phi_0}{\partial x} = 0$ ，则其不改变对角系数也不改变源项。

7.12.2 OpenFOAM 边界条件速查

activeBaffleVelocity: 可穿透的速度挡板边界条件。是否穿透取决于压力。

```

1 <patchName>
2 {
3     type          activeBaffleVelocity;
4     p             p;
5     cyclicPatch  cyclic1;
6     orientation   1;
7     openFraction  0.2;
8     openingTime   5.0;
9     maxOpenFractionDelta 0.1;
10 }
```

$$x = x_{old} + sign(F_{net}) \frac{\Delta t}{\Delta T} \quad (7.143)$$

其中 x 表示开放程度， x_{old} 表示上一个时间步的开放程度， F_{net} 表示单位面积的压力，其值为 $\sum p_f |\mathbf{S}_f|$ ， Δt 表示时间步， ΔT 表示用多少时间去决定是否开启或者关闭。

advective: 基于 `mixed` 边界条件的出口边界条件。

```

1 <patchName>
2 {
3     type          advective;
4     phi           phi;
5 }
```

该边界条件通过求解下述方程获得 ψ 的边界值

$$\frac{\partial \psi}{\partial t} + W \nabla_f \psi = \frac{W'}{l_{inf}} (\psi_{ref} - \psi) \quad (7.144)$$

⁶⁴ $1/d$ 为 `this->patch().deltaCoeffs()`

其中 W 表示传输速度:

$$W = \frac{\phi_f}{|\mathbf{S}_f|} \quad (7.145)$$

其中 ϕ_f 表示通量, l_{inf} 表示参考距离, ψ_{inf} 表示参考未知量, $\nabla_f \psi$ 表示面法向梯度, 为一个标量。针对亚音速与超音速, 方程右侧具有不同的形式。首先定义近似库朗数为

$$\alpha = \frac{\phi_f}{|\mathbf{S}_f|} \frac{\Delta t}{\Delta d} \quad (7.146)$$

若未指定参考距离 (通常为亚音速) 的情况, $W' = 0$, 采用欧拉时间格式, 其离散形式为

$$\frac{\psi^{t+\Delta t} - \psi^t}{\Delta t} + \frac{\phi_f}{|\mathbf{S}_f|} \frac{\psi^{t+\Delta t} - \psi_P}{\Delta d} = 0 \quad (7.147)$$

方程的解为

$$\psi^{t+\Delta t} = w\psi' + (1-w)\phi_P \quad (7.148)$$

$$\psi' = \psi^t \quad (7.149)$$

$$w = \frac{1}{1 + \alpha} \quad (7.150)$$

若指定参考距离 (通常为超音速) 的情况, $W' = W$, 采用欧拉时间格式, 其离散形式为

$$\frac{\psi^{t+\Delta t} - \psi^t}{\Delta t} + \frac{\phi_f}{|\mathbf{S}_f|} \frac{\psi^{t+\Delta t} - \psi_P}{\Delta d} = \frac{1}{l_{inf}} \frac{\phi_f}{|\mathbf{S}_f|} (\psi_{ref} - \psi^{t+\Delta t}) \quad (7.151)$$

其解为

$$\psi^{t+\Delta t} = \frac{k\psi_{ref} + \psi^t}{1 + \alpha + k} + \frac{\alpha}{1 + \alpha + k} \psi_P \quad (7.152)$$

写成通用形式即

$$\psi^{t+\Delta t} = w\psi' + (1-w)\phi_P \quad (7.153)$$

$$\psi' = \frac{\psi^t + k\psi_{ref}}{1 + k} \quad (7.154)$$

$$w = \frac{1 + k}{1 + \alpha + k} \quad (7.155)$$

fixedValue: 固定值边界条件。

inletOutlet: 在面通量小于 0 的时候 (进口), 给定固定值边界条件, 在面通量大于 0 的时候 (出口), 给定零法向梯度边界条件。

$$\psi_f = w\psi_{fixedValue} + (1-w)\psi_P$$

其中 w 为权重, 其值取决于面通量。

freestreamPressure: 对于压力的逐渐过渡的固定值边界条件与零法向梯度混合边界条件。

$$\psi_f = w\psi_{fixedValue} + (1-w)(\nabla_f \psi \Delta d)$$

其中 w 为权重，在亚音速的情况下：

$$w = 0.5 + 0.5 \frac{\mathbf{U}_f \cdot \mathbf{n}_f}{|\mathbf{U}_f|}$$

其中 \mathbf{n}_f 表示面单位矢量， \mathbf{U}_f 表示面上的速度。在超音速的情况下：

$$w = 0.5 - 0.5 \frac{\mathbf{U}_f \cdot \mathbf{n}_f}{|\mathbf{U}_f|}$$

权重逐渐过渡，其值位于 0, 1 之间。

freestreamVelocity: 对于速度的逐渐过渡的固定值边界条件与零法向梯度混合边界条件。

$$\psi_f = w\psi_{fixedValue} + (1 - w)(\nabla_f \psi \Delta d)$$

其中 w 为权重：

$$w = 0.5 - 0.5 \frac{\mathbf{U}_f \cdot \mathbf{n}_f}{|\mathbf{U}_f|}$$

权重逐渐过渡，其值位于 0, 1 之间。

freestream: 在面通量小于 0 的时候（进口），给定固定值边界条件，其值通过 `freestreamValue` 指定，在面通量大于 0 的时候（出口），给定零法向梯度边界条件。其等同于 `inletOutlet` 边界条件。在压力、速度基于 `freestreamPressure` 与 `freestreamVelocity` 指定时，其他变量（如湍流变量）可指定为 `freestream` 或 `inletOutlet`。`freestream` 的权重固定，要么是 0，要么是 1，不存在中间过渡状态。

mixed: 混合固定参考值/面法向梯度边界条件。

$$\psi_f = w\psi_{refValue} + (1 - w)(\psi_P + \nabla_f \psi y) \quad (7.156)$$

其中 $\nabla_f \psi$ 表示面法向梯度， $\psi_{refValue}$ 为固定参考值。下面是一个使用范例：

```

1  upperWall
2  {
3      type          mixed;
4      refValue      uniform (0 0 0);
5      refGradient   uniform (0 0 0);
6      valueFraction uniform 0.5;
7 }
```

