

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

THESIS OF BACHELOR



论文题目： 三维风机流场与气动性能计算

学生姓名： 王 强

学生学号： 5080109162

专 业： 船舶与海洋工程

指导教师： 万德成 教授

学院(系)： 船舶海洋与建筑工程学院

上海交通大学

本科生毕业设计（论文）任务书

课题名称： 三维风机流场与气动性能计算

执行时间： 2012 年 2 月 至 2012 年 6 月

教师姓名： 万德成 职称： 教授

学生姓名： 王强 学号： 5080109162

专业名称： 船舶与海洋工程

学院(系)： 船舶海洋与建筑工程学院

毕业设计（论文）基本内容和要求：

本设计需要结合本科所学知识，尤其是流体力学知识，通过进一步的调研、文献阅读和学习，最终完成三维风机的流场和气动性能计算。关键是通过计算分析掌握风机的流场与气动性能，为风机设计提供理论指导。其中包括以下内容：

- 1、熟悉 Linux 操作系统，学习 OpenFOAM 数值计算软件；
- 2、深入调研风力发电的背景，风机流场与气动性能的国内外研究现状。
通过广泛的文献阅读明确问题与解决思路；
- 3、学习并使用 CATIA 软件对风机进行三维建模；
- 4、学习并使用 SnappyHexMesh 方法对计算模型进行网格划分；
- 5、学习动网格技术（Generalized Grid Interface）技术；
- 6、使用 OpenFOAM 对风机流场与气动性能进行数值计算；
- 7、整理计算结果，与相关的模型实验进行广泛比较分析；
- 8、（时间充裕可选）对两台前后排列的风机一齐进行数值模拟，预报尾流对风机空气动力性能的影响；
- 9、整理并完成论文。

毕业设计（论文）可用资料：

美国国家可再生能源实验室 NREL Phase VI 风机风洞实验数据。

毕业设计（论文）进度安排:

序号	毕业设计（论文）各阶段内容	时间安排	备注
1	熟悉 Lunix 操作系统与 OpenFOAM	1.15~2.14	已完成
2	阅读文献，了解研究背景与现状	2.15~3.5	
3	建立风机三维模型并划分网格	3.6~3.20	
4	风机流场与气动性能计算	3.21~4.20	
5	整理数据，与实验值比较	4.21~5.8	
6	论文整理	5.9~5.28	

课题信息:

课题性质 : 设计 论文 √

课题来源*: 国家级 √ 省部级 校级 横向 预研

项目编号 11072154

其他_____

指导教师签名: _____

年 月 日

学院（系）意见:

院长（系主任）签名: _____

年 月 日

学生签名: _____

年 月 日

上海交通大学

毕业设计（论文）学术诚信声明

本人郑重声明：所呈交的毕业设计（论文），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日期： 年 月 日

上海交通大学

毕业设计（论文）版权使用授权书

本毕业设计（论文）作者同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业设计（论文）。

保密□，在____年解密后适用本授权书。

本论文属于

不保密□。

(请在以上方框内打“√”)

作者签名:

指导教师签名:

日期: 年 月 日

日期: 年 月 日

三维风机流场与气动性能计算

摘要

本文基于开源 CFD 计算平台 OpenFOAM 对美国国家可再生能源实验室 NREL phase VI 风机进行数值计算。计算选用瞬态不可压雷诺平均方程(Reynolds-Averaged Navier-Stokes, RANS)求解器 pimpleDyMFoam，并采用任意网格界面元法(Arbitrary Mesh Interface, AMI)处理网格的相对运动。本文计算了单个风轮和整台风机两个计算模型在固定转速不同风速(5, 10, 15 和 25m/s)固定桨距角(pitch angle) 3°的情况下三维流场与气动性能。研究得出了不同风速下风机的推力和功率曲线，叶片不同位置处剖面的法向力系数、切向力系数和压力系数分布，风机周围流线分布、速度场及涡的分布等。计算结果显示风机周围流场有着非常显著的三维效应，整机模型捕捉到了比单个风轮更加多的流场特征，如叶片和塔筒的相互作用等。计算结果与实验整体吻合较好，展现了 pimpleDyMFoam 求解器在处理三维风机复杂流场的巨大潜力，为后续将求解器运用到更加复杂的问题上奠定了基础，如海上浮式风机等。本文也为风机设计尤其是叶片设计提供了有价值的参考。

关键词：风机流场，风机气动性能，NREL phase VI，OpenFOAM，AMI，RANS，叶片与塔筒相互作用

NUMERICAL STUDY OF 3D FLOWS AND AERODYNAMIC PERFORMANCE OF WIND TURBINE

ABSTRACT

Simulations of the National Renewable Energy Laboratory (NREL) phase VI wind turbine based on the open source CFD platform OpenFOAM are presented. The simulations are performed using unsteady Reynolds-Averaged Navier-Stokes (RANS) code pimpleDyMFoam, and Arbitrary Mesh Interface (AMI) is adopted to handle mesh movements. Computation cases include different wind speed (5, 10, 15, and 25m/s) at a fixed blade pitch angle of 3°with constant rotational speed, including a rotor only model and a whole wind turbine. Extensive comparison against experimental results, including total power and thrust, local pressure coefficient and sectional performance of normal and tangential force coefficient, shows good predictions. Besides, visualized 3D effects of flows are obtained, including streamlines around the wind turbine, velocity field and vorticity field. More detailed 3D effects of flows are observed on the whole wind turbine model, including blade-tower interaction. The work shows a great potential of pimpleDyMFoam in dealing with complex 3D flows of wind turbine which lays a solid foundation for more complex computations, such as floating offshore wind turbines. The work also offers valuable guides for wind turbine design.

Key words: wind turbine flows, wind turbine aerodynamic, NREL phase VI, OpenFOAM, AMI, RANS, blade-tower interaction

目 录

第一章 绪论	1
1.1 风能综述	1
1.1.1 世界风能概况	1
1.1.2 国内风能概况	1
1.1.3 风能开发问题挑战	2
1.2 风机气动性能主要研究方法	3
1.2.1 叶素-动量理论	3
1.2.2 三维势流模型	3
1.2.3 三维粘性流模型	3
1.3 风机气动性能 CFD 研究现状	4
1.3.1 NREL 非定常空气动力学实验	4
1.3.2 风机气动性能 CFD 研究现状	4
1.4 开源软件 OpenFOAM 简介	6
1.5 本文主要工作	6
第二章 数学建模与数值方法	8
2.1 控制方程	8
2.1.1 连续介质力学的控制方程	8
2.1.2 Navier-Stokes 方程	8
2.2 湍流模拟	9
2.2.1 常见湍流模拟方法	9
2.2.2 RANS 方程湍流模型	10
2.2.3 近壁面处理	13
2.3 本章小结	14
第三章 有限体积法	15
3.1 计算域的离散	15
3.2 输运方程的离散	16
3.2.1 对流项的离散	17
3.2.2 粘性耗散项的离散	19
3.2.3 源项的离散	21
3.2.4 时间导数项的离散	21
3.3 Navier-Stokes 方程离散	24
3.3.1 动量方程的非线性处理	24
3.3.2 压力速度的耦合处理	24
3.3.3 Navier-Stokes 方程求解流程	26
3.4 本章小结	26
第四章 OpenFOAM 相关方法介绍	27
4.1 OpenFOAM 中偏微分方程的表达	27
4.2 snappyHexMesh 自动网格划分介绍	27
4.3 topoSet 网格操作介绍	29

4.4 Arbitrary Mesh Interface 动网格处理介绍	30
4.5 pimpleDyMFoam 求解器	31
4.6 本章小结	32
第五章 NREL Phase VI 单个风轮数值计算	33
5.1 建立几何模型	33
5.1.1 叶片模型描述	33
5.1.2 风轮模型建立过程	35
5.2 网格绘制	39
5.2.1 计算域确定	39
5.2.2 纯 snappyHexMesh 划分网格	39
5.2.3 ICEM-CFD 和 snappyHexMesh 联合绘制网格	42
5.3 初始条件、边界条件设置	46
5.4 算例结构与求解参数设置	47
5.4.1 算例结构与求解参数设置	47
5.4.2 算例运行流程	48
5.5 结果与讨论	49
5.5.1 涡、速度场和各截面流线分布	49
5.5.2 推力和转矩	54
5.5.3 C _n 和 C _t	54
5.5.4 C _p	56
5.5.5 关于计算误差的进一步讨论	59
5.6 本章小结	60
第六章 NREL Phase VI 整机模型数值计算	61
6.1 建立几何模型	61
6.1.1 建立轮毂和机舱模型	61
6.1.2 建立塔筒模型	62
6.2 网格绘制	62
6.3 其他参数设置	64
6.4 结果与讨论	64
6.4.1 涡	64
6.4.2 推力	67
6.4.3 C _p	67
6.4.4 塔筒压力分布	67
6.5 本章小结	71
第七章 结语	72
7.1 总结	72
7.2 展望	72
参考文献	73
谢辞	75

第一章 绪论

1.1 风能综述

进入 21 世纪后，全球面临着“两个空前的危机”，一个是化石能源的日益枯竭，另一个是全球气候变暖和环境的持续恶化。为了应对这两个危机，世界许多国家都兴起了开发可再生能源的热潮。风能作为一种储量丰富，清洁并相对廉价可再生能源，愈发受到世界各国的青睐。据估算，全球可以利用的陆上和近岸风能总量达 72TW，是当下世界能源消耗总量的五倍多^[1]。

1.1.1 世界风能概况

早在两千年前，人类就开始利用风能。在这之后的一千多年时间里，风磨，风车等开始广为应用，但主要都是用于农业抽水和碾磨。直到 1887 年，英国人 James Blyth 建造了第一台真正意义上用于发电的风机，用于给自己房子的灯泡供电^[2]。十九世纪 90 年代，丹麦科学家 Poul la Cour 第一个提出了转速越快，叶片数量越少的风机风能利用效率越高的理论，对风机发展产生重大影响。20 世纪前半叶，化石能源的大量使用降低了人们对于风能的开发热情。1956 年，Poul la Cour 的学生 Johannes Juul^[3]，在丹麦建造了著名的 200kW 三叶片风机 Gedser mill，对后来风机的发展产生了重大影响。1973 年能源危机之后，美国政府在风机研发上大力投入，先后建造了 13 台试验风机，对风电技术产生了里程碑式的影响^[4]。从那之后，风机开始走向大型化，风电也开始在全世界范围内兴起。

进入 21 世纪，风电的发展更是加快了步伐。世界风电累计装机容量在 2000 至 2010 年间年均增长达到 28%。即便在 2011 年各行业遭受了世界金融和欧洲债务危机恶劣影响的情况下，全球的风电新装机容量依旧达到了 40.5GW，累计装机容量达到 238GW，比 2010 年增长了 20%^[5]。



图 1-1 全球累计风电装机容量 1996-2011^[5]

预计到 2013 年，风电占全球电力消耗比例将达到 3.35%，而到 2018 年这个比例将达到 8%^[6]。风电在欧洲一些国家的全国电力消耗上已经占到了相当大的比重。以 2011 年统计数据排序^[2]，前三名分别为丹麦（28%）、葡萄牙（19%）、西班牙（16%）。这些数据表现出风电在未来发展上还有着非常巨大的潜力。据全球风能理事会(Global Wind Energy Council, GWEC)^[5]预计，全球 2016 年的风电总装机容量将由 1012 年的 238GW 达到 500GW，年均增幅 16%。

1.1.2 国内风能概况

在全球风电开发迎来热潮的时候，中国风电的发展速度也让世界瞩目。中国广袤的大

陆和绵长的海岸线为它提供了丰富的风电资源。根据第三次全国风能资源普查^[3]，中国可开发的海上风能储量约为 2580GW。我国风电行业虽然起步较晚，但是发展十分迅速。而随着 2006 至 2009 风电总装机容量连续四年的翻倍增长，至 2010 年中国风电总装机容量已经达到 42.2GW，位列世界第一。而 2011 年新增装机容量 20GW 使总装机容量达到 62GW，继续保持世界第一的席位，风电占全国电力消耗比例达到 1.5%^[5]。

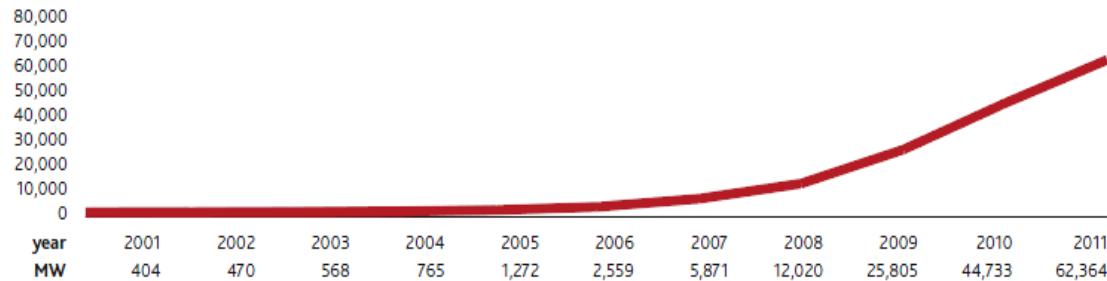


图 1-2 中国风电总装机容量^[5]

截止 2011 年，中国已经有 13 个省份的总装机容量超过 1GW。除了下表所列的前十个省份外，还有山西（1881.1MW）、广东（1302.4MW）和福建（1025.7MW）^[5]。

	2010 Cumulative (MW)	2011 Annual (MW)	2011 Cumulative (MW)
1	Inner Mongolia	13,858.0	3,736.4
2	He Bei	4,794.0	2,175.5
3	Gansu	4,944.0	465.2
4	Liao Ning	4,066.9	1,182.5
5	Shan Dong	2,637.8	1,924.5
6	Ji Lin	2,940.9	622.5
7	Hei Longjiang	2,370.1	1,075.8
8	Ning Xia	1,182.7	1,703.5
9	Xin Jiang	1,363.6	952.5
10	Jiang Su	1,595.3	372.3

图 1-3 中国风电装机容量排名前十省份^[5]

2011 年是“十二五”开局之年，根据国家能源局 12 月 15 日公布的“十二五”可再生能源规划目标，到 2015 年时，中国风电的总装机容量将达到 100GW，其中 70GW 来自大型风电项目，30GW 来自小型风电项目，也包含了 5GW 的海上风电项目。