其可以实现一种半滑移边界条件。

totalPressure: 总压边界条件。OpenFOAM 中的总压边界条件中的总压定义为 p_0 。其可分为 4 种类型：不可压缩低音速情况下静压可以这样计算：

$$p = p_0 - \frac{1}{2} |\mathbf{U}|^2 \quad (7.157)$$

在使用的时候需要给定 p_0 的值。可压缩低音速情况下静压可以这样计算：

$$p = p_0 - \frac{1}{2} \rho |\mathbf{U}|^2 \quad (7.158)$$

在使用的时候需要给定 p_0 、 rho 的值。可压缩近音速情况下静压可以这样计算：

$$p = \frac{p_0}{1 + 0.5\psi|U|^2} \quad (7.159)$$

在使用的时候需要给定 p_0 、 psi 的值。其中 ψ 表示可压缩性，其定义为 $\psi = \rho/p$ 。可压缩超音速情况下静压可以这样计算：

$$p = \frac{p_0}{(1 + 0.5\psi G|U|^2)^{\frac{1}{G}}} \quad (7.160)$$

在使用的时候需要给定 p_0 、 psi 、 G 的值。其中 G 为 $\frac{\gamma}{1-\gamma}$ ，其中 γ 为 C_p/C_v 。

`totalPressure` 压力边界条件通常与 `pressureInletOutletVelocity` 速度边界条件一起使用。在超音速情况下，`totalPressure` 压力边界条件一般和 `totalTemperature` 总温边界条件一起使用。实际应用中，对于回流不是很强的出口，也可以指定固定值压力边界结合零法向速度边界。对于回流很强的出口（同时有入流同时有出流，类似开放的卷吸边界），建议使用 `totalPressure+pressureInletOutletVelocity`。

totalTemperature: 总温边界条件。参考(6.173)，依据总温的定义，结合下列公式：

$$Ma = \frac{|\mathbf{U}|}{c}, c = \sqrt{\gamma RT}, \psi = \frac{1}{RT} \quad (7.161)$$

有

$$T = \frac{T_0}{1 + \frac{1}{2} \frac{\gamma-1}{\gamma} \frac{1}{RT} |\mathbf{U}|^2} \quad (7.162)$$

方程(7.162)即 OpenFOAM 植入的总温边界条件。

fixedFluxPressure: 一种压力梯度边界条件。主要用于密度存在变化的流动，如多相流以及传热流动。首先考虑 2D 稳态不可压缩流动（或者局部区域沿着 x 方向速度无变化），壁面法向 y 方向的动量方程可以写为

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = - \frac{\partial p}{\partial y} + \nu \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} \right) \quad (7.163)$$

在 x 方向， u, v 均无变化，即 $\partial u / \partial x = \partial v / \partial x = 0$ 。同时可以得出扩散项为 0。结合连续性方程 $\partial u / \partial x + \partial v / \partial y = 0$ ，有 $\partial v / \partial y = 0$ 。结合方程(7.163)，有

$$\frac{\partial p}{\partial y} = 0 \quad (7.164)$$

在三维情况下可以写为：

$$\nabla p \cdot \mathbf{n} = 0 \quad (7.165)$$

其中 $\mathbf{n} = (0, 1, 0)$ 。压力零法向梯度边界条件的物理解释为壁面附近的速度不可能继续冲向壁面以及远离壁面。方程(7.164)即为压力零梯度边界条件。

下面考虑可压缩 2D 稳态流动，壁面法向 y 方向的动量方程可以写为

$$\rho u \frac{\partial v}{\partial x} + \rho v \frac{\partial v}{\partial y} = -\frac{\partial p_{rgh}}{\partial y} + gh \frac{\partial \rho}{\partial y} \quad (7.166)$$

类似的，有：

$$\frac{\partial p_{rgh}}{\partial y} = -gh \frac{\partial \rho}{\partial y} \quad (7.167)$$

在三维情况下为：

$$\nabla p_{rgh} \cdot \mathbf{n} = (\mathbf{g} \cdot \mathbf{h} \nabla \rho) \cdot \mathbf{n} \quad (7.168)$$

OpenFOAM 部分求解器求解的为 p_{rgh} 方程，因此在壁面处， p_{rgh} 不应给定零法向梯度边界条件，因为其值与密度法向梯度有关。同时，若边界存在速度（方程(7.167)左侧不为 0），方程(7.167)也应该考虑速度的影响。考虑可压缩稳态流动：

$$\begin{aligned} \nabla \cdot (\rho \mathbf{U}) &= 0 \\ \nabla \cdot (\rho \mathbf{U} \mathbf{U}) &= -\nabla p_{rgh} - \mathbf{g} \cdot \mathbf{h} \nabla \rho \end{aligned} \quad (7.169)$$

压力边界条件在使用的过程中，要实施到压力方程中。在这里简述压力方程的组建过程⁶⁵：

1. 首先略去压力项，通过动量方程获得预测速度 \mathbf{U}^* ，通过预测速度 \mathbf{U}^* 组建 n 次迭代步速度 \mathbf{HbyA}^n ；
2. 在 n 次迭代步速度 \mathbf{HbyA}^n 的基础上，考虑重力等其他体积力，组建通量 $\mathbf{phiHbyA}^n$ ；
3. 考虑压力项后的通量 $\mathbf{phiHbyA}^n$ 应该符合连续性方程，此步可组建压力泊松方程用于求解下一个迭代步的压力：

$$\sum \rho_f^n \left(\mathbf{phiHbyA}_f^n - \frac{1}{A_{p,f}} \nabla_f p_{rgh}^{n+1} \right) \cdot \mathbf{S}_f = 0 \quad (7.170)$$

方程(7.170)在壁面处可以实施边界条件，或固定值边界条件 ($p_{rgh} = A$, A 为某个值)，或梯度边界条件 ($\nabla p_{rgh} = A$) 等。fixedFluxPressure 边界条件为一种梯度边界条件，在边界处，有严格的连续性方程：

$$\sum \rho_f^n \mathbf{U} \cdot \mathbf{S}_f = 0 \quad (7.171)$$

压力边界条件可以从方程(7.171)推导而来。将方程(7.171)与(7.170)相减有：

$$-\sum \rho_f^n \left(\mathbf{phiHbyA}_f^n - \frac{1}{A_{p,f}} \nabla_f p_{rgh}^{n+1} \right) \cdot \mathbf{S}_f + \sum \rho_f^n \mathbf{U}_f \cdot \mathbf{S}_f = 0 \quad (7.172)$$

⁶⁵ 更详细的内容请参考 [icoFoam 解析](#)。

有：

$$\sum \rho_f^n \left(\frac{1}{A_{p,f}} \nabla_f p_{rgh}^{n+1} \right) \cdot \mathbf{S}_f = \sum \rho_f^n (\mathbf{phiHbyA}^n_f - \mathbf{U}_f) \cdot \mathbf{S}_f \quad (7.173)$$

即

$$\sum \nabla_f p_{rgh}^{n+1} \cdot \mathbf{n}_f = \frac{\sum (\mathbf{phiHbyA}^n_f - \mathbf{U}_f) \cdot \mathbf{S}_f}{|\mathbf{S}_f| \frac{1}{A_{p,f}}} \quad (7.174)$$

由于大部分情况下边界处仅存在一个面，即

$$\nabla_f p_{rgh}^{n+1} \cdot \mathbf{n}_f = \frac{(\mathbf{phiHbyA}^n_f - \mathbf{U}_f) \cdot \mathbf{S}_f}{|\mathbf{S}_f| \frac{1}{A_{p,f}}} \quad (7.175)$$

方程(7.175)即为 OpenFOAM 中 fixedFluxPressure 植入的边界条件最终形式。需要注意的是，如果仅仅考虑重力的作用，方程(7.175)与(7.168)是一致的。

prghPressure: 一种压力固定值边界条件。计算公式为

$$p_{rgh} = p - \rho g (H - H_{ref}) \quad (7.176)$$

其中 H_{ref} 表示参考高度，默认为 0, p 表示用户给定的压力固定值。相对于 fixedValue 固定值边界条件，尤其是出口可能存在多相时，prghPressure 可以依据密度进行调整。prghPressure 更适用于 interFoam 以及 multiPhaseEulerFoam 之类求解器（如图7.27所示范例）。

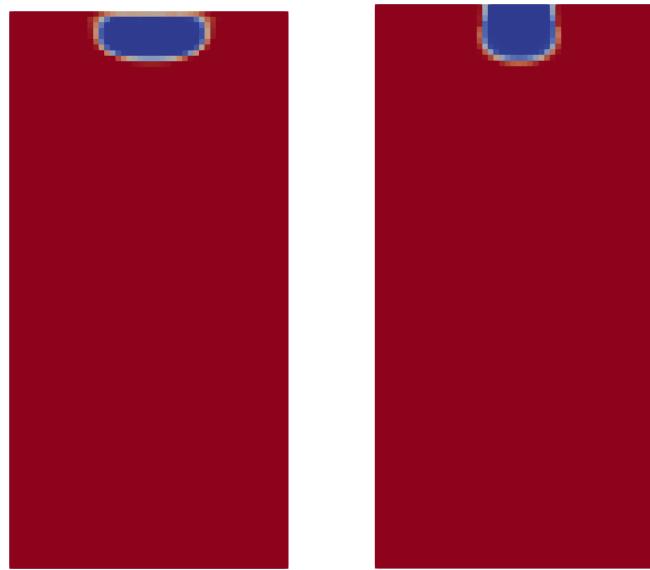


图 7.27: 对 p_{rgh} 场应用 fixedValue (左侧) 以及 prghPressure (右侧) 模拟的单气泡自由上浮。fixedValue 边界导致气泡困在流场中。

uniformFixedValue: 一种简单的固定值边界条件，但是可以依据时间进行变化。下面是一个使用范例：

```
1  inlet
2  {
3      type          uniformFixedValue;
4      uniformValue   table
5      (
6          (0    (0 0 0))
7          (5    (10 0 0))
8      );
9 }
```

上述代码实现的是随着时间变化的固定值边界条件。在 0-5 秒的时候，速度是 x 方向 0m/s，在 5-10 秒的时候，速度是 x 方向 5m/s。

附录 A 10 网格 1D 方程离散

A.1 扩散方程 1

如图A.1所示，本节对一个10网格点的一维稳态扩散方程进行离散，相应的控制方程如下¹：

$$-\frac{\partial}{\partial x} \left(\frac{\partial \phi}{\partial x} \right) = K \quad (\text{A.1})$$

在这个算例中，我们假定左侧为固定值边界条件 $\phi = \phi_w$ ，右侧为固定梯度边界条件 $\frac{\partial \phi}{\partial x} = G$ 。

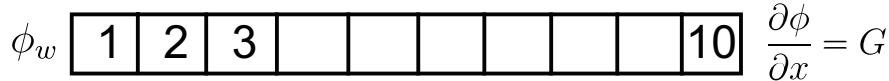


图 A.1: 一维 10 网格示意图。

且在网格中存在非均一分布的源项：

$$K_i \neq 0, i = 5, 6, 7 \quad (\text{A.2})$$

首先对内部网格点 2 – 9 进行离散，有：

$$\begin{aligned} -\phi_1 + 2\phi_2 - \phi_3 &= 0, \\ -\phi_2 + 2\phi_3 - \phi_4 &= 0, \\ -\phi_3 + 2\phi_4 - \phi_5 &= 0, \\ -\phi_4 + 2\phi_5 - \phi_6 &= K_5 \Delta x^2, \\ -\phi_5 + 2\phi_6 - \phi_7 &= K_6 \Delta x^2, \\ -\phi_6 + 2\phi_7 - \phi_8 &= K_7 \Delta x^2, \\ -\phi_7 + 2\phi_8 - \phi_9 &= 0, \\ -\phi_8 + 2\phi_9 - \phi_{10} &= 0 \end{aligned} \quad (\text{A.3})$$