虽然中国的风电行业依旧存在着各种问题与挑战，但是可以预见，未来中国的风电依旧有着十分巨大的潜力。

1.1.3 风能开发问题挑战

风能开发过程中需要面对的挑战很多，涉及很多领域。比如风资源的准确评估，大规模风电并网问题，减低风电成本并提高风机可靠度等等。任何一个问题都能够最终影响到整个风场投资的效益。而整个风场的效益，很大程度上取决于每一台风机捕获风能的效率的高低。而风机的风能捕获效率涉及到风机设计研发过程中的两个关键问题，一是控制系统设计问题，另一个就是风机的空气动力性能问题。本文主要关注后者。

水平轴的风机是目前用于捕获风能资源最广为应用的一种形式。一般风机都是位于大气的边界层内，所以风速是随着离开地面高度的升高而增加的，这样就产生了剪切流动^[7]。这种剪切流动导致了风速在空间和时间分布上的不均匀。具体到某一台风机叶片上某一个剖面来说，来流速度和攻角都是非定常的，因而风机的受力也将是非定常的。此外，对于风场中的风机来说，风流过前面一台风机后会旋转并产生涡，大大增加了作用到下一台风机上载荷的非定常性。因此，准确预报风机空气动力学性能和周围的流场情况对于优化风机的设计与布局，从而提高风场的风能利用率具有非常重要的作用。这也是本文工作的主要意义。

1.2 风机气动性能主要研究方法

风机叶尖速度一般最高可以达到 70~100m/s, 而声速是 340m/s, 所以对于风机周围流场, 其马赫数(Mach number)小于 0.3, 故一般可将其作为不可压流体来求解。如今国内外已经发展了各种各样的方法来预报风机的空气动力学性能。

1.2.1 叶素-动量理论

叶素-动量理论(Blade Element Moment methods, BEM)^[8]由于计算方便, 在实际工程中应用广泛。该方法类似于经典的“切片理论”, 首先将叶片分成一个个微段, 称为叶素。然后将每一个叶素都看作“鼓动盘”来计算作用在上面的载荷。最后将这些载荷叠加得到作用在整个叶片上的载荷。考虑到实际的风机叶片是一个三维物体, 人们依照该方法开发求解器时又引入了叶尖损失(tip loss)、动态失速(dynamic stall)等模型^[9, 10]。之后该方法被广泛应用到了风力机空气动力性能计算领域, 目前风能工业领域大量的求解器均是基于该方法开发而成。

由于该方法在计算时必须预先知道不同叶素的升力系数和阻力系数, 即需要得知不同雷诺数下机翼剖面在不同攻角下的这类参数, 故其应用范围和结果的准确性也直接受到了翼型数据的约束。此外, 如上所述, 叶素-动量理论需要引入经验系数来表达二维翼型的三维特性, 如叶尖损失(tip loss)、有旋流动(rotational flow)和动态失速(dynamic stall)等, 往往会有偏差。所以该方法对于预测新开发的风机叶片空气动力性能, 研究叶片附近流场细节等方面有明显的不足。

1.2.2 三维势流模型

为了获得更详细的风机空气动力特性, 并且保持计算的高效, 人们引入了三维势流空气动力学模型, 如升力线理论(lifting line)^[11]、升力面理论(panel)^[12]、尾涡理论(vortex)^[13]、边界积分方程(Boundary Integral Equation)^[14]等。虽然这些方法计算相对还算高效, 也可以比叶素-动量理论反映更多的叶片三维效应。但是, 由于这些方法是基于势流理论开发, 所以无法很好处理粘性和分离等问题。而风机周围流场往往存在很强的粘性作用和分离现象, 故限制了这些方法在风机空气动力性能邻域进一步的应用。

1.2.3 三维粘性流模型

更进一步的方法便是考虑粘性作用的计算流体力学(computational Fluid Dynamics, CFD)方法(下文出现的 CFD 均指考虑粘性作用的计算方法)。主要是通过求解雷诺平均方程(Reynolds-Averaged Navier-Stokes, RANS), 附加一些湍流模型来计算风机的空气动力性能和风机周围流场。但是运用 CFD 的方法时, 人们也会面临不小的挑战。

首先用 CFD 计算风机的流场时, 需要保证足够大的计算域, 以保证流场内部的流动少受边界的影响, 同时需要保证叶片表面的网格足够密以精确求解近壁面的流动, 所以 CFD 往往需要较大的网格量。此外, 风机叶片往往是扭曲变形非常严重的。故对风机叶片网格的划分也带来了不小的挑战。其次, 如何选用或开发合适的湍流模型来精确模拟风机叶片边界层分离和泄涡的情况也是一大挑战。最后, 如何处理计算域中有部分旋转区域的问题也是一个比较大的问题。所以这种方法较前两种来讲要更加复杂, 所需要的计算量和计算时间也远远高过前两种方法。

但是随着计算机技术的快速发展, 人们已经能够通过并行计算等方法有效地处理此类问题。CFD 方法相较与其他方法的优越性也逐渐体现出来了。它不需要预先知道任何翼型的空气动力数据, 反而可以为 BEM 等方法提供此类数据; 它还可以自动考虑三维旋转效应, 也不需要依赖任何经验公式来考虑各种三维现象带来的偏差, 如叶尖损失, 动态失速等。它也可以更加精确的捕捉到流场的细节, 描绘出涡的情形, 探讨尾涡对风机空气动力学的影响等等。

1.3 风机气动性能 CFD 研究现状

1.3.1 NREL 非定常空气动力学实验

计算流体力学方法应用到风机空气动力性能计算上历史还不是很长，而直接推动这一方法在该领域应用的事件便是美国国家可再生能源实验室(National Renewable Energy Laboratory, NREL)于 1999 年开展的大规模非定常空气动力学实验(Unsteady Aerodynamic Experiment, UAE)^[15, 16]。实验在美国国家航空航天局(National Aeronautics and Space Administration, NASA) Ames 研究中心 24.4m×36.6m 风洞中进行，实验对象是一台两叶片，直径 10m 的 20MW 风机，实验工况多达 1700 多种。实验测量了风机在各工况下的空气动力性能和结构响应，为日后的研究人员研究设计风机提供了弥足珍贵的资料。

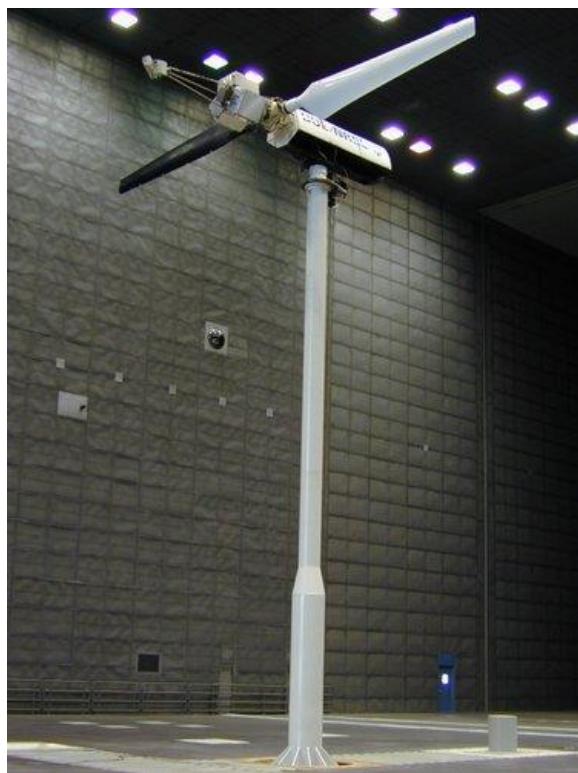


图 1-4 NREL 非定常空气动力学实验风机^[16]

实验完成后不久，实验室又在全世界范围内组织了一次盲比(blind comparison)^[17]，参加比较的求解器几乎囊括了当时在风机工业界和科研界流行的各种空气动力性能计算软件。实验结果显示，大部分求解器计算功率输出模型都有不足之处，与实验相比相差范围在 25%~175%，在大风速时偏差更是达到了 30%~275%。而其中基于 Navier-Stokes 方程的 CFD 求解器的结果与实验吻合最好。这就极大地促进了 CFD 在计算风机空气动力学性能上的应用。

1.3.2 风机气动性能 CFD 研究现状

目前，大部分针对风机的数值计算结果都会与 NREL 的非定常空气动力学实验数据对比。而其中最广为参照的结果是 Phase VI 风机实验。

(1) 国外研究现状

Sørensen^[18]等人使用基于分区有限体积法(multiblock finite volume)的不可压 RANS 求解器 EllipSys3D 对 Phase VI 风机单个叶片在不同风速下的气动性能进行了计算。计算采用了 k- ω SST 湍流模型。计算结果发现当风速超过 13m/s 后，扭矩与实验值有 20% 左右的偏差；而压力分布等都有很好的结果。这也初步证明了 CFD 方法在计算风机空气动力性能上的优

势。

Duque^[19]等人对 NREL Phase VI 风机风轮采用基于有限差分法(Finite Differences Approach) 和重叠网格(overset grid)的可压 RANS 求解器 Overflow-D 进行求解。他们将结果与势流求解器 CAMRADII 结果以及实验进行了详细的对比，讨论了风机的各项气动性能指标，如轴功率、轴向力和压力系数等，得出 RANS 求解器相较于势流求解器有更高的准确性。

Potsdam 和 Mavriplis^[20]采用非结构化分区 RANS 求解器 NSU3D 对一个孤立叶片的气动性能进行了计算，并与实验和 Overflow 软件计算值进行了对比。

Li^[21]等采用基于有限差分和重叠网格的 RANS 和大涡模拟(Detached Eddy Simulation, DES)求解器 CFDIowa-ship v4.5 对 NREL Phase VI 整个风机进行了建模求解。计算采用 $k-\omega$ SST 湍流模型，得到了不同风速下风机的空气动力性能，并研究了风机不同桨距角(pitch angle)对其空气动力学性能的影响，与实验结果有很好的吻合。

Digraskar^[22]使用基于有限体积法的 CFD 平台 OpenFOAM 对该风机进行了数值计算，采用包括基于多参考系(multi reference frames)稳态 RANS 求解器 MRFSimpleFoam 和基于通用网格界面(General Grid Interface, GGI)的瞬态 RANS 求解器 TurbDyMFoam 的求解。作者将二者结果与 NREL 的实验进行了对比，但是结果上有较大偏差。

(2) 国内研究现状

由于国内对风机空气动力性能研究起步较晚，加上几乎没有独立开发的求解器平台，对于这方面的研究不是很多，大部分的工作都是基于商用软件 Fluent。

张义华^[23]使用 Fluent 中旋转坐标系方法对 NREL Phase VI 风轮单个叶片在不同风速下的空气动力性能进行了计算，得到的扭矩与实验值有 30% 左右偏差。他还探讨了不同桨距角对风机空气动力性能的影响，后者没有与实验对比。

俞国华，杜朝辉^[24]使用 Fluent 对 NREL 单个叶片进行了计算，采用 $k-\omega$ SST 湍流模型，计算结果与实验较为接近。

王强^[25]使用 Fluent 中旋转坐标系和滑移网格两种方法对 NREL Phase VI 风机进行了计算，采用 $k-\omega$ SST 湍流模型，计算得到的扭矩与实验值也有 30% 左右偏差，加密网格后有所改善。其他结果与实验有较好的匹配。

(3) 国内外研究现状小结

通过上面的叙述，我们可以看到：

① 国外对于风机的空气动力性能研究起步比较早，而其已经做了很多工作，也开发了很多各具特色的求解平台。而国内对该领域研究工作近几年才起步，基本没有形成独立开发的软件平台，大部分工作都是基于商用软件开展，如 Fluent。

② 大部分研究对象都是仅有风轮结构的简化后风机，这其中又有绝大多数是使用单个叶片加周期性边界来代表整个风机，而排除了轮毂、机舱和塔筒对于结果的影响。这样可以很大程度上减小计算的复杂性与计算量，但是也可能会遗漏一些重要的物理现象。

③ 大部分研究工作都采用了 $k-\omega$ SST 湍流模型，其计算结果也基本验证了该模型在风机空气动力性能计算上的适用性。

④ CFD 的计算结果与实验有较好的吻合，但是也还存在很多不尽如人意的地方，其原因也有待进一步探讨。

⑤ 新出现的开源 CFD 计算平台 OpenFOAM 在风机空气动力性能计算上的能力还有待进一步研究与验证。

1.4 开源软件 OpenFOAM 简介

近十几年来，计算机的计算性能迅速提高，这为软件业带来空前繁荣。八十年代出现了以 Fluent 为代表的商业 CFD 软件市场后，CFD 软件在工业领域的应用正快速发展。然而在商业 CFD 软件占据广泛市场的同时，开源 CFD 软件也在迅速成长，这主要是源于商业 CFD 的高度封装性使得用户无法接触其内部代码，尽管在物理模型离散格式和算法方面它们为用户提供了一定的选择空间，但是用户必须在购买维护升级培训方面花上可观的费用，且并不能在计算模拟结果的可靠性上获得保障。而在科研院校这边，开源软件源代码的开放大大促进了技术交流和学术合作，也带来了软件自身的迅速扩展升级，跟商业软件相比有后来居上的发展趋势，其中 OpenFOAM 就是在这样的背景下成长起来的，且越来越受到研究院校和工业界的青睐^[26]。

OpenFOAM^[27]全称Open Field Operation and Manipulation，即场的运算和操作的开源平台的意思。它实质是一个基于有限体积法，可用于对连续介质力学问题进行数值计算的面向对象的C++库。OpenFOAM的前身为FOAM(Field Operation and Manipulation)^[28]，后来作为开源代码被公布到网上，并改名为OpenFOAM。第一版于2004年年底发布，经过7年时间的发展，目前最新版本为2011年年底发布的OpenFOAM-2.1.0。

OpenFOAM的优越性有如下几点：^[29, 30]