对于左边的固定值边界：

$$3\phi_1 - \phi_2 = \phi_w \quad (\text{A.4})$$

¹ 在这里并不能够严格的区分有限差分或者有限体积，因为离散后的形式是完全一样的。

对于右边的固定梯度边界：

$$\phi_{10} - \phi_9 = G \quad (\text{A.5})$$

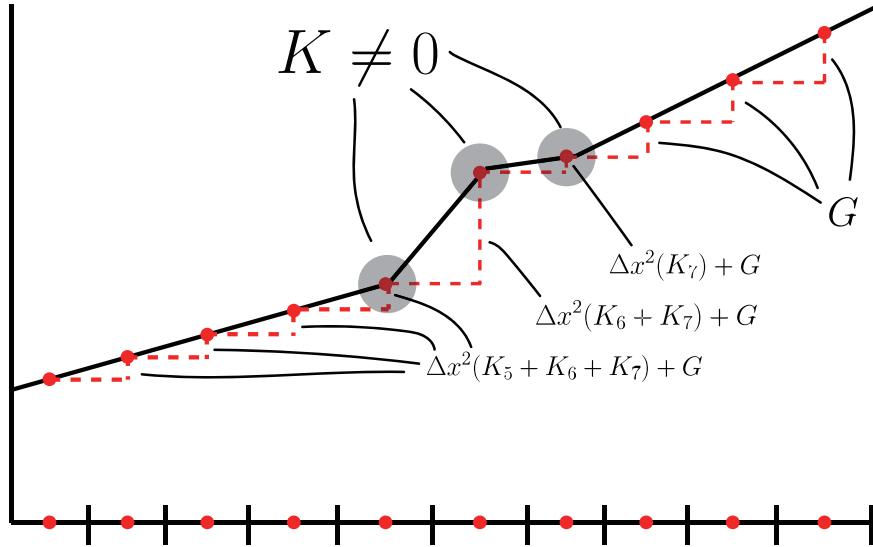
对这些网格点组建矩阵系统有：

$$\begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \\ \phi_{10} \end{bmatrix} = \begin{bmatrix} \phi_w \\ 0 \\ 0 \\ 0 \\ K_5 \Delta x^2 \\ K_6 \Delta x^2 \\ K_7 \Delta x^2 \\ 0 \\ 0 \\ G \end{bmatrix} \quad (\text{A.6})$$

其解为：

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \\ \phi_{10} \end{bmatrix} = \begin{bmatrix} \phi_w + 0.5G + 0.5\Delta x^2(K_5 + K_6 + K_7) \\ \phi_w + 1.5G + 1.5\Delta x^2(K_5 + K_6 + K_7) \\ \phi_w + 2.5G + 2.5\Delta x^2(K_5 + K_6 + K_7) \\ \phi_w + 3.5G + 3.5\Delta x^2(K_5 + K_6 + K_7) \\ \phi_w + 4.5G + 4.5\Delta x^2(K_5 + K_6 + K_7) \\ \phi_w + 5.5G + 4.5\Delta x^2K_5 + 5.5\Delta x^2(K_6 + K_7) \\ \phi_w + 6.5G + 4.5\Delta x^2K_5 + 5.5\Delta x^2K_6 + 6.5\Delta x^2K_7 \\ \phi_w + 7.5G + 4.5\Delta x^2K_5 + 5.5\Delta x^2K_6 + 6.5\Delta x^2K_7 \\ \phi_w + 8.5G + 4.5\Delta x^2K_5 + 5.5\Delta x^2K_6 + 6.5\Delta x^2K_7 \\ \phi_w + 9.5G + 4.5\Delta x^2K_5 + 5.5\Delta x^2K_6 + 6.5\Delta x^2K_7 \end{bmatrix} \quad (\text{A.7})$$

现在对结果进行分析。如果方程(A.9)中没有源项 (K 全部为 0)，则结果简化为：

图 A.2: 一维 10 网格点计算的 ϕ 。

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \\ \phi_{10} \end{bmatrix} = \begin{bmatrix} \phi_w + 0.5G \\ \phi_w + 1.5G \\ \phi_w + 2.5G \\ \phi_w + 3.5G \\ \phi_w + 4.5G \\ \phi_w + 5.5G \\ \phi_w + 6.5G \\ \phi_w + 7.5G \\ \phi_w + 8.5G \\ \phi_w + 9.5G \end{bmatrix} \quad (\text{A.8})$$

ϕ 的结果如图 A.2 所示, 可见:

- ϕ 在 x 方向的分布取决于右侧的固定梯度边界条件。固定梯度的值 G , 决定了 ϕ 在 x 方向分布的斜率 (梯度)。如果 $G = 0$ (零法向梯度), 则为均一分布。
- ϕ 左侧的固定值边界条件, 决定了 ϕ 的参考水平。

若 K 部分不为 0, 有:

- 从 $K \neq 0$ 最左侧的网格 (含网格单元 5) 向左起, 对于网格单元 1、2、3、4、5, 虽然 $K = 0$, 但面上 ϕ 的梯度也收到 K 的值的影响进而发生变化, 其值为 $\partial\phi/\partial x = \Delta x^2(K_5 + K_6 + K_7) + G$ 。
- 从 $K \neq 0$ 最右侧的网格 (含网格单元 7) 向右起, 对于网格单元 7、8、9、10, 其面上的梯度还是仅仅收到右侧固定梯度边界条件的影响: $\partial\phi/\partial x = G$ 。

- 对于其中的网格单元 5、6、7，其梯度发生变化，例如对于 5、6 网格单元，其面上的梯度 $\partial\phi/\partial x = \Delta x^2(K_6 + K_7) + G$ 。对于 6、7 网格单元，其面上的梯度 $\partial\phi/\partial x = \Delta x^2K_7 + G$ 。

A.2 扩散方程 2

同样考虑下述控制方程：

$$-\frac{\partial}{\partial x} \left(\frac{\partial\phi}{\partial x} \right) = K \quad (\text{A.9})$$

左侧为固定值边界条件 $\phi = \phi_w$ ，右侧为固定梯度边界条件 $\frac{\partial\phi}{\partial x} = G$ 。但在网格中各点都存在非均一分布的源项：

$$K_i \neq 0, i = 1, 2, \dots, 10 \quad (\text{A.10})$$

离散后的矩阵为：

$$\begin{aligned} 3\phi_1 - \phi_2 &= \phi_w + K_1\Delta x^2, \\ -\phi_1 + 2\phi_2 - \phi_3 &= K_2\Delta x^2, \\ -\phi_2 + 2\phi_3 - \phi_4 &= K_3\Delta x^2, \\ -\phi_3 + 2\phi_4 - \phi_5 &= K_4\Delta x^2, \\ -\phi_4 + 2\phi_5 - \phi_6 &= K_5\Delta x^2, \\ -\phi_5 + 2\phi_6 - \phi_7 &= K_6\Delta x^2, \\ -\phi_6 + 2\phi_7 - \phi_8 &= K_7\Delta x^2, \\ -\phi_7 + 2\phi_8 - \phi_9 &= K_8\Delta x^2, \\ -\phi_8 + 2\phi_9 - \phi_{10} &= K_9\Delta x^2, \\ \phi_{10} - \phi_9 &= G + K_{10}\Delta x^2 \end{aligned} \quad (\text{A.11})$$