- (1) 它是最早利用 C++语言编写而成的科学软件包之一，由于采用了面向对象编程技术，新模型的加入变得轻松自如，改变了商业软件修改困难的问题，极大地方便了用户对软件的自由修改；
- (2) 它利用C++的运算符重载功能使得顶层代码在对偏微分方程的描述上十分简单且可读性较强，这使得OpenFOAM成为了一种十分适用于描述物理问题的编程语言；
- (3) 它是最早采用多面体单元网格的通用CFD软件包，而这个功能得以实现是源于对模拟对象采用分层描述的自然结果；
- (4) 它几乎所有的操作（网格划分、求解计算、前后处理）都可以简单实现并行计算，为用户节省大量时间；
- (5) 它是目前发布于开源许可下的最强大的通用 CFD 软件包。

总体来说，OpenFOAM为计算流体力学领域的科研人员提供了一个强大的平台。基于OpenFOAM的研究和开发工作正在欧美国家的科研院校和科技公司如火如荼的展开，相关的交流活动和社区活动也已经在欧美地区火热地开展^[31]。以这样的发展速度预计，未来不久，开源CFD软件就可以与商业软件相媲美了。但是与使用已经开发完全的商业软件相比，使用开源CFD软件需要付出的努力也将更大，尤其是开源软件往往意味着更少的技术支持，相关指导手册与书籍的缺乏以及更加困难的数值编程工作。因此，使用开源软件需要用户自己更多地去搜寻探索已有的资源，也促使他们更多地去分享经验。这也正是开源软件迅速发展的核心推动力之一。

1.5 本文主要工作

本文致力于采用开源 CFD 计算平台 OpenFOAM 对美国国家可再生能源实验室 NREL Phase VI 风机进行数值计算。本文探讨采用 OpenFOAM 开发的 snappyHexMesh 工具对风机及流场进行六面体网格划分。计算选用瞬态不可压 RANS 求解器 pimpleDyMFoam，并采用 OpenFOAM-2.1.0 版本新推出的任意网格界面元法(Arbitrary Mesh Interface, AMI)来处理网格的相对运动。通过这次工作，本文试图对 snappyHexMesh 工具在对高度扭曲的复杂几何体的网格划分能力做出探索；对 pimpleDyMFoam 求解器以及 AMI 方法在处理计算域内部有相对运动问题上的能力做出探索。

本文的主要计算单个风轮和整个风机两种模型在固定转速固定桨距角不同风速下的空气动力学性能。

研究主要针对 NREL Phase VI 实验中的 Sequence S 系列。研究内容包括：

- (1) 相同转速相同桨距角不同风速下风机的空气动力学性能。包括不同风速下风机的推力和功率曲线，叶片不同位置处剖面的压力系数、切向力系数和法向力系数分布，以及风机周围流线分布、速度场及涡的分布等，深入探讨风机的三维现象。
- (2) 深入比较两种模型，研究轮毂、机舱、塔筒等结构对于风机流场及其空气动力学性能的影响。

第二章 数学建模与数值方法

2.1 控制方程

2.1.1 连续介质力学的控制方程

对于连续介质中任意物理量G都满足下列物质体积和控制体体积的关系式，即雷诺输运定理(Reynolds Transport Theorem):

$$\frac{d}{dt} \iiint_{MV} G dV = \frac{\partial}{\partial t} \iiint_{CV} G dV + \iint_{CS} G \mathbf{U} \cdot \mathbf{n} dA \quad (1-1)$$

其中：MV——material volume，该积分项为物质体体积分；

CV——control volume，该积分项表示控制体体积分；

CS——control surface，该积分项为控制体面积分。

\mathbf{U} ——速度矢量；

\mathbf{n} ——控制体表面的法向矢量；

A——控制体表面面积。

通过对G用不同的物理量代换，我们可以得到下列形式方程：

(1) 连续性方程 (质量守恒方程, $G=\rho$)

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (1-2)$$

(2) 动量方程 (动量守恒定律, $G=\rho \mathbf{U}$)

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = \rho g + \nabla \cdot \boldsymbol{\sigma} \quad (1-3)$$

其中： $\boldsymbol{\sigma}$ ——应力张量；

上面的方程适用于任何流体，但是方程个数小于未知数个数，因此还无法求解。

2.1.2 Navier-Stokes 方程

为了使方程组封闭，我们必须引入额外的关系式。对于牛顿流体而言，它必定遵循以下本构方程(constitutive equation):

$$\boldsymbol{\sigma} = -(P + \frac{2}{3} \mu \nabla \cdot \mathbf{U}) \mathbf{I} + \mu [\nabla \mathbf{U} + (\nabla \mathbf{U})^T] \quad (1-4)$$

其中：P——压强；

μ ——流体动力粘度；

\mathbf{I} ——单位矩阵；

将上式代入(2-3)(2-4)中即可得到封闭的偏微分方程组，即Navier-Stokes方程：

(1) 连续性方程

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (1-5)$$

(2) 动量方程

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = \rho g - \nabla \cdot (P \mathbf{I} + \frac{2}{3} \mu \nabla \cdot \mathbf{U} \mathbf{I}) + \nabla \cdot [\mu (\nabla \mathbf{U} + (\nabla \mathbf{U})^T)] \quad (1-6)$$

对于不可压流体，我们有：

$$\frac{D\rho}{Dt} = 0 \quad (1-7)$$

所以不可压流体的Navier-Stokes方程，可简化为下式：

$$\nabla \cdot \bar{\mathbf{U}} = 0 \quad (1-8)$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U} \mathbf{U}) = g - \nabla p + \nabla \cdot (\nu \nabla \mathbf{U}) \quad (1-9)$$

其中：p——动压， $p=P/\rho$ ；

2.2 湍流模拟

2.2.1 常见湍流模拟方法

现实世界中，流体不是均匀的，而往往是有湍流度的。湍流可以看成是流体流动时连续不定常的波动。湍流在流体中的尺度范围往往非常大，从非常微小的涡结构到与模型尺度接近的大涡都有可能存在。

对湍流的模拟常见的有几种方法。

(1) 直接数值模拟(Direct Numerical Simulation, DNS)

DNS 方法不借助任何其它湍流模型，直接将控制方程在计算域内对整个湍流尺度积分求解。该方法需要有非常细的网格和很小的时间步长以获得准确结果，虽然精确但计算量往往很大。故工程问题一般不采用该方法。

(2) 大涡模拟(Large Eddy Simulation, LES)

LES 对不同尺度的湍流进行了区分，对大尺度的湍流结构，直接计算求解；而其他尺度的湍流成分不直接求解，而只考虑其对大尺度湍流的影响。当网格逐渐加密时，需要直接求解的湍流部分也就逐渐增加了，就越来越接近上面所诉的 DNS 方法，故该方法计算量虽然比 DNS 小，但想要获得较为精确的结果，计算量依旧是比较大的。

(3) 雷诺平均方程(Reynolds-Averaged Navier-Stokes, RANS)

另外一种方法是采用统计的手法来模拟湍流，即采用雷诺平均方程，这是目前工业界运用最广泛的一种处理湍流的方法，也是本文所采用的方法。该方法将湍流分成均值部分和在均值上下波动的脉动部分，取均值的方法依照不同的湍流特性而定。

雷诺平均算子(Reynolds averaging technique operates)定义如下：

$$\phi(\mathbf{x}, t) = \bar{\phi}(\mathbf{x}, t) + \phi'(\mathbf{x}, t) \quad (1-10)$$

式中： $\phi(\mathbf{x}, t)$ ——物理量；

$\phi'(\mathbf{x}, t)$ ——湍流中脉动部分；

$\bar{\phi}(\mathbf{x}, t)$ ——湍流中的均值部分，定义如下：

$$\bar{\phi}(\mathbf{x}, t) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \phi_i(\mathbf{x}, t) \quad (1-11)$$

式中：N——重复事件的次数。

将 \mathbf{U} 用上面的方法写成平均部分和波动部分后代入不可压 N-S 方程，即可得到不可压 RANS 方程组：

$$\nabla \cdot \bar{\mathbf{U}} = 0 \quad (1-12)$$

$$\frac{\partial \bar{U}}{\partial t} + \nabla \cdot (\bar{U} \bar{U}) = g - \nabla p + \nabla \cdot (\nu \nabla \bar{U}) + \nabla \cdot (\bar{U}' \bar{U}') \quad (1-13)$$

观察上式，我们可以发现，它多了和 $\bar{U}' \bar{U}'$ 相关的项，该项为雷诺应力项(Reynolds stress tensor).这就使得原来已经封闭的方程组又有了新的变量，故需要引入额外的模型来使得方程封闭。

2.2.2 RANS 方程湍流模型

通过上面的分析可知，这里引入湍流模型的目的就是要将上文中出现的雷诺应力用已知量来表达，以封闭方程组。目前有两种常见的处理方法。

(1) 雷诺应力模型

该方法直接构建表示雷诺应力的方程，然后与 RANS 方程联立来求解。但是该方法在构建雷诺应力方程时往往引入新的变量，故实际需要构建更多的方程来封闭整个方程组。

(2) 涡粘模型

第二种更为常见的处理方法是采用涡粘模型，也是本文采用的方式。这种方法不直接处理雷诺应力项，而是建立一个雷诺应力和平均速度的梯度之间的关系式。最常见的处理方式是假定这两者之间是线性关系，即 Boussinesq 假定^[32]:

$$\bar{U}' \bar{U}' = \nu_t (\nabla U + (\nabla U)^T) - \frac{2}{3} k I \quad (1-14)$$

式中：

$$k = \frac{1}{2} \bar{U}' \bar{U}' \quad (1-15)$$

上式中涡粘系数(kinematic eddy viscosity) ν_t 可以有很多种表达方法，这也就导致了目前各种各样的涡粘模型（也就是我们常说的湍流模型）。依照模型微分方程数目的多少，线性涡粘模型包括零方程模型、一方程模型、两方程模型等。

两方程模型是最常用的湍流模型，如 $k-\epsilon$ 模型和 $k-\omega$ 模型早已经成为工业领域的标准湍流模型，两者各有优劣。而 $k-\omega$ SST 模型全称剪切应力运输模型(Shear Stress Transport model, SST)^[33]，它结合了 $k-\epsilon$ 模型和 $k-\omega$ 模型各自的优点。这使得 $k-\omega$ SST 模型在近十几年有了非常广泛的应用。同时又考虑到 $k-\omega$ SST 模型在以往风机空气动力性能和流场计算中有较为优秀的表现（参见本文国内外研究现状部分），故本文也选用该模型。

$k-\omega$ SST 模型全部方程组如下所示：^[34]

$$\begin{aligned} \frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho U_i k)}{\partial x_i} &= P_k - \beta^* \rho k \omega + \frac{\partial}{\partial x_i} [(\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_i}] \\ \frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho U_i \omega)}{\partial x_i} &= \alpha \frac{1}{V_t} P_k - \beta \rho \omega^2 + \frac{\partial}{\partial x_i} [(\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_i}] + 2(1 - F_1) \rho \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \\ V_t &= \frac{a_1 k}{\max(a_1 \omega, S F_2)} \\ S &= \sqrt{2 S_{ij} S_{ij}} \\ P_k &= \mu_t \frac{\partial U_i}{\partial x_j} (\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}) \rightarrow P_k = \min(P_k, 10 \beta^* \rho k \omega) \\ F_1 &= \tanh \{ \{ \min[\max(\frac{\sqrt{k}}{\beta^* \omega y}, \frac{500 \nu}{y^2 \omega}), \frac{4 \rho \sigma_{\omega 2} k}{C D_{k \omega} y^2}] \}^4 \} \\ F_2 &= \tanh [[\max(\frac{2 \sqrt{k}}{\beta^* \omega y}, \frac{500 \nu}{y^2 \omega})]^2] \\ C D_{k \omega} &= \max(2 \rho \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}, 10^{-10}) \end{aligned} \quad (1-16)$$

上式中 k 为湍流动能, ω 为湍流频率, y 是到最近的壁面的距离, S 是应力率, ρ 是密度, U_i 是流体速度。 F_1 和 F_2 是混合方程, 在远离壁面处值为 0, 此时 SST 模型转化为 $k-\epsilon$ 模型; 在边界层内其值为 1, 此时 SST 模型又转化为 $k-\omega$ 模型。上式中出现的所有常量, 都是由 $k-\epsilon$ 和 $k-\omega$ 模型中相应的常量通过类似 $\alpha = \alpha_1 F_1 + \alpha_2 (1 - F_1)$ 的方式组合计算而得。模

型中涉及的常量为: $\beta^* = 0.09$, $\alpha_1 = 5/9$, $\beta_1 = 3/40$, $\sigma_{k1} = 0.85$, $\sigma_{\omega 1} = 0.5$,

$\alpha_2 = 0.44$, $\beta_2 = 0.0828$, $\sigma_{k2} = 1$, $\sigma_{\omega 2} = 0.856$ 。该模型相比最初提出时的 SST 模型^[33]有了一些改进, 这里不再赘述。

OpenFOAM 不可压求解器中植入 $k-\omega$ SST 模型时, 采用了张量形式表达, 其中的一些项和参数也有稍许调整。下文将详细比较 OpenFOAM 中 $k-\omega$ SST 模型 (方程左右约去 ρ) 和上文模型形式之间的异同。

表 2-1 上文 $k-\omega$ SST 模型和 OpenFOAM 中 $k-\omega$ SST 模型比较

总体方程形式比较	
k 方程比较 (下式为 OpenFOAM 中形式)	ω 方程比较 (下式为 OpenFOAM 中形式)
$\begin{aligned} \frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho U_i k)}{\partial x_i} &= P_k - \beta^* \rho k \omega \\ &+ \frac{\partial}{\partial x_i} [(\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_i}] \\ \frac{\partial k}{\partial t} + \nabla \cdot (\mathbf{U} k) - (\nabla \cdot \mathbf{U}) k - \nabla \cdot D_{k,\text{eff}}(F_1) \nabla k \\ &= \min(G, c_1 \beta_d k \omega) - \beta_d k \omega \end{aligned}$	$\begin{aligned} \frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho U_i \omega)}{\partial x_i} &= \alpha \frac{1}{V_t} P_k - \beta \rho \omega^2 \\ &+ \frac{\partial}{\partial x_i} [(\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_i}] + 2(1 - F_1) \rho \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \\ \frac{\partial \omega}{\partial t} + \nabla \cdot (\mathbf{U} \omega) - (\nabla \cdot \mathbf{U}) \omega - \nabla \cdot D_{\omega,\text{eff}}(F_1) \nabla \omega \\ &= \gamma(F_1) S_2 - \beta(F_1) \omega^2 + (1 - F_1) C D_{k \omega} \end{aligned}$

续表 2-1

k 方程和 ω 方程中各项形式依次比较			
上文中形式	OpenFOAM 中形式	上文中形式	OpenFOAM 中形式
$\frac{\partial(\rho k)}{\partial t}$	$\frac{\partial k}{\partial t}$	$\frac{\partial(\rho \omega)}{\partial t}$	$\frac{\partial \omega}{\partial t}$
$\frac{\partial(\rho U_i k)}{\partial x_i}$	$\nabla \cdot (\mathbf{U} k) - (\nabla \cdot \mathbf{U}) k^*$	$\frac{\partial(\rho U_i \omega)}{\partial x_i}$	$\nabla \cdot (\mathbf{U} \omega) - (\nabla \cdot \mathbf{U}) \omega^*$
P_k $P_k = \min(P_k, 10 \beta^* \rho k \omega)$ $P_k = \mu_t \frac{\partial U_i}{\partial x_j} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right); \quad c_1 = 10$	$\min(G, c_l \beta_d k \omega)$ $G = \nu_t S_2$	$\alpha \frac{1}{\nu_t} P_k^{**}$	$\gamma(F_1) S_2$
$\beta^* \rho k \omega$	$\beta_d k \omega$	$\beta \rho \omega^2$	$\beta(F_1) \omega^2$
$\frac{\partial}{\partial x_i} [(\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_i}]$ $D_{k,eff}(F_1) = \alpha_k(F_1) \nu_t + \nu$	$\nabla \cdot D_{k,eff}(F_1) \nabla k$	$\frac{\partial}{\partial x_i} [(\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_i}]$ $D_{\omega,eff}(F_1) = \alpha_\omega(F_1) \nu_t + \nu$	$\nabla \cdot D_{\omega,eff}(F_1) \nabla \omega$
--	--	$2(1-F_1) \rho \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}$	$(1-F_1) C D_{k\omega}$ $C D_{k\omega} = (2 \alpha_{\omega 2} \nabla k) \frac{\nabla \omega}{\omega}$

k 方程和 ω 方程公用参数对比	
上文中形式	OpenFOAM 中形式
$\nu_t = \frac{a_1 k}{\max(a_1 \omega, S F_2)}$	$\nu_t = \frac{a_1 k}{\max(a_1 \omega, \sqrt{S_2} F_2)}$
$S = \sqrt{2 S_{ij} S_{ij}}; \quad S_{ij} = \frac{1}{2} \nabla \mathbf{U} + (\nabla \mathbf{U})^T $	$S_2 = \frac{1}{2} \nabla \mathbf{U} + (\nabla \mathbf{U})^T ^2$
$F_1 = \tanh\{\min[\max(\frac{\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega}, \frac{4\rho\sigma_{\omega 2} k}{C D_{k\omega} y^2})]^4\}$	$F_1 = \tanh(\arg_1^4)^{***}$ $\arg_1 = \min\{\min[\max(\frac{\sqrt{k}}{\beta_d \omega y}, \frac{500\nu}{y^2 \omega}, \frac{4\alpha_{\omega 2} k}{C D_{k\omega}^+ y^2}), 10]\}$
$F_2 = \tanh[[\max(\frac{2\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega})]^2]$	$F_2 = \tanh(\arg_2^2)^{***}$ $\arg_2 = \min[\max(\frac{2\sqrt{k}}{0.09\omega y}, \frac{500\nu}{y^2 \omega}), 100]$

k 方程和 ω 方程公用参数对比	
上文中形式	OpenFOAM 中形式
$CD_{k\omega} = \max(2\rho\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}, 10^{-10})$	$CD_{k\omega}^+ = \max(CD_{k\omega}, 10^{-10})$
$\beta^* = 0.09; \beta_1 = 0.075; \beta_2 = 0.0828$	$\beta_d = 0.09; \beta_1 = 0.075; \beta_2 = 0.0828$
$a_1 = 0.31; \alpha_1 = 5/9; \alpha_2 = 0.44$	$a_1 = 0.31; \gamma_1 = 0.5532; \gamma_2 = 0.4403$
$\sigma_{k1} = 0.85; \sigma_{k2} = 1$	$\alpha_{k1} = 0.85034; \alpha_{k2} = 1$
$\sigma_{\omega 1} = 0.5; \sigma_{\omega 2} = 0.856$	$\alpha_{\omega 1} = 0.5; \alpha_{\omega 2} = 0.85616$

[#]: 对于不可压流体 $\nabla \cdot \mathbf{U} = 0$, OpenFOAM 这么处理可能为了求解方便;

^{##}: ω 方程中 P_k 取值和 k 方程一样;

^{###}: 因为 $\tanh(10^2) = \tanh(10^4) = 1$, 即为 F_1 和 F_2 的上限, 故此处左右表达式一致。

通过上面的比较, 我们发现 OpenFOAM 中的 k- ω SST 模型和上文提到的模型基本相同, 主要区别在于一些参数取值上的差异; 此外上文方程中的 P_k 项, 在 OpenFOAM 中用 S_2 来表达。

将上述 k- ω SST 模型与之前得到的 RANS 方程联立, 就可以得到本文的封闭控制方程组。

2.2.3 近壁面处理

对于粘性绕流问题, 我们在处理被绕流物体时一般将其看成不可滑移的壁面 (即流体在壁面上的法向和切向的速度均与壁面相同)。于是湍流在这些不可穿透的壁面上的物理特性与在流场的其他地方肯定会有很大的差别。所以, 在近壁面处对湍流模型进行处理就变得十分有必要了。

一般近壁面区域(inner layer)可以分成三层区域^[35], 如下图所示。最里层为粘性底层(viscous sublayer), 流动几乎为层流, 粘性主导作用; 中间为过渡区域(buffer layer), 粘性和湍流作用相当; 外部为对数区域(log-law region), 湍流扮演重要角色。

从图中我们可以看到, 在粘性底层无因次化的速度 u^+ 和无因次化的距避免距离 y^+ 满足 $u^+ = y^+$ 的关系, 而到了对数区域, 二者之间则更接近对数关系。这种不同的关系也带来了对近壁面流动两种不同的处理方式。

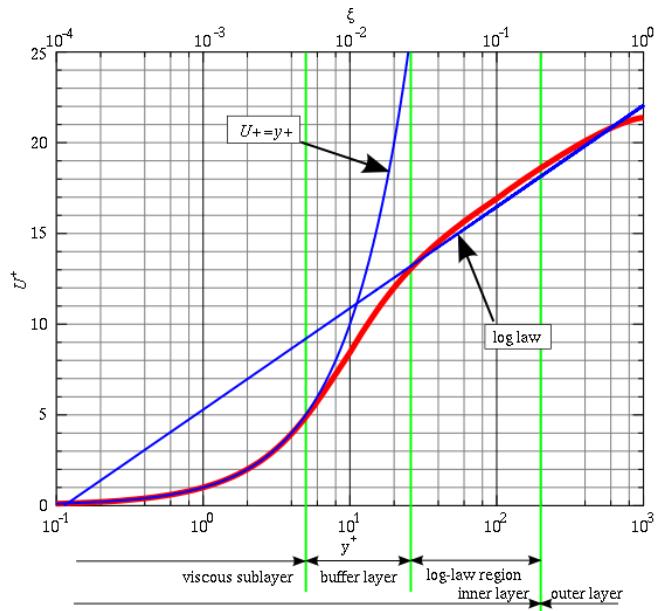


图 2-1 壁面区域三个子层的划分

(1) 低雷诺数模型(low-Re model)

采用低雷诺数模型时,求解器将所有方程在整个计算域内积分,即直接对近壁面处的流动求解,故要求近壁面处的第一层网格高度足够小以保证 y^+ 在 1 左右。这样往往会导致网格量非常大,且时间步长变得非常小,使得计算时间有较大增加。

(2) 高雷诺数模型(high-Re model)

该方法不直接求解近壁面处的流场,而是通过采用壁面函数(wall function)来简单表达壁面处的湍流。壁面函数基于对数区域内流动关系建立,故要求第一层网格位于对数区域,即 $30 < y^+ < 300$ 。用这种方法可以大大减少计算网格量,相对更为经济,受工程界欢迎,也是本文选取的方式。

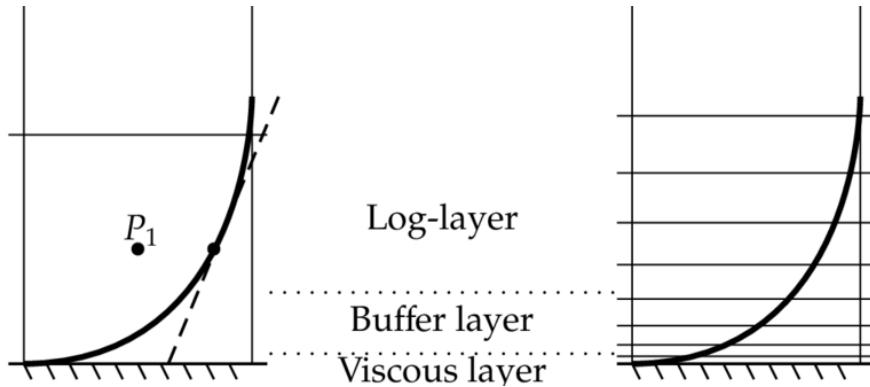


图2-2 高雷诺数模型（左）与低雷诺数模型（右）的比较^[36]

2.3 本章小结

本章主要从连续介质力学控制方程出发,由雷诺输运定理和本构方程,推导得到 Navier-Stokes 方程。对不同湍流模拟方法进行论述,并着重介绍了雷诺平均方程。针对 RANS 方程的湍流模型,本章主要介绍了 $k-\omega$ SST 湍流模型,并将其普通形式与 OpenFOAM 中的形式进行了逐项对比。此外,本章对近壁面处理也进行了探讨,确定选用高雷诺数模型。

第三章 有限体积法

由上文的介绍，我们已经知道不可压粘性流动是由一系列的偏微分方程组所控制，直接求解这些方程组以得到解析解往往是非常困难的，而采用数值方法便可以方便地得到所求解方程的相对精确的近似结果，这也是目前工程界广为使用的处理方法。数值方法的目标是把那些偏微分方程组离散成代数方程组，从而通过求解代数方程组得到原始方程的近似解。离散的过程可以分成两部分：计算域的离散和方程组的离散。

计算域离散后就会包括内部点的位置坐标和边界条件的一些信息。计算域在空间上将被划分成很多离散的小区域，称为控制体或控制单元。对于瞬态数值模拟，时间间隔也会被划分成很多小的时间步长。而方程的离散就是通过一系列的变换，将原来的方程转化成一些代数表达式。

本小节接下来将主要介绍 OpenFOAM 采用的数值离散方法：有限体积法(Finite Volume Method, FVM)^[37]。

3.1 计算域的离散

如前所述，计算域的离散包括时间和空间两部分。

方程最后的解是通过对方程从初始条件设定开始，随着时间步长不断往后推算得到的。因此，对于时间离散，只要预先规定好计算所需的时间步长就足够了。

采用有限体积法时，空间将被离散成有限个控制体(control volume, CV)，也就是网格单元。各控制体互相不重叠并且能完全填充整个计算空间，并为所有变量共享。

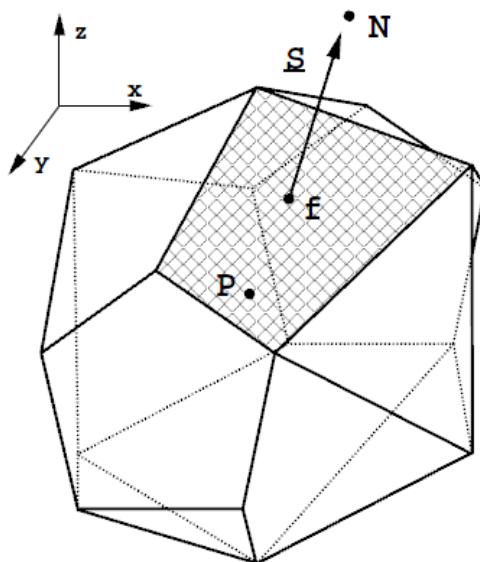


图 3-1 控制体示意图

图3-1即为一个典型的控制体。控制体由一系列的面围成，每一个面只能被一个相邻的控制体共享。计算点P位于控制体中心，即：

$$\int_{V_p} (\mathbf{x} - \mathbf{x}_p) dV = 0 \quad (2-1)$$

网格单元上的面分为两种：位于两个控制体之间的交界面(internal faces)和与边界重合的边界面(boundary faces)。如上图所示，规定交界面的面积法向矢量 \mathbf{S}_f 方向从主体单元(owner)

指向相邻单元(neighbor)，模等于该面面积大小。而边界面的面积法向矢量的方向则是从计算域指向外边界。主体单元和相邻单元的中心分别用P和N表示，交界面的面积法向矢量S_f由主体单元P指向相邻单元的N。所有的控制体的面均用f表示，f也代表这个面的中心。

有限体积法对离散后的控制体形状要求只要是多面体即可，这使得该方法在处理复杂几何物体上有非常大的优势。

3.2 输运方程的离散

任意标量 ϕ 的标准输运方程如下所示：

$$\frac{\partial \rho\phi}{\partial t} + \underbrace{\nabla \cdot (\rho \mathbf{U} \phi)}_{\text{时间导数}} - \underbrace{\nabla \cdot (\rho \Gamma_\phi \nabla \phi)}_{\text{对流项}} = \underbrace{S_\phi(\phi)}_{\text{粘性耗散项}} \quad (2-2)$$

这是一个二阶方程，因为对流项中包含了物理量 ϕ 的空间二阶导数。故有必要使方程离散阶数大于方程本身的阶数以保证解的精确性。从随后本文会介绍对该方程空间和时间上的二阶离散方式。

离散的精度取决于函数 $\phi = \phi(\mathbf{x}, t)$ 在点P附近空间和时间上变化的假定。为了得到二阶精度，这个函数在P点附近空间和时间上至少需要保持线性关系，如下所示：

$$\phi(\mathbf{x}) = \phi_P + (\mathbf{x} - \mathbf{x}_P) \cdot (\nabla_\phi)_P \quad (2-3)$$

$$\phi(t + \Delta t) = \phi^t + \Delta t \left(\frac{\partial \phi}{\partial t} \right)^t \quad (2-4)$$