对这些网格点组建矩阵系统有：

$$\left[\begin{array}{cccccccccc} 3 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{array} \right] \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \\ \phi_{10} \end{bmatrix} = \begin{bmatrix} \phi_w + K_1\Delta x^2 \\ K_2\Delta x^2 \\ K_3\Delta x^2 \\ K_4\Delta x^2 \\ K_5\Delta x^2 \\ K_6\Delta x^2 \\ K_7\Delta x^2 \\ K_8\Delta x^2 \\ K_9\Delta x^2 \\ G + K_{10}\Delta x^2 \end{bmatrix} \quad (\text{A.12})$$

矩阵存在逆矩阵, 为:

$$\begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ 0.5 & 1.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 4.5 & 4.5 & 4.5 & 4.5 & 4.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 5.5 & 5.5 & 5.5 & 5.5 & 5.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 5.5 & 6.5 & 6.5 & 6.5 & 6.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 5.5 & 6.5 & 7.5 & 7.5 & 7.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 5.5 & 6.5 & 7.5 & 8.5 & 8.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 5.5 & 6.5 & 7.5 & 8.5 & 9.5 \end{bmatrix} \quad (\text{A.13})$$

其解为:

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \\ \phi_{10} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ 0.5 & 1.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 & 3.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 4.5 & 4.5 & 4.5 & 4.5 & 4.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 5.5 & 5.5 & 5.5 & 5.5 & 5.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 5.5 & 6.5 & 6.5 & 6.5 & 6.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 5.5 & 6.5 & 7.5 & 7.5 & 7.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 5.5 & 6.5 & 7.5 & 8.5 & 8.5 \\ 0.5 & 1.5 & 2.5 & 3.5 & 4.5 & 5.5 & 6.5 & 7.5 & 8.5 & 9.5 \end{bmatrix} \cdot \begin{bmatrix} \phi_w + K_1 \Delta x^2 \\ K_2 \Delta x^2 \\ K_3 \Delta x^2 \\ K_4 \Delta x^2 \\ K_5 \Delta x^2 \\ K_6 \Delta x^2 \\ K_7 \Delta x^2 \\ K_8 \Delta x^2 \\ K_9 \Delta x^2 \\ G + K_{10} \Delta x^2 \end{bmatrix} \quad (\text{A.14})$$

现在对结果进行分析。首先看左侧第一个网格, 有:

$$\phi_1 = 0.5(\phi_w + \sum K_i \Delta x^2 + G) \quad (\text{A.15})$$

如果方程(A.14)中右侧的 $G = 0$, 即零梯度, 有:

$$\phi_1 = 0.5(\phi_w + \sum K_i \Delta x^2) \quad (\text{A.16})$$

$$\frac{\phi_1 - \phi_w}{\frac{1}{2} \Delta x} = \sum K_i \Delta x \quad (\text{A.17})$$

可见, 左侧边界处的梯度的值为 $(\sum K_i) \Delta x$ 。其表示的是:

- 对于扩散方程, 左侧固定值, 右侧零梯度, 最左侧网格的梯度取决于所有网格点的源项。

- 方程(A.14)可以看出，对于扩散方程，如果每个网格存在源项，其会影响所有网格点的值。
- 右侧固定梯度边界条件的值 G ，对所有网格点的值都存在影响。

A.3 扩散方程 3

现在考虑另外一种方程形式：

$$-\frac{\partial}{\partial x} \left(\sigma \frac{\partial \phi}{\partial x} \right) = 0 \quad (\text{A.18})$$

左侧为固定值边界条件 $\phi = \phi_w$ ，右侧为固定梯度边界条件 $\frac{\partial \phi}{\partial x} = G$ 。其中 σ_s 是一种非均一分布，其

$$\begin{aligned} \sigma_1 &= \sigma_{ref}, \sigma_2 = \sigma_{ref}, \sigma_3 = \sigma_{ref}, \sigma_4 = \sigma_{ref}, \sigma_8 = \sigma_{ref}, \sigma_9 = \sigma_{ref}, \sigma_{10} = \sigma_{ref}, \\ \sigma_5 &= \sigma_6 = \sigma_7 = 0, \end{aligned} \quad (\text{A.19})$$

后续会发现，由于扩散系数为 0，会导致方程奇异。在这里要注意的是，当扩散系数为 0 的时候，方程并无意义。但本算例仅仅为了显示数值算法的意义。

首先对内部网格点进行离散，有：

$$\begin{aligned} 3\phi_1 - \phi_2 &= \phi_w, \\ -\phi_1 + 2\phi_2 - \phi_3 &= 0, \\ -\phi_2 + 2\phi_3 - \phi_4 &= 0, \\ -\phi_3 + \frac{3}{2}\phi_4 - \frac{1}{2}\phi_5 &= 0, \\ -\phi_4 + \phi_5 &= 0, \\ -\phi_5 + 2\phi_6 - \phi_7 &= 0, \\ \phi_7 - \phi_8 &= 0, \\ -\frac{1}{2}\phi_7 + \frac{3}{2}\phi_8 - \phi_9 &= 0, \\ -\phi_8 + 2\phi_9 - \phi_{10} &= 0, \\ \phi_{10} - \phi_9 &= G \end{aligned} \quad (\text{A.20})$$

对这些网格点组建矩阵系统有：

$$\begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 3/2 & -1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/2 & 3/2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \\ \phi_{10} \end{bmatrix} = \begin{bmatrix} \phi_w \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ G \end{bmatrix} \quad (\text{A.21})$$

上述矩阵系统奇异，因此无解。在这里需要注意，对于固定扩散系数的扩散方程，左侧固定值，右侧零法向梯度边界条件，是不应该存在方程奇异的。但是针对本例，由于其中某些网格点的扩散系数为 0，会导致方程奇异，求解稳定性发生问题。这也是本算例要演示的意义。为了对其做数值求解，可以做这样一种尝试，对于 σ 近似于 0 的区域，可以将其设置为一个非常小的值，如 1×10^{-6} 。这样，离散后的矩阵为：

$$\begin{aligned} 3\phi_1 - \phi_2 &= \phi_w, \\ -\phi_1 + 2\phi_2 - \phi_3 &= 0, \\ -\phi_2 + 2\phi_3 - \phi_4 &= 0, \\ -\phi_3 + \frac{3}{2}\phi_4 - \frac{1}{2}\phi_5 &= 0, \\ -\phi_4 + 1.000001\phi_5 - 0.000001\phi_6 &= 0, \\ -\phi_5 + 2\phi_6 - \phi_7 &= 0, \\ -0.000001\phi_6 + 1.000001\phi_7 - \phi_8 &= 0, \\ -\frac{1}{2}\phi_7 + \frac{3}{2}\phi_8 - \phi_9 &= 0, \\ -\phi_8 + 2\phi_9 - \phi_{10} &= 0, \\ \phi_{10} - \phi_9 &= G \end{aligned} \quad (\text{A.22})$$