式中， $\phi_P = \phi(\mathbf{x}_P)$ ， $\phi^t = \phi(t)$ 。

我们再把函数 $\phi = \phi(\mathbf{x}, t)$ 在点P处进行泰勒展开：

$$\begin{aligned} \phi(\mathbf{x}) &= \phi_P + (\mathbf{x} - \mathbf{x}_P) \cdot (\nabla_\phi)_P + \frac{1}{2} (\mathbf{x} - \mathbf{x}_P)^2 : (\nabla \nabla_\phi)_P + \frac{1}{3!} (\mathbf{x} - \mathbf{x}_P)^3 : (\nabla \nabla \nabla_\phi)_P \\ &\quad + \dots + \underbrace{\frac{1}{n!} (\mathbf{x} - \mathbf{x}_P)^n}_{n} : \underbrace{(\nabla \nabla \dots \nabla)}_n \phi + \dots \end{aligned} \quad (2-5)$$

上式中中的表达式 $(\mathbf{x} - \mathbf{x}_P)^n$ 表示矢量 $(\mathbf{x} - \mathbf{x}_P)$ 自身与自身的n阶矢量积，得到一个n阶张量。运算符“::”表示两个n阶张量的内积，得到一个标量。

比较(3-3)与(3-5)我们发现这种线性假定的误差阶数和 $|(\mathbf{x} - \mathbf{x}_P)^2|$ 相当。因此之间假定的函数 $\phi = \phi(\mathbf{x}, t)$ 在空间上有二阶精度。类似的，我们在时间上也可以验证之前线性假定的表达式的误差阶数和 Δt^2 相当，即函数 $\phi = \phi(\mathbf{x}, t)$ 在时间上也有二阶精度。

有限体积法要求在P附近的控制体体积V_P范围内都满足方程(3-2)的积分形式：

$$\int_t^{t+\Delta t} \left[\frac{\partial}{\partial t} \int_{V_P} \rho \phi dV + \int_{V_P} \nabla \cdot (\rho \mathbf{U} \phi) dV - \int_{V_P} \nabla \cdot (\rho \Gamma_\phi \nabla \phi) dV \right] = \int_t^{t+\Delta t} \left(\int_{V_P} S_\phi(\phi) dV \right) dt \quad (2-6)$$

接下来本文对上面方程(3-6)中的每一项的离散过程进行详细说明。

这之前，首先对方程离散的过程中需要用到高斯定理的相关表达式如下：

$$\int_V \nabla \cdot \mathbf{a} dV = \oint_{\partial V} dS \cdot \mathbf{a} \quad (2-7)$$

$$\int_V \nabla \phi dV = \oint_{\partial V} dS \phi \quad (2-8)$$

$$\int_V \nabla \cdot \mathbf{a} dV = \oint_{\partial V} dS \cdot \mathbf{a} \quad (2-9)$$

上式中 ∂V 是值控制体的封闭表面，而 dS 代表一个法向量垂直 ∂V 向外的面微元。

于是，对于控制体内任意变量 ϕ ，我们有：

$$\begin{aligned} \int_{V_p} \phi(\mathbf{x}) dV &= \int_{V_p} [\phi_p + (\mathbf{x} - \mathbf{x}_p) \cdot (\nabla \phi)_p] dV \\ &= \phi_p \int_{V_p} dV + \left[\int_{V_p} (\mathbf{x} - \mathbf{x}_p) dV \right] \cdot (\nabla \phi)_p \\ &= \phi_p V_p \end{aligned} \quad (2-10)$$

式中， V_p 是这个控制单元的体积，而方程的第二项积分为 0，因为 P 点位于控制体的中心。

接下来我们考虑有带有散度算子的项。考虑到一个控制体是由很多面围成的，方程(3-7)的积分可转化成对各个面的求和：

$$\begin{aligned} \int_{V_p} \nabla \cdot \mathbf{a} dV &= \oint_{\partial V_p} dS \cdot \mathbf{a} \\ &= \sum_f \int_f dS \cdot \mathbf{a} \end{aligned} \quad (2-11)$$

由之前对变量 ϕ 的线性假定，我们可以将上式改写成：

$$\begin{aligned} \int_f dS \cdot \mathbf{a} &= \left(\int_f dS \right) \cdot \mathbf{a}_f + \left[\int_f dS (\mathbf{x} - \mathbf{x}_f) \right] : (\nabla \mathbf{a})_f \\ &= S \cdot \mathbf{a}_f \end{aligned} \quad (2-12)$$

式中由于 f 是面的中心，所以第二项积分也等于 0。结合上面的方程(3-10)(3-11)(3-12)，我们就可以得到高斯定理的一个二阶离散形式：

$$(\nabla \cdot \mathbf{a}) V_p = \sum_f S \cdot \mathbf{a}_f \quad (2-13)$$

式中下标 f 指代变量 \mathbf{a} 在面中点的值， S 是一个指向外侧的面法向量。在计算上式的求和项是要注意 S 所在的控制体类型，若是主体单元， S 指向外侧；若是相邻单元， S 指向内侧。所以上式就可以分成主体单元和相邻单元两部分相加而成，即：

$$\sum_f S \cdot \mathbf{a}_f = \sum_{owner} S_f \cdot \mathbf{a}_f - \sum_{neighbour} S_f \cdot \mathbf{a}_f \quad (2-14)$$

这个式子对所有面求和时都适用，在下文的讨论中将默认该式成立。

3.2.1 对流项的离散

根据(3-13)即可得到对流项的离散公式：

$$\begin{aligned}\int_{V_p} \nabla \cdot (\rho \mathbf{U} \phi) dV &= \sum_f S \cdot (\rho \mathbf{U} \phi)_f \\ &= \sum_f S \cdot (\rho \mathbf{U})_f \phi_f \\ &= \sum_f F \phi_f\end{aligned}\quad (2-15)$$

式中：F代表通过该面的质量流量；

$$F = S \cdot (\rho \mathbf{U})_f \quad (2-16)$$

由上可知，质量流量F可通过对 ρ 和 \mathbf{U} 差值得到。值得注意的是，上式中的 ϕ_f 是变量 ϕ 在面中心位置的值，而我们仅已知其在控制体中心的值，故需要选取适当的差分格式由体积中心的值插值得到面中心的值。

对流项差分的方式有很多，这里仅选取几个基本的差分格式予以介绍。

(1) 中心差分格式(Central Differencing, CD)

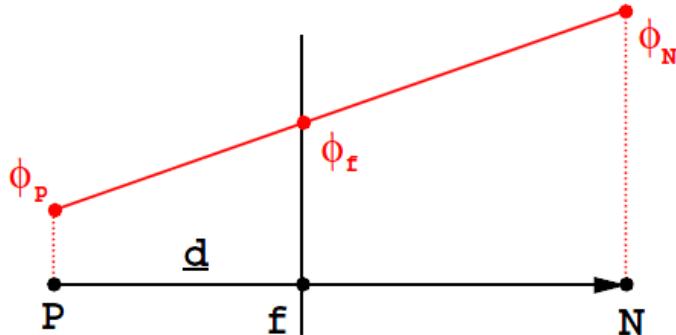


图3-2 中心差分格式示意图

这种格式假定变量 ϕ 的值在P和N之间是线性变化的，故在交界面中心点的 ϕ_f 值为：

$$\phi_f = f_x \phi_P + (1 - f_x) \phi_N \quad (2-17)$$

式中， f_x 定义为f到N点的距离和P到N点的距离之比：

$$f_x = \frac{\overline{fN}}{\overline{PN}} \quad (2-18)$$

通过类似之前的泰勒展开即可得知该方法有二阶精度。但是也有研究发现它在处理对流主导的一些流动问题时会出现不符合物理常识的振动，进而影响解的有界性。

(2) 迎风格式(Upwind Differencing, UD)

在这种格式里，变量 ϕ 的值主要和流体的流向有关，即：

$$\phi_f = \begin{cases} \phi_f = \phi_P & \text{for } F \geq 0 \\ \phi_f = \phi_N & \text{for } F < 0 \end{cases} \quad (2-19)$$

由上面表达式我们可知该方法为一阶精度，它的优势在于可以很好的保证解的有效性，

但这是以牺牲解的精确性为代价的。

(3) 混合差分格式(Blended Differentencing, BD)

这种格式尝试同时保证解的有界性和精确性, 它是上面所述中心差分格式和迎风格式的线性组合:

$$\phi_f = (1-\gamma)(\phi_f)_{UD} + \gamma(\phi_f)_{CD} \quad (2-20)$$

式中, 混合因子 γ 的取值范围为[0, 1], 取0和1时上述混合格式分别转化成迎风格式和中心差分格式。

当然还有很多其他的差分格式, 其中比较有前景的是高阶的迎风格式, 如Linear Upwind Differencing、QUICK等。

3.2.2 粘性耗散项的离散

耗散项的离散方式与对流项相似。结合(3-3)(3-13)可得:

$$\begin{aligned} \int_{V_p} \nabla \cdot (\rho \Gamma_\phi \nabla \phi) dV &= \sum_f S \cdot (\rho \Gamma_\phi \nabla \phi)_f \\ &= \sum_f (\rho \Gamma_\phi)_f S \cdot (\nabla \phi)_f \end{aligned} \quad (2-21)$$

如果网格是正交的, 以下图为例, 那么矢量 \mathbf{d} 和 \mathbf{S} 将是相互平行, 那么下式就会成立:

$$S \cdot (\nabla \phi)_f = |S| \frac{\phi_N - \phi_P}{|\mathbf{d}|} \quad (2-22)$$

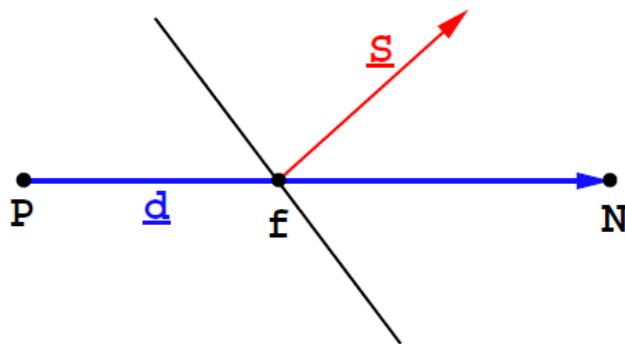


图3-3 耗散项离散示意图

运用上式, 我们就可以用变量 ϕ 在P和N两点的值来计算它在f处的梯度。

但是事实上网格往往是非正交的, 为了让方程(3-22)保留高精度, 我们把 $S \cdot (\nabla \phi)_f$ 分成两项:

$$S \cdot (\nabla \phi)_f = \underbrace{\Delta \cdot (\nabla \phi)_f}_{\text{正交}} + \underbrace{k \cdot (\nabla \phi)_f}_{\text{非正交}} \quad (2-23)$$

上述方程中引入的两个向量 Δ 和 k 需要满足关系式:

$$\mathbf{S} = \Delta + \mathbf{k} \quad (2-24)$$

上式中, Δ 与 \mathbf{d} 平行, 这样我们就可以对对正交项运用方程(3-22)。故方程的精确性仅取决于非正交项。对于 \mathbf{S} 的分解方法常见的有如下三种:

(1) 最小修正法(Minimum Correction Approach)

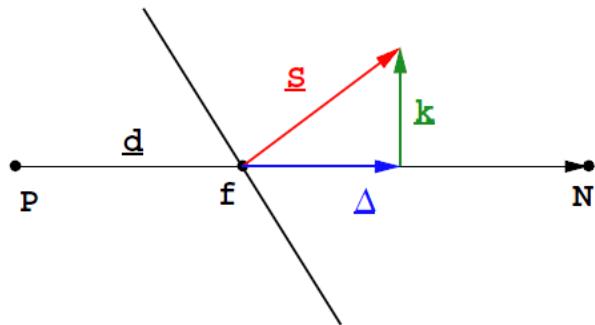


图3-4 最小修正法示意图

这种分解形式使得 \mathbf{k} 矢量与 \mathbf{d} 矢量垂直，故保证了 \mathbf{k} 矢量的模最小。故可得：

$$\Delta = \frac{\mathbf{d} \cdot \mathbf{S}}{\|\mathbf{d}\|} \mathbf{d} \quad (2-25)$$

而 \mathbf{k} 可由方程(3-24)得到。故随着网格非正交性增加， Δ 逐渐减小，即来自 ϕ_p 和 ϕ_N 的贡献将逐渐减小。

(2) 正交修正法(Othogonal Correction Approach)

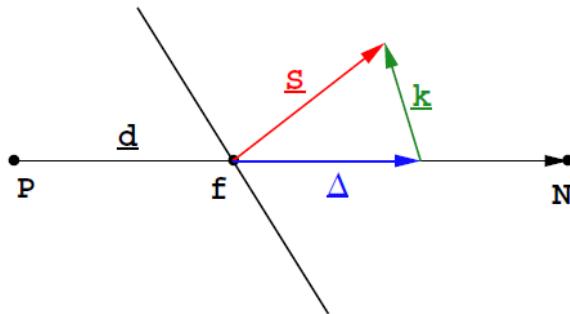


图3-5 正交修正法示意图

这种方法保证了 Δ 和 \mathbf{S} 的模相等，即保证来自 ϕ_p 和 ϕ_N 的贡献与网格的正交性无关。想要达到这个效果，只要定义：

$$\Delta = \frac{d}{\|d\|} S \quad (2-26)$$

(3) 过松弛法(Over-Relaxed Approach)

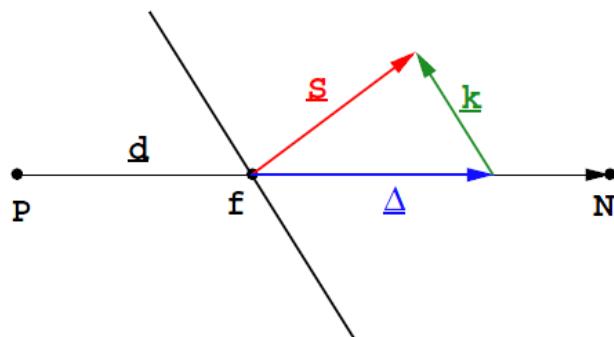


图3-6 过松弛法示意图

这种方法和第一种方法相反，即当网格的非正交性增加时，来自 ϕ_p 和 ϕ_N 的贡献反而增加。此时：

$$\Delta = \frac{\mathbf{d}}{d \square S} / S^{\rho} \quad (2-27)$$

由上面三种分解方法最后得出的耗散项的最终离散形式是一样的。由于 \mathbf{d} 和 Δ 是平行的，故方程(3-23)中正交部分可以写成：

$$\Delta \square (\nabla \phi)_f = |\Delta| \frac{\phi_N - \phi_P}{|\mathbf{d}|} \quad (2-28)$$

这样方程(3-23)就可以写成如下形式：

$$S \square (\nabla \phi)_f = |\Delta| \frac{\phi_N - \phi_P}{|\mathbf{d}|} + k \square (\nabla \phi)_f \quad (2-29)$$

其中，第二项中涉及的 $(\nabla \phi)_f$ 可以先计算与所求面相邻的两个控制体中心点处的梯度然后再线性差值到面上得到，即：

$$(\nabla \phi)_P = \frac{1}{V_P} \sum_f S(\phi)_f \quad (2-30)$$

$$(\nabla \phi)_f = f_x (\nabla \phi)_P + (1 - f_x) (\nabla \phi)_N \quad (2-31)$$

3.2.3 源项的离散

原方程中所有无法写成对流项、耗散项或时间导数项的内容均作为源项。源项 $S_\phi(\phi)$ 可以简单看做是变量 ϕ 的一个函数。在对它进行离散之前，我们首先需要对其进行线性化处理：

$$S_\phi(\phi) = Su + Sp\phi \quad (2-32)$$

式中 Su 和 Sp 也可以和变量 ϕ 相关，由方程(3-10)，我们可得：

$$\int_{V_P} S_\phi(\phi) dV = Su V_P + Sp V_P \phi_P \quad (2-33)$$

3.2.4 时间导数项的离散

通过之前的内容，我们已经实现了输运方程在空间上的离散。下面让我们重新回到输运方程的积分形式，即方程(3-6)。

若控制体体积不随时间变化，由方程(3-15)(3-31)(3-33)可得：

$$\int_t^{t+\Delta t} \left[\left(\frac{\partial \rho \phi}{\partial t} \right)_P V_P + \sum_f F \phi_f - \int_{V_P} (\rho \Gamma_\phi)_f S \square [\nabla \phi] dt \right] dt = \int_t^{t+\Delta t} (Su V_P + Sp V_P \phi_P) dt \quad (2-34)$$

上式被称为输运方程的“半离散”形式。

(1) 二阶Crank-Nicholson方法

结合方程(3-4)，我们可以得到下式：

$$\left(\frac{\partial \rho \phi}{\partial t}\right)_P = \frac{\rho_P^n \phi_P^n - \rho_P^o \phi_P^o}{\Delta t} \quad (2-35)$$

$$\int_t^{t+\Delta t} \phi(t) dt = \frac{1}{2} (\phi^o + \phi^n) \Delta t \quad (2-36)$$

式中，

$$\begin{aligned} \phi^n &= \phi(t + \Delta t) \\ \phi^o &= \phi(t) \end{aligned} \quad (2-37)$$

假定流体密度和耗散率不随时间改变，则根据 (3-35)(3-36), (3-34) 可进一步离散为：

$$\begin{aligned} &\frac{\rho_P^n \phi_P^n - \rho_P^o \phi_P^o}{\Delta t} V_p + \frac{1}{2} \sum_f F \phi_f^n - \frac{1}{2} \sum_f (\rho \Gamma_\phi)_f S \square (\nabla \phi)_f^n \\ &+ \frac{1}{2} \sum_f F \phi_f^o - \frac{1}{2} \sum_f (\rho \Gamma_\phi)_f S \square (\nabla \phi)_f^o \\ &= S u V_p + \frac{1}{2} S p V_p \phi_P^n + \frac{1}{2} S p V_p \phi_P^o \end{aligned} \quad (2-38)$$

(3-38)中的这种时间差分形式叫做Crank-Nicholson法。它可以达到时间上的二阶精度。该方法需要新旧时刻位于控制体中心的点和交界面中心点的 ϕ 和 $\nabla \phi$ 的值。交界面的数值和之前处理类似，对流项采用合适的差分格式由相邻控制体中心点的值差值而得，而耗散项通过方程(3-29)而得。我们的目标是确定新时刻的 ϕ_p 的值。而控制体各个面上的 ϕ_f 和 $(\nabla \phi)_f$ 值可由该控制体和与其相邻的控制体中心点的 ϕ 值得到，故式子可简化为如下的代数方程：

$$a_p \phi_p^n + \sum_N a_N \phi_N^n = R_p \quad (2-39)$$

计算域中每一个控制体积都遵循上述方程。即主体单元 ϕ_p^n 的值依赖于其所有相邻单元的值。因此我们将计算域中所有控制体的方程联立起来，即可得到一个代数方程组：

$$[A][\phi] = [R] \quad (2-40)$$

其中， $[A]$ 为一个稀疏矩阵，系数 a_p 位于该矩阵的对角线上， a_N 则不在对角线上； $[\phi]$ 中存放了所有控制体中心处的 ϕ 值； $[R]$ 矩阵用于存放源项。矩阵的稀疏性取决于控制体的编号，每一个非对角线上的元素都对应于网格的一个面。

求解这个代数方程组，就可得到新时刻下的各控制体中心处的 ϕ 值。方程(3-39)中系数 a_p 包括所有与 ϕ_p^n 有关的项——时间导数项、对流项、耗散项以及源项中的线性部分。系数 a_N 则由所有相邻控制体内相应的项叠加而成。源项则由所有可以由旧时刻的数值直接得

到的所有项组成。

(2) 时间一阶离散格式

Crank-Nicholson 方法对时间离散是稳定的，但是往往不能保证解的有界性。与对流项的处理一样，只要方程按一阶精度离散，就可以保证解的有界性。

在处理时间项时，人们一般习惯性地假定 ϕ 和 $\nabla\phi$ 的在交界面上值不随时间变化。这就使得方程(3-38)简化为：

$$\begin{aligned} \frac{\rho_P^n \phi_P^n - \rho_P^o \phi_P^o}{\Delta t} V_p + \sum_f F \phi_f - \sum_f (\rho \Gamma_\phi)_f S \square(\nabla \phi)_f \\ = SuV_p + SpV_p \phi_P \end{aligned} \quad (2-41)$$

上述方程只有一阶时间精度，依照对 ϕ 和 $\nabla\phi$ 在交界面上值的不同处理方式，可分为下面两种：

- 显式离散格式(explicit discretisation)

这种方法中 ϕ 和 $\nabla\phi$ 在交界面上值完全取决于旧时刻，即：

$$\phi_f = f_x \phi_P^o + (1 - f_x) \phi_N^o \quad (2-42)$$

$$S \square(\nabla \phi)_f = |\Delta| \frac{\phi_N^o - \phi_P^o}{|\boldsymbol{d}|} + \mathbf{k} \square(\nabla \phi)_f^o \quad (2-43)$$

源项中的线性部分也是仅由旧时刻决定。于是，方程(3-41)可化为如下形式：

$$\phi_P^n = \phi_P^o + \frac{\Delta t}{\rho_P V_p} + \left[\sum_f F \phi_f - \sum_f (\rho \Gamma_\phi)_f S \square(\nabla \phi)_f + SuV_p + SpV_p \phi_P^o \right] \quad (2-44)$$

这样处理后，就再也不用求解之前所述的代数方程组了。新时刻的 ϕ_P^n 的值完全可以由旧时刻的值直接通过上式计算而得。这种方法的缺点是要受到克朗数(Courant number)限制，克朗数过大，将会导致结果不稳定。

- 欧拉隐式离散格式(Euler implicit discretisation)

这种方法中 ϕ 和 $\nabla\phi$ 在交界面上值完全取决于新时刻，即：

$$\phi_f = f_x \phi_P^n + (1 - f_x) \phi_N^n \quad (2-45)$$

$$S \square(\nabla \phi)_f = |\Delta| \frac{\phi_N^n - \phi_P^n}{|\boldsymbol{d}|} + \mathbf{k} \square(\nabla \phi)_f^n \quad (2-46)$$

这依旧是一阶精度的离散格式，但是和显示的离散不同，它也会产生一个类似代数方程组(3-39)的系统。这个系统中的耦合要比显示的方式强很多，而且即便克朗数限制被突破，这个系统依旧能保持稳定。这种处理方法与之前的方法最大的不同，便是它能够保证解的有界性。这也正是本文所采用的时间离散格式。

3.3 Navier-Stokes 方程离散

现在我们可以将之前所述的有限体积法的离散方法运用到不可压流体的Navier-Stokes方程上了，即方程(2-8)(2-9):

$$\nabla \cdot \mathbf{U} = 0$$

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

我们可以发现，该方程和之前的输运方程有两个特殊的地方：动量方程的非线性和压力与速度的耦合。

3.3.1 动量方程的非线性处理

(2-9)中的非线性项为 $\nabla \cdot (\mathbf{U} \mathbf{U})$ ，即速度“被它自身所输运”。如此，该项离散后形式中必然包含速度的平方项，进而导致代数方程组变成非线性。对这个问题，有两种处理方法：一是找到一种能够直接求解非线性方程组的方法；二是将非线性项进行线性化处理。由之前对对流项的离散方法，我们可以得到下式：

$$\begin{aligned} \nabla \cdot (\mathbf{U} \mathbf{U}) &= \sum_f S \cdot (\mathbf{U})_f (\mathbf{U})_f \\ &= \sum_f F(\mathbf{U})_f \\ &= a_p \mathbf{U}_p + \sum_N a_N \mathbf{U}_N \end{aligned} \quad (2-47)$$

式中， F 、 a_p 、 a_N 均为速度 \mathbf{U} 的函数。值得注意的是，通量 F 必须满足连续性方程(2-8)。因此，(2-8)和(2-9)应当联立起来一起求解，从而得到了一个更大的非线性方程组。而这样不仅使非线性方程组的求解器更加复杂，也使得计算量大大增加。因此，我们更倾向于采用上述第二种方法，即将非线性项线性化。

所谓线性化，就是我们先预估一个满足连续性方法(2-8)的速度，然后根据这个预估速度计算出(3-47)中的 a_p 、 a_N ，从而使得(3-47)又变成了线性代数方程组得以求解。但是这样的线性化必然会带来非线性项滞后效应。这种滞后效应主要影响于非定常问题。对此，我们有两种处理办法：一是对非线性项进行迭代计算，而是直接忽略非线性滞后效应。值得注意的是，除非计算时间步长比较大，否则迭代过程会显著增加计算量。其优势在于在每一个时间步长内，非线性方程组都是完全求解，其计算量大小完全取决于对时间的精度要求。换言之，如果想得到较为精确的瞬态解，那么时间步长就需要取得比较小。此外，时间步长若取得较小，前后两次迭代结果的差异也相应变小，因此非线性滞后效应也会减小。

3.3.2 压力速度的耦合处理

(1) 压力方程的推导

为了导出压力方程，我们需要使用动量方程的半离散形式：

$$a_p \mathbf{U}_p = H(\mathbf{U}) - \nabla p \quad (2-48)$$

上式通过对动量方程的积分形式用前面所诉方法进行离散得到。值得注意的是，压力梯度目前先保留。

$H(\mathbf{U})$ 这一项由两部分组成：一项是“输运项”，由所有相邻控制体的矩阵系数乘以对应的速度而得；另一项是“源项”，由旧时刻的源项部分和除压力梯度以外的其他源项（这

里没有)组成。 $H(\mathbf{U})$ 可以表达为下式:

$$H(\mathbf{U}) = -\sum_N a_N \mathbf{U}_N + \frac{\mathbf{U}^o}{\Delta t} \quad (2-49)$$

连续性方程(2-8)的离散形式如下所示:

$$\nabla \cdot \mathbf{U} = \sum_f S \cdot \mathbf{U}_f = 0 \quad (2-50)$$

我们可以利用(3-48)得到 \mathbf{U} 的表达式:

$$\mathbf{U}_P = \frac{H(\mathbf{U})}{a_P} - \frac{\nabla p}{a_P} \quad (2-51)$$

交界面上的速度值可以表达为(3-51)向面上插值的形式:

$$\mathbf{U}_f = \left(\frac{H(\mathbf{U})}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f (\nabla p)_f \quad (2-52)$$

上式可用之后于计算通量。

将(3-52)代入(3-50)中即可得到如下形式的压力方程:

$$\begin{aligned} \nabla \cdot \left(\frac{1}{a_P} \nabla p \right) &= \nabla \cdot \left(\frac{H(\mathbf{U})}{a_P} \right) \\ &= \sum_f S \cdot \left(\frac{H(\mathbf{U})}{a_P} \right)_f \end{aligned} \quad (2-53)$$

(3-53)左边的拉普拉斯项可以利用之前所述的方法进行离散。

故不可压流体Navier-Stokes方程组最终的离散形式如下所示:

$$a_P \mathbf{U}_P = H(\mathbf{U}) - \sum_f S(p)_f \quad (2-54)$$

$$\sum_f S \cdot \left(\frac{1}{a_P} \right)_f (\nabla p)_f = \sum_f S \cdot \left(\frac{H(\mathbf{U})}{a_P} \right)_f \quad (2-55)$$

而对流项中的通量 F 可以利用(3-52)式计算:

$$F = S \cdot \mathbf{U}_f = S \cdot \left[\left(\frac{H(\mathbf{U})}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f (\nabla p)_f \right] \quad (2-56)$$

(2) 压力与速度的耦合算法

根据上节中得到的NS方程组的离散形式(3-54)(3-55), 我们会发现速度线性依赖于压力而压力也同时线性依赖于速度。对这种方程内部的耦合目前有两种特殊的处理方式: 同步式解法(Simultaneous algorithms)和分离式解法(segregated approach)。

同步式解法在整个流场内同时求解整个方程组。该方法由于计算量相当大, 对计算机要求较高, 故这种方法只适用于计算节点较少的情况, 不太常用。

分离式解法是按一定的顺序求解方程。PISO, SIMPLE以及它们的衍生算法都是如今十分常用的求解压力速度耦合的方法。PISO算法主要用于瞬态求解而SIMPLE算法主要用于定常求解。接下来本文将介绍PISO算法流程。