对这些网格点组建矩阵系统有：

$$\begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 3/2 & -1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1.000001 & -0.000001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.000001 & 1.000001 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/2 & 3/2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \\ \phi_{10} \end{bmatrix} = \begin{bmatrix} \phi_w \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ G \end{bmatrix} \quad (\text{A.23})$$

其解为：

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \\ \phi_{10} \end{bmatrix} = \begin{bmatrix} 0.5\phi_w + 0.5G \\ 0.5\phi_w + 1.5G \\ 0.5\phi_w + 2.5G \\ 0.5\phi_w + 3.5G \\ 0.5\phi_w + 5.5G \\ 0.5\phi_w + 2000005.5G \\ 0.5\phi_w + 4000005.5G \\ 0.5\phi_w + 4000007.5G \\ 0.5\phi_w + 4000008.5G \\ 0.5\phi_w + 4000009.5G \end{bmatrix} \quad (\text{A.24})$$

在这里需要注意的是，一个原本奇异的矩阵系统，通过将 0 设定为一个非常小的数，暂且称之为稳定性因子，整个系统可以求解。但是，同时可以发现，矩阵的解，取决于用户自行设定的稳定性因子。本质上，这种方程是不显示的。实际操作中，最好通过多域求解来进行。

参考文献

- [1] About fvc::reconstruct (1). <https://www.jianguoyun.com/p/DeD70AkQ9s3ZBhiAtPQC>. Accessed: 2020-03-12.
- [2] About fvc::reconstruct (2). <https://www.jianguoyun.com/p/DVqidAUQ9s3ZBhiBtPQC>. Accessed: 2020-03-12.
- [3] Change of variables in multiple integrals (jacobians). [https://math.libretexts.org/Courses/Monroe_Community_College/MTH_212_Calculus_III/Chapter_14%3A_Multiple_Integration/14.7%3A_Change_of_Variables_in_Multiple_Integrals_\(Jacobians\)](https://math.libretexts.org/Courses/Monroe_Community_College/MTH_212_Calculus_III/Chapter_14%3A_Multiple_Integration/14.7%3A_Change_of_Variables_in_Multiple_Integrals_(Jacobians)). Accessed: 2022-09-20.
- [4] The fourier coefficients. <http://www.thefouriertransform.com/series/coefficients.php>. Accessed: 2020-12-05.
- [5] Is there a way to order nodes in a mesh to efficiently compute sparse matrices? <https://www.cfd-online.com/Forums/main/196509-there-way-order-nodes-mesh-efficiently-compute-sparse-matrices.html>. Accessed: 2022-01-03.
- [6] Multigrid method. https://en.wikipedia.org/wiki/Multigrid_method. Accessed: 2021-02-7.
- [7] Numerator layout. https://en.wikipedia.org/wiki/Matrix_calculus#Layout_conventions. Accessed: 2022-06-15.
- [8] Reynolds stress models. <https://www.cfd-online.com/Forums/main/1779-reynolds-stress-models.html>. Accessed: 2021-08-03.
- [9] Rotating frame of reference. <https://www.physics-in-a-nutshell.com/article/29/rotating-frame-of-reference#bibliography>. Accessed: 2021-04-13.
- [10] Turbulent boundary layer (dns). <https://www.youtube.com/watch?v=Wr984E0mNaY&t=12s>. Accessed: 2020-11-05.

- [11] Tutorial: Bandwidth reduction—the cuthill–mckee algorithm. <https://ciprian-zavoianu.blogspot.com/2009/01/project-bandwidth-reduction.html>. Accessed: 2022-01-03.
- [12] 同济大学数学系. 高等数学. 高等教育出版社, 2007.
- [13] 约翰. 安德森. 计算流体力学基础及其应用. 机械工业出版社, 2007.
- [14] 宇波. 流动与传热数值计算: 若干问题的研究与探讨. 科学出版社, 2015.
- [15] 宇波等. 数值传热学实训. 科学出版社, 2018.
- [16] R. Beetstra, M.A. van der Hoef, and J.A.M. Kuipers. Drag force of intermediate reynolds number flow past mono-and bidisperse arrays of spheres. AIChE journal, 53:489–501, 2007.
- [17] G.A. Bird. Molecular gas dynamics. NASA STI/Recon Technical Report A, 76, 1976.
- [18] J.P. Boris, F.F. Grinstein, E.S. Oran, and R.L. Kolbe. New insights into large eddy simulation. Fluid dynamics research, 10:199, 1992.
- [19] S.T. Bose, P. Moin, and D. You. Grid-independent large-eddy simulation using explicit filtering. Physics of Fluids, 22:105103, 2010.
- [20] D. Bouris and G. Bergeles. 2D LES of vortex shedding from a square cylinder. Journal of Wind Engineering and Industrial Aerodynamics, 80:31–46, 1999.
- [21] M. Breuer. Large eddy simulation of the subcritical flow past a circular cylinder: numerical and modeling aspects. International Journal For Numerical Methods in Fluids, 28:1281–1302, 1998.
- [22] W.L. Briggs, V.E. Henson, and S.F. McCormick. A multigrid tutorial. SIAM, 2000.
- [23] J. Capecelatro, O. Desjardins, and R.O. Fox. Numerical study of collisional particle dynamics in cluster-induced turbulence. Journal of Fluid Mechanics, 747, 2014.
- [24] C. Chalons, D. Kah, and M. Massot. Beyond pressureless gas dynamics: quadrature-based velocity moment models. arXiv preprint arXiv:1011.2974, 2010.
- [25] P.Y. Chou. On velocity correlations and the solutions of the equations of turbulent fluctuation. Quarterly of Applied Mathematics, 3:38–54, 1945.
- [26] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In Proceedings of the 1969 24th national conference, pages 157–172, 1969.