- 动量预估(momentum predictor)

首先求解动量方程。由于我们不知道当下时刻的压力梯度值, 故先用上一时刻的压力梯

度值代替。然后求解方程(3-54)，我们就可以得到新时刻速度场的一个近似解。

- 压力求解(pressure solution)

利用上一步求得的速度，我们就可以计算出 $H(\mathbf{U})$ ，进一步求解压力方程(3-55)，我们可以得到新时刻压力场的估计值。

- 显式速度修正(explicit velocity correction)

求解方程(3-56)，我们可以得到一系列与新压力场对应的通量。此时可以用方程对新压力场下的速度场进行修正。

仔细观察(3-51)，我们发现对速度场的修正包括两部分：一是由于压力梯度的变化引起的修正 ($\frac{\nabla p}{a_p}$ 项)；一是相邻控制体的速度修正后输运产生的影响 ($\frac{H(\mathbf{U})}{a_p}$ 项)。所谓的速度显式修正指的是修正时忽略上面提到的第二个部分。即假定整个速度场的误差仅源自压力项。当然，这显然是不对的。因此我们需要对 $H(\mathbf{U})$ 项进行修正重新求解压力方程，并重复上述流程。换句话说，PISO循环是由动量预估得到初始速度场，再计算得到压力场，然后对速度场进行修正，最后不断循环直到预定的误差容忍范围。

压力速度修正次数可通过算例文件fvSolution 中PISO 部分设定nCorrectors的值来设定。

还有一个值得注意的问题就是 $H(\mathbf{U})$ 中的系数依赖于通量场。每一次的压力求解后，我们就可以得到一系列新的通量，此时我们需要重新计算 $H(\mathbf{U})$ 。但是事实上算法并没有这么做，原因是算法假定非线性的耦合并没有压力与速度的耦合重要，这和之前对动量方程的线性化时采取的做法是一致的。因此在一个循环中， $H(\mathbf{U})$ 的系数保持不变直至下个循环。

3.3.3 Navier-Stokes 方程求解流程

不可压湍流 Navier-Stokes 方程的瞬态求解流程如下所示：

- (1) 为所有场中变量设定初始条件；
- (2) 开始下一个时间步长的计算；
- (3) 由已知的通量，建立并求解动量方程，即动量预估；
- (4) 进行 PISO 循环，直到相应变量达到压力-速度耦合系统的残差要求。此时，我们可以得到新时刻的压力场、速度场以及一系列的通量值。
- (5) 用得到的通量来求解方程系统中的其他所有方程。如果是湍流，则由相应的湍流变量计算湍流粘度。
- (6) 如果没有到达最终时间，重复(2)~(6)。

3.4 本章小结

本章主要详细介绍了 OpenFOAM 中采用的数值离散方法，包括计算域的离散和方程组的离散两个部分。本章首先对输运方程离散方式进行了介绍，包括对流项的离散方式及其差分格式、粘性耗散项的离散格式及非正交项的修正方式、源项的离散方式和时间导数项的离散格式等。之后本章介绍了对 Navier-Stokes 方程的离散进行了介绍，包括动量方程的非线性化处理，及压力与速度的耦合算法。本文着重介绍了 PISO 算法，并给出了 Navier-Stokes 方程求解流程。

第四章 OpenFOAM 相关方法介绍

4.1 OpenFOAM 中偏微分方程的表达

对连续介质力学而言，偏微分方程是最自然的建模语言。如果能在计算机中用将偏微分方程用它最自然的语言表达，将会给程序的编写和阅读带来很大的方便，这也正是OpenFOAM架构设计背后最核心的理念。

例如，有如下的动量方程：

$$\frac{\partial(\rho\mathbf{U})}{\partial t} + \nabla \cdot (\phi\mathbf{U}) - \nabla \cdot \mu \nabla \mathbf{U} = -\nabla p \quad (3-1)$$

在OpenFOAM中^[38]，可表示为：

```
solve
(
    fvm::ddt(rho, U)
    + fvm::div(phi, U)
    - fvm::laplacian(mu, U)
    ==
    - fvc::grad(p)
);

```

即便是一个对C++语言一窍不通的人，看到这两种表达，都会发现其内在的联系非常清晰。上面OpenFOAM语句中的solve()函数用于求解动量方程离散后得到的代数方程组。此外，我们可以非常直观而明显的发现OpenFOAM语句中ddt(rho, U)、div(phi, U)、laplacian(mu, U) 和 grad(p) 就分别对应于上面微分方程中的 $\frac{\partial(\rho\mathbf{U})}{\partial t}$ 、 $\nabla \cdot (\phi\mathbf{U})$ 、

$\nabla \cdot \mu \nabla \mathbf{U}$ 和 ∇p 。而方程时间导数项、对流项和耗散项前面的fvm即Finite Volume Method的简称，即对该项用有限体积法进行离散。而fvc即Finite Volume Calculus简称，在这里通过对压力梯度进行积分操作得到压力场。OpenFOAM中所有的隐式离散都在fvm空间中进行，所有的显示离散都在fvc空间中进行。上式中的“+，-，==”操作符已经通过C++中的运算符重载功能赋予新的意义，即可以对场直接进行相应的代数操作。

现在我们对上面OpenFOAM中的微分方程有了更深入的认识。方程左边将各项离散后得到的稀疏矩阵进行代数相加，与方程右边得到的压力场一起就得到了需要的线性方程组。而solver就可以依据设定选取一款线性求解器对方程组进行求解。

4.2 snappyHexMesh 自动网格划分介绍

谈到 CFD 不得不提到网格划分，可以说如何简单快速地生成高质量网格是大家都关心的一个话题。而事实是大部分人准备用 CFD 计算一个问题时，会花费大量时间在网格划分上。此外，目前国内比较流行的网格划分软件如 Gambit、Ansys ICEM CFD 等都要求研究者有相当丰富的网格划分经验才能获得高质量的网格。为了获得高质量网格，研究者往往需要耗费大量时间来学习深入网格的划分技术，这也就占去了他们宝贵的时间。而 OpenFOAM 内置的 snappyHexMesh 工具就是为试图解决这个问题而设计，它可以自动快速地划分六面

体网格，而且这个过程可以并行，大大提高了网格划分的速度。下面本文将简要阐述一下 snappyHexMesh 的网格划分流程。

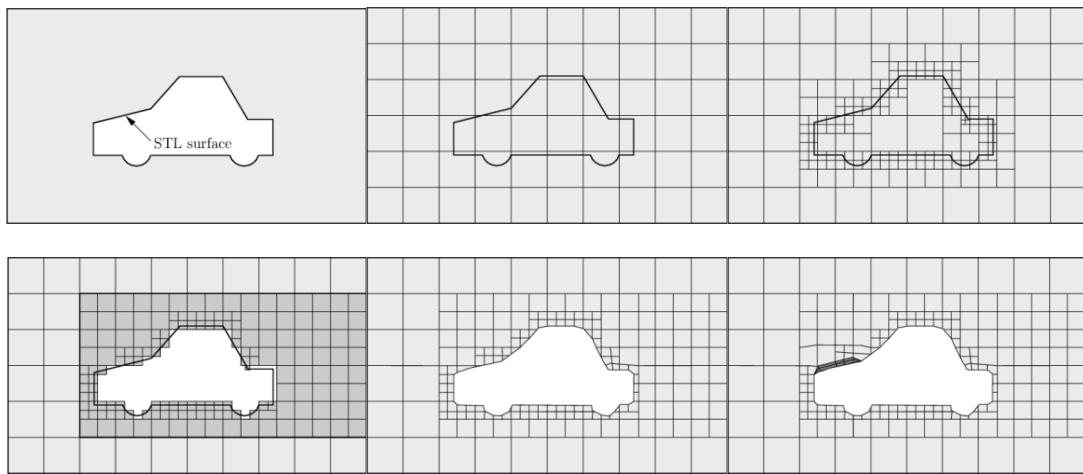


图 4-1 snappyHexMesh 划分网格流程^[38]

我们举一个二维的例子，如上图所示。snappyHexMesh 划分网格大体分为四个步骤。

(1) 准备工作

首先要准备好封闭几何物体的 STL 文件，或者是二进制、ASCII 文件。其次需要对所求的计算域进行初步的网格划分，即要预先提供一套背景网格。要求背景网格全部是六面体结构，并保证背景网格单元与 STL 表面有相交。此外网格单元在物体表面附近的边长比 (Aspect ratio) 最好接近 1，最大不超过 2。如上图 4-1 第 1、2 幅所示。

(2) 网格分裂(castellatedMesh)

此过程首先程序自动探测与背景网格相交的物体边界和表面，然后依照 snappyHexMeshDict 中的设定对该位置的网格进行分裂。比如，

```
refinementSurfaces { car {level (3 4);} }
```

上面的 car 就是物体的名称，使用者希望在 car 的表面附近网格加密 3 级。默认背景网格为 0 级，加密一级即一个网格单元分裂成 4 个（对于三维的即一个网格分裂成 8 个），以此类推。而后面的 4 级是希望在物面变化剧烈的地方（默认设为角度变化超过 30°）加密 4 级。同样的我们也可以对物体的边单独加密。如上图 4-1 第 3 幅图所示。

此外 snappyHexMesh 也支持对指定计算域再加密，如图 4-1 中第 4 幅中深色区域所示。这样可以方便实现不同密度网格的嵌套，减少总的网格量。

在加密过程结束后，程序通过 snappyHexMeshDict 中指定的点的坐标，删去不需要的网格（如车体内部网格）。如上图 4-1 中第 4 幅所示。

(3) 捕捉(snap)

网格划分已经基本完成，这时候需要对物体表面附近六面体网格通过变形操作，使得网格能够捕捉到物体表面。这个过程由多次的迭代光顺计算完成，最终使得物面附近的网格上的点投射到物体表面，如图 4-1 中第 5 幅所示。

OpenFOAM-2.0 之后版本加强了对于物体边界的捕捉能力，只要在程序运行前通过 surfaceExtract 命令提取所需的边界即可。这种情况下，程序在进行 snap 的迭代运算时会保证边界处的网格点落到指定边界上。

(4) 添加网格层(add layer)

snappyHexMesh 支持添加紧贴物体表面的网格层，以方便数值计算时对边界层流动的捕捉。

至此，snappyHexMesh 网格划分就可以开始了。值得一提的是，上面所述的步骤完全可

以一步步分开运行，也可以联合起来运行，甚至采用并行计算。

可见，这种网格划分方式可以大大提高网格划分的效率，具有很好的前景。虽然该方法目前还比较简陋，有不少缺陷，这需要更多的探索和改善。本文也将尝试用该方法对计算模型进行网格，以检验其网格划分的能力。

4.3 topoSet 网格操作介绍

通过上小节的方法可以得到本文所需的网格，但这时的网格还不能用于计算，因为本文计算域中将包含一个相对旋转区域，具体可参见下小节的介绍。故需要从现有的网格中通过一些操作命令获得所需的区域，这时就要用到 topoSet 工具。topoSet 是 OpenFOAM-2.0.0 版本推出的一款工具，它集中了之前相对零散网格操作命令（如 cellSet、faceSet、pointSet 等），并可以依照一定次序同时执行多个上述命令，大大提高了 OpenFOAM 网格操作的效率。

topoSet 的具体操作，由事先放置在算例目录的 system 文件夹下相应的 topoSetDict 控制。理论上该控制文件可以任意命名，但通常以 “***.topoSetDict” 命名以示区别。需要执行该命令时，在算例目录下执行 “topoSet –dict system/***.topoSetDict” 就可以了。下面将简要介绍一下 topoSetDict 的命令组成及使用方式。

一般而言，一个 topoSetDict 的基本操作单元组成如下所示：

```

actions<+
(
{
    name      innerCylinderSmall;
    type      cellSet;
    action    new;
    source    cylinderToCell;
    sourceInfo<+
    {
        p1      (-0.8 0 0);
        p2      (0.6 0 0);
        radius   7.5;
    }
}
)

```

图 4-2 topoSetDict 操作单元示例

topoSet 操作对象是各类的 Set 和 ZoneSet，可以理解为集和域集。包括点集(pointSet)、面集(faceSet)、单元集(cellSet)、面域集(faceZoneSet)、单元域集(cellZoneSet)。其中关键字 actions 括号下包含所有将要执行的网格操作命令。前面两行定义操作对象的名称及类型，如上图操作对象名称为 innerCylinderSmall，它是一个 cellSet。再接下来就是描述需要执行的操作内容。具体的 action 命令分成两类：

(1) 不需要源(source)的操作（即直接对相应的操作对象实现操作）。该类命令包括：

- 清空(clear)，即除去操作对象(Set/ZoneSet)内所有内容；
- 反选(invert)，即选中操作对象之外的其他所有未选中的元素；
- 删除(remove)，即删除选中的操作对象。

(2) 需要源(source)的操作。该类命令包括：

- 新建(new)，即通过源新建一个集(Set)或者域集(ZoneSet)；
- 添加(add)，即将源中的内容添加操作对象上；
- 删去(delete)，即从选中的操作对象中删去源中包含的内容；
- 子集(subSet)，即通过操作对象和源衍生出子集。

上文的源(source)可以指代任意适用的网格信息，亦包含各类 Set 和 ZoneSet。依照不同

的操作对象类型有不同种类的源。源的具体种类比较多，本文不再赘述，具体请参见 OpenFOAM 相关文档。

如图 4-2 所示，我们的目的是通过源创建一个名字为 innerCylinderSmall 的 cellSet。源的类型为 cylinderToCell，即通过一个圆柱形区域内的网格单元创建成所需的 cellSet。该圆柱形区域由 sourceInfo 具体描述。对于本文为一个中心线起始端点为(-0.8, 0, 0)，终止端点为(0.6, 0, 0)，半径为 7.5 的圆柱。以上数字在 OpenFOAM 默认以 m 为单位。

4.4 Arbitrary Mesh Interface 动网格处理介绍

在 OpenFOAM 中处理网格运动有多种方法，适用于处理像风机这种旋转运动的有两种，一个是采用旋转参考系的方法，这种方法计算时网格不动而参考系运动。但是这种方法有两个缺陷，一是它只能用于处理单纯风轮结构，如果再加上塔筒和地面等结构，就无法利用该方法；目前在 OpenFOAM 中该方法只开发了定常求解器，若要研究风机流场的瞬态特性，则需要采用其他方法。而 AMI 就没有这样的限制，故为本文采用。