- [27] M.S. Darwish and F. Moukalled. Tvd schemes for unstructured grids. *International Journal of heat and mass transfer*, 46:599–611, 2003.
- [28] J.J. Derksen and S. Sundaresan. Direct numerical simulations of dense suspensions: wave instabilities in liquid-fluidized beds. *Journal of Fluid Mechanics*, 587:303–336, 2007.
- [29] J. Ding and D. Gidaspow. A bubbling fluidization model using kinetic theory of granular flow. *AIChE Journal*, 36:523–538, 1990.
- [30] M.S. Dodd and A. Ferrante. On the interaction of Taylor length scale size droplets and isotropic turbulence. *Journal of Fluid Mechanics*, 806:356–412, 2016.
- [31] D.A. Drew. Mathematical modeling of two-phase flow. *Annual Review of Fluid Mechanics*, 15:261–291, 1982.
- [32] P.A. Durbin. Separated flow computations with the k-epsilon-v-squared model. *AIAA journal*, 33:659–664, 1995.
- [33] J.L. Favero, A.R. Secchi, N.S.M. Cardozo, and H. Jasak. Viscoelastic fluid analysis in internal and in free surface flows using the software OpenFOAM. *Computers & Chemical Engineering*, 34:1984–1993, 2010.
- [34] J.H. Ferziger, M. Perić, and R.L. Street. *Computational methods for fluid dynamics*, volume 3. Springer, 2002.
- [35] R.O. Fox. Large-eddy-simulation tools for multiphase flows. *Annual Review of Fluid Mechanics*, 44:47–76, 2012.
- [36] C. Fureby and F.F. Grinstein. Monotonically integrated large eddy simulation of free shear flows. *AIAA Journal*, 37:544–556, 1999.
- [37] Z. Gao, D. Li, A. Buffo, W. Podgórska, and D.L. Marchisio. Simulation of droplet breakage in turbulent liquid–liquid dispersions with CFD-PBM: Comparison of breakage kernels. *Chemical Engineering Science*, 142:277–288, 2016.
- [38] M. Germano, U. Piomelli, P. Moin, and W.H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3:1760–1765, 1991.
- [39] S. Ghosal and P. Moin. The basic equations for the large eddy simulation of turbulent flows in complex geometry. *Journal of Computational Physics*, 118:24–37, 1995.

- [40] S. Godunov and I. Bohachevsky. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematičeskij sbornik*, 47:271–306, 1959.
- [41] G.W. Griffiths. *Numerical Analysis Using R*. Cambridge University Press, 2016.
- [42] L. Han, S. Gong, Y. Li, Q. Ai, H. Luo, Z. Liu, and Y. Liu. A novel theoretical model of breakage rate and daughter size distribution for droplet in turbulent flows. *Chemical Engineering Science*, 102:186–199, 2013.
- [43] A. Hellsten. Some improvements in Menter’s k-omega SST turbulence model. In *29th AIAA, Fluid Dynamics Conference*, page 2554, 1998.
- [44] C.W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of computational physics*, 39:201–225, 1981.
- [45] M. Ishii and S. Kim. Development of one-group and two-group interfacial area transport equation. *Nuclear Science and Engineering*, 146:257–273, 2004.
- [46] H. Jasak. Error analysis and estimation for the finite volume method with applications to fluid flows. 1996.
- [47] P.L.C. Lage. On the representation of QMOM as a weighted-residual method—The dual-quadrature method of generalized moments. *Computers and Chemical Engineering*, 35:2186–2203, 2011.
- [48] C.K.G. Lam and K. Bremhorst. A modified form of the k- ε model for predicting wall turbulence. *Journal of Fluid Engineering*, 103:456–460, 1981.
- [49] B.E. Launder, G.J. Reece, and W. Rodi. Progress in the development of a reynolds-stress turbulence closure. *Journal of fluid mechanics*, 68:537–566, 1975.
- [50] B.E. Launder and B.I. Sharma. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Letters In Heat And Mass Transfer*, 1:131–137, 1974.
- [51] B.E. Launder and D.B. Spalding. The numerical computation of turbulent flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pages 96–116. Elsevier, 1983.
- [52] B.P. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer methods in applied mechanics and engineering*, 19:59–98, 1979.

- [53] R.J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge university press, 2002.
- [54] D. Li, A. Buffo, W. Podgórska, D.L. Marchisio, and Z. Gao. Investigation of droplet breakup in liquid-liquid dispersions by CFD-PBM simulations: The influence of the surfactant type. *Chinese Journal of Chemical Engineering*, 25:1369–1380, 2017.
- [55] D. Li, Z. Gao, A. Buffo, W. Podgorska, and D. Marchisio. Droplet breakage and coalescence in liquid-liquid dispersions: Comparison of different kernels with EQMOM and QMOM. *AIChE Journal*, 63:2293–2311, 2017.
- [56] D. Li, Z. Li, and Z. Gao. Compressibility induced bubble size variation in bubble column reactors: Simulations by the CFD-PBE. *Chinese Journal of Chemical Engineering*, 26(10):2009–2013, 2018.
- [57] D. Li, Z. Li, and Z. Gao. Quadrature-based moment methods for the population balance equation: An algorithm review. *Chinese Journal of Chemical Engineering*, 27:483–500, 2019.
- [58] D. Li, D. Marchisio, C. Hasse, and D. Lucas. Comparison of Eulerian QBMM and classical Eulerian–Eulerian method for the simulation of polydisperse bubbly flows. *AIChE Journal*, 65:e16732, 2019.
- [59] D. Li, D. Marchisio, C. Hasse, and D. Lucas. twoWayGPBEFoam: An open-source Eulerian QBMM solver for monokinetic bubbly flows. *Computer Physics Communications*, page in press, 2019.
- [60] D. Li, Y. Wei, and D. Marchisio. QEEFoam: A Quasi-Eulerian-Eulerian model for polydisperse turbulent gas-liquid flows. Implementation in OpenFOAM, verification and validation. *International Journal of Multiphase Flow*, 136:103544, 2021.
- [61] J. Li and W. Huang. From multiscale to mesoscience: addressing mesoscales in mesoregimes of different levels. *Annual review of chemical and biomolecular engineering*, 9:41–60, 2018.
- [62] F.S. Lien and G. Kalitzin. Computations of transonic flow with the v2-f turbulence model. *International Journal of Heat and Fluid Flow*, 22:53–61, 2001.
- [63] Z. Liu and B. Li. Scale-adaptive analysis of euler-euler large eddy simulation for laboratory scale dispersed bubbly flows. *Chemical Engineering Journal*, 338:465–477, 2018.

- [64] N.K. Madavan, C.L. Merkle, and S. Deutsch. Numerical investigations into the mechanisms of microbubble drag reduction. *Journal of Fluids Engineering*, 107:370–377, 1985.
- [65] D.L. Marchisio and R.O. Fox. *Computational models for polydisperse particulate and multiphase systems*. Cambridge University Press, 2013.
- [66] F. Menter. Zonal two equation k-w turbulence models for aerodynamic flows. In 23rd fluid dynamics, plasmadynamics, and lasers conference, page 2906, 1993.
- [67] F.R. Menter, M. Kuntz, and R. Langtry. Ten years of industrial experience with the SST turbulence model. *Turbulence, Heat And Mass Transfer*, 4:625–632, 2003.
- [68] M. Milelli, B.L. Smith, and D. Lakehal. Large-eddy simulation of turbulent shear flows laden with bubbles. In Direct and Large-Eddy Simulation IV, pages 461–470. Springer, 2001.
- [69] K.W. Morton and D. F. Mayers. *Numerical solution of partial differential equations: an introduction*. Cambridge university press, 2005.
- [70] A. Ozel, J. Kolehmainen, S. Radl, and S. Sundaresan. Fluid and particle coarsening of drag force for discrete-parcel approach. *Chemical engineering science*, 155:258–267, 2016.
- [71] N. Panicker, A. Passalacqua, and R.O. Fox. On the hyperbolicity of the two-fluid model for gas–liquid bubbly flows. *Applied Mathematical Modelling*, 57:432–447, 2018.
- [72] R. Picardi, L. Zhao, and F. Battaglia. On the ideal grid resolution for two-dimensional eulerian modeling of gas–liquid flows. *Journal of Fluids Engineering*, 138:114503, 2016.
- [73] D. Ramkrishna. *Population balances: Theory and applications to particulate systems in engineering*. Academic Press, 2000.
- [74] P.L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.
- [75] J. Roenby, H. Bredmose, and H. Jasak. A computational method for sharp interface advection. *Royal Society open science*, 3:160405, 2016.
- [76] I. Roghair, M.V.S. Annaland, and J.A.M. Kuipers. An improved Front-Tracking technique for the simulation of mass transfer in dense bubbly flows. *Chemical Engineering Science*, 152:351–369, 2016.

- [77] E. Roohi, A.P. Zahiri, and M. Passandideh-Fard. Numerical simulation of cavitation around a two-dimensional hydrofoil using VOF method and LES turbulence model. *Applied Mathematical Modelling*, 37:6469–6488, 2013.
- [78] T.H. Shih. *A realizable Reynolds stress algebraic equation model*, volume 105993. National Aeronautics and Space Administration, 1993.
- [79] T.H. Shih, W.W. Liou, A. Shabbir, Z. Yang, and J. Zhu. A new k-epsilon eddy viscosity model for high reynolds number turbulent flows: Model development and validation. NASA Sti/recon Technical Report N, 95:11442, 1994.
- [80] S.W. Sloan. An algorithm for profile and wavefront reduction of sparse matrices. *International Journal for Numerical Methods in Engineering*, 23:239–251, 1986.
- [81] P.R. Spalart. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. In *Proceedings of first AFOSR international conference on DNS/LES*. Greyden Press, 1997.
- [82] C.G. Speziale and S. Thangam. Analysis of an RNG based turbulence model for separated flows. *International Journal of Engineering Science*, 30:1379–1389, 1992.
- [83] A.I. Stamou and G. Papadonikolaki. Modeling the 3-D flow around a cylinder using the SAS hybrid model. *Global Nest Journal*, 16:901–918, 2014.
- [84] H. Struchtrup. *Macroscopic transport equations for rarefied gas flows*. Springer, 2005.
- [85] S. Sundaresan, A. Ozel, and J. Kolehmainen. Toward constitutive models for momentum, species, and energy transport in gas-particle flows. *Annual review of chemical and biomolecular engineering*, 9:72–85, 2018.
- [86] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114:146–159, 1994.
- [87] S. Tenneti, R. Garg, and S. Subramaniam. Drag law for monodisperse gas-solid systems using particle-resolved direct numerical simulation of flow past fixed assemblies of spheres. *International journal of multiphase flow*, 37:1072–1092, 2011.
- [88] E.F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.
- [89] M. Torrilhon. Modeling nonequilibrium gas flow based on moment equations. *Annual review of fluid mechanics*, 48:429–458, 2016.

- [90] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.J. Jan. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169:708–759, 2001.
- [91] C. Tsouris and L.L. Tavlarides. Breakage and coalescence models for drops in turbulent dispersions. *AIChE Journal*, 40:395–406, 1994.
- [92] O. Ubbink. Numerical prediction of two fluid systems with sharp interfaces. PhD thesis, University of London, 1997.
- [93] Salih Ozen Unverdi and Grétar Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of computational physics*, 100:25–37, 1992.
- [94] H.K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson education, 2007.
- [95] D.C. Wilcox. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA journal*, 26:1299–1310, 1988.
- [96] D.C. Wilcox. *Turbulence modeling for CFD*. DCW industries La Canada, CA, 1998.
- [97] K. Xu. A gas-kinetic BGK scheme for the Navier–Stokes equations and its connection with artificial dissipation and Godunov method. *Journal of Computational Physics*, 171:289–335, 2001.
- [98] V. Yakhot, S.A. Orszag, S. Thangam, T.B. Gatski, and C.G. Speziale. Development of turbulence models for shear flows by a double expansion technique. *Physics of Fluids A: Fluid Dynamics*, 4:1510–1520, 1992.
- [99] M.E. Young and A. Ooi. Comparative assessment of LES and URANS for flow over a cylinder at a Reynolds number of 3900. In *Proceedings of the 16th Australian Fluid Mechanics Conference*, Gold Coast, Australia, 1063-1070, 2007.
- [100] S. Zhang and X. Zhao. General formulations for Rhie-Chow interpolation. In *Heat Transfer Summer Conference*, volume 46911, pages 567–573, 2004.
- [101] S. Zhang, X. Zhao, and S. Bayyuk. Generalized formulations for the Rhie–Chow interpolation. *Journal of Computational Physics*, 258:880–914, 2014.