任意网格界面元方法(Arbitrary Mesh Interface, AMI)是由最新版本 OpenFOAM-2.1.0 新采用的处理网格运动的一种方法。它可以实现完全并行计算。这种方法的原理与在 OpenFOAM 的开发者版本中的通用网格界面元(Generalized Grid Interface, GGI)基本一致，但也存在一些细微区别。这种方法类似于“静态”的滑移网格，但是它不需要每次都重新生成网格，所以更加简便。下面主要结合 GGI 原理^[39]介绍一下 AMI 动网格处理流程。

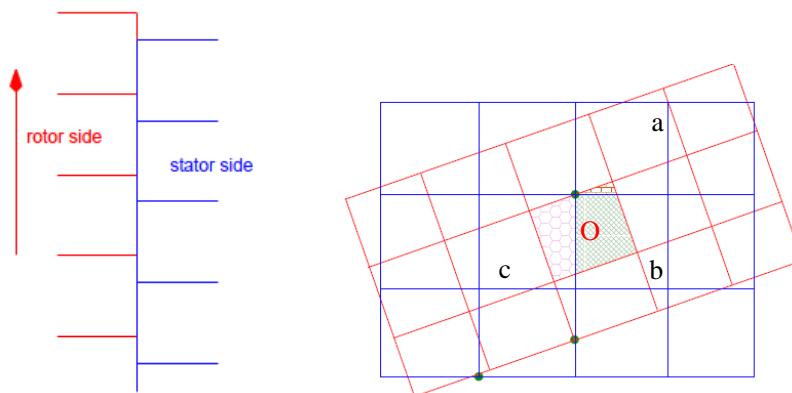


图 4-3 相互滑移网格交界面的处理^[40]

(1) 创建 AMI 交界面

首先我们需要创建 AMI 交界面，这个可以通过网格的拓扑运算完成，如 OpenFOAM 中可以使用 topoSet 工具来选取相应的面域。之后需要把该交界面复制一份，使得一个面属于运动部分(rotor side)，一个属于静止部分(stator side)，如图 4-2 所示。这步操作可以使用 OpenFOAM 中的 createBaffles 和 splitOrMergeBaffles 两个工具完成。工具的具体应用方法及其相应的操作流程可以参见 OpenFOAM-2.1.0 版本中\$FOAM_TUTORIALS/incompressible/pimpleDyMFoam/propeller 算例的设置。

(2) 寻找相邻面

当 AMI 交界面生成后，就可以开始计算了。计算的第一步便是寻找相邻面。一般我们把运动部分称为主面(master face)，静止部分称为从面(slave face)。寻找相邻面的过程简单来说就是给定一个主面，如何找到和他相邻的从面。

如图 4-2 中所示，运动部分的主面 O 与静止部分的从面 a, b, c 三个面相邻，故可得这三个面为主面 O 的相邻面。OpenFOAM 中有相应的算法来搜寻相邻面。

(3) 计算权重

权重的计算是相互的，即既计算从主面到从面的权重，也计算从从面到主面的权重。权

重由相重叠区域占该面的面积来表示，如下：

主面到从面的权重：

$$W_{M_j-to-S} = \frac{|\underline{S}_{M_j-to-S}|}{|\underline{S}_{M_j}|}, \text{ 其中 } W_{M_j-to-S} \in [0,1] \quad (3-2)$$

从面到主面的权重：

$$W_{S_i-to-M} = \frac{|\underline{S}_{S_i-to-M}|}{|\underline{S}_{S_i}|}, \text{ 其中 } W_{S_i-to-M} \in [0,1] \quad (3-3)$$

式中，i——某一个主面的第 i 个相邻从面；

j——某一个从面的第 j 个相邻主面；

$|\underline{S}_{M_j-to-S}|$ 、 $|\underline{S}_{S_i-to-M}|$ ——第 j 个主面和从面重叠区域面积大小，第 i 个从面和主面重叠面积大小

$|\underline{S}_{M_j}|$ 和 $|\underline{S}_{S_i}|$ ——某一从面第 j 个主面的面积和某一主面第 i 个从面的面积；

例如对于上图 4-2，我们想要计算从面 b 到主面 o 的权重，那么通过以上计算得到的权重就是，上图绿色叠影区域面积占从面 b 面积的比例。由此我们可以得到与主面 o 相邻的所有从面到主面的权重。

(4) 差值

有了权重，那么就可以通过带权重差值的方式把速度场、压力场等在 AMI 交界面两侧实现数据交互。

由主面中变量差值得出从面中变量值：

$$\phi_S = \sum W_{M_j-to-S} \phi_{M_j} \quad (3-4)$$

由主面中变量差值得出从面中变量值：

$$\phi_M = \sum W_{S_i-to-M} \phi_{S_i} \quad (3-5)$$

式中， ϕ_{M_j} 和 ϕ_{S_i} ——某一从面第 j 个主面中 ϕ 值和某一主面第 i 个从面中的 ϕ 值。

通过这种带权重的差值，我们就实现了网格相对运动时主面和从面之间数据的交互。

相较于原来的 GGI 方法，AMI 在并行计算上做了很大改善，当并行的核越多时效果越明显。但是为此，AMI 也有了一个较大的缺陷，即无法处理有裸露表面出现的问题。因为对于这样的面，它无法找到相邻的面，由此计算得到的权重就会是 0，从而无法计算。相应的，它对 AMI 交界面处的网格质量要求也比较高，该交界面如果不平整，当网格相互运动时主面和从面容易产生较大间隙，也容易出现无法判断相邻面而导致权重为 0 的情况出现。

4.5 pimpleDyMFoam 求解器

pimpleDyMFoam 求解器是一款可在运动网格基础上求解不可压牛顿流体的瞬态求解器，速度与压力耦合采用 PIMPLE 算法（PISO 和 SIMPLE 的混合算法）。

求解器结构如下图所示：

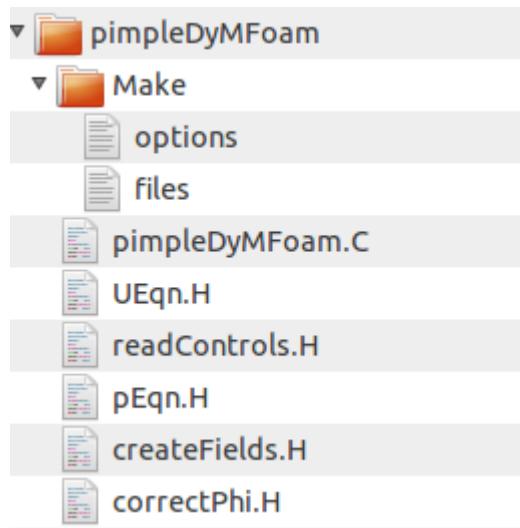


图 4-4 pimpleDyMFoam 求解器结构

OpenFOAM 中一个求解器一般由主程序 “.C” 文件，相关头文件 “.H” 文件和 Make 文件夹组成，可以通过执行 wmake 命令进行编译。Make 文件夹中包含 options 和 files 两个文件，其中 options 文件包含所有需要的头文件和库文件目录地址，而 files 文件则包含所有需要的源文件。用户也可以自行编译所需的求解器。对于 pimpleDyMFoam 求解器：readControl.H 文件主要读入用户设定的相关控制参数；UEqn.H 用于求解动量方程，即进行动量预估；pEqn.H 即采用 PIMPLE 算法求解速度与压力耦合，其中也包含了对网格运动的处理。

4.6 本章小结

本章主要介绍了本文研究运用到的 OpenFOAM 中相关的方法。本章首先介绍了 OpenFOAM 中偏微分方程的表达形式。其次，本章对本文运用到的网格划分工具 snappyHexMesh 的网格划分流程进行了详细介绍。之后，本文有 OpenFOAM 中的网格拓扑处理工具 topoSet 进行了详细介绍。然后，本文结合通用网格界面元(GGI)原理介绍了任意网格界面元(AMI)处理网格相对运动的方式。最后，本章对本文采用的求解器 pimpleDyMFoam 进行了介绍。

第五章 NREL Phase VI 单个风轮数值计算

本章开始将对 NREL phase VI 风机按之前所述方法进行数值计算，从简单开始，本章将对单个风轮结构进行建模计算。本章内容包含了从模型建立、网格划分到最后求解计算的整个过程，除了给出最后的结果，本章也重点关注并分析了本文作者在这整个过程中遇到的困难及其解决方式。

5.1 建立几何模型

5.1.1 叶片模型描述

本文选取的是 NREL phase VI 试验风机。该风机为上风向的 20kW 风机，风轮直径 10.058m。风机有两个叶片，叶片主体部分为 NREL S809 剖面，S809 翼型相对于普通翼型要厚而且尾部拥有一个尖端。叶片不同位置剖面有不同程度的扭转和缩减，这使得整个叶片主体非常扭曲。相应数据参见之前提到的报告^[15]。S809 翼型主要参数如下表所示：

表 5-1 S809 翼型数据

Upper Surface				Lower Surface			
x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
0.00037	0.00275	0.52005	0.09237	0.0014	-0.00498	0.52837	-0.08697
0.00575	0.01166	0.56801	0.08356	0.00933	-0.01272	0.57663	-0.07442
0.01626	0.02133	0.61747	0.07379	0.02321	-0.02162	0.62649	-0.06112
0.03158	0.03136	0.66718	0.06403	0.04223	-0.03144	0.6771	-0.04792
0.05147	0.04143	0.71606	0.05462	0.06579	-0.04199	0.72752	-0.03558
0.07568	0.05132	0.76314	0.04578	0.09325	-0.05301	0.77668	-0.02466
0.1039	0.06082	0.80756	0.03761	0.12397	-0.06408	0.82348	-0.01559
0.1358	0.06972	0.84854	0.03017	0.15752	-0.07467	0.86677	-0.00859
0.17103	0.07786	0.88537	0.02335	0.19362	-0.08447	0.90545	-0.0037
0.2092	0.08505	0.91763	0.01694	0.23175	-0.09326	0.93852	-0.00075
0.24987	0.09113	0.94523	0.01101	0.27129	-0.1006	0.96509	0.00054
0.29259	0.09594	0.96799	0.006	0.31188	-0.10589	0.98446	0.00065
0.33689	0.09933	0.98528	0.00245	0.35328	-0.10866	0.99612	0.00024
0.38223	0.10109	0.99623	0.00054	0.39541	-0.10842	1	0
0.42809	0.10101	1	0	0.43832	-0.10484	0	0
0.47384	0.09843			0.48234	-0.09756		

将上述数据按顺序绘制出来，即可得下面的 S809 翼型图。

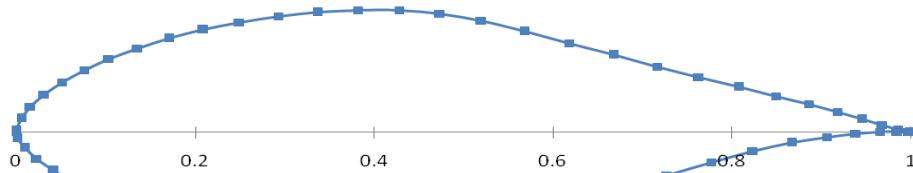


图 5-1 S809 翼型图

叶片不同位置处的剖面参数如下表所示：

表 5-2 NREL phase VI 风机叶片参数表

Radial Distancer (m)	Span Station (r/5.029 m)	Chord Legth (m)	Twist (degrees)	Thickness (m)	Twist Axis (%) chord)
0.508	0.101	0.218 (root hub adapter)	0.0 (root hub adapter)	0.218	50 (root hub adapter)
0.66	0.131	0.218	0	0.218	50
0.883	0.176	0.183	0	0.183	50
1.008	0.2	0.349	6.7	0.163	35.9
1.067	0.212	0.441	9.9	0.154	33.5
1.133	0.225	0.544	13.4	0.154	31.9
1.257	0.25	0.737	20.04	0.154	30
1.343	0.267	0.728	18.074	20.95% chord	30
1.51	0.3	0.711	14.292	20.95% chord	30
1.648	0.328	0.697	11.909	20.95% chord	30
1.952	0.388	0.666	7.979	20.95% chord	30
2.257	0.449	0.636	5.308	20.95% chord	30
2.343	0.466	0.627	4.715	20.95% chord	30
2.562	0.509	0.605	3.425	20.95% chord	30
2.867	0.57	0.574	2.083	20.95% chord	30
3.172	0.631	0.543	1.15	20.95% chord	30
3.185	0.633	0.542	1.115	20.95% chord	30
3.476	0.691	0.512	0.494	20.95% chord	30
3.772	0.75	0.483	0	20.95% chord	30
3.781	0.752	0.482	-0.015	20.95% chord	30
4.023	0.8	0.457	-0.381	20.95% chord	30
4.086	0.812	0.451	-0.475	20.95% chord	30
4.391	0.873	0.42	-0.92	20.95% chord	30
4.696	0.934	0.389	-1.352	20.95% chord	30
4.78	0.95	0.381	-1.469	20.95% chord	30
4.938	0.982	0.358	-1.689	20.95% chord	30
5.029	1	0.356	-1.689	20.95% chord	30

叶片区域从离开转动中心 0.508m 的地方开始，0.508m~0.66m 之间为圆柱，0.66m~1.257m 为过渡区域，1.257m 及之后为 S809 剖面。其中 4.938m~5.029m 处为叶片尾部尖端，这段区域没有扭转，上表中最后两行的数据依照原表格插值得到。图中加粗体的五个剖面位置处每个剖面表面安装有 22 个压力测头(pressure tap)，为实验测量剖面空气动力系数所用。

本文主要参照的 Sequence S 系列实验，该实验风机模型设置参数为 0° 偏航，3° 桨距角。所谓的叶片桨距角(blade pitch angle)指的叶片末端剖面的弦线与转动平面的夹角。故当我们依照上面的数据把叶片建模完成后，还需要将它绕着叶片转动轴从 -1.689° 旋转至 3°。

下面规定计算的坐标轴与正方向。本文计算坐标原点定在叶片旋转中心，x 方向为顺风方向，y 方向水平，z 方向竖直向上。顺着风的方向看风机，逆时针转动为正，初始时刻风机叶片上下分布。剖面桨距角的正方向为向迎风方向旋转。相关示意图如下所示。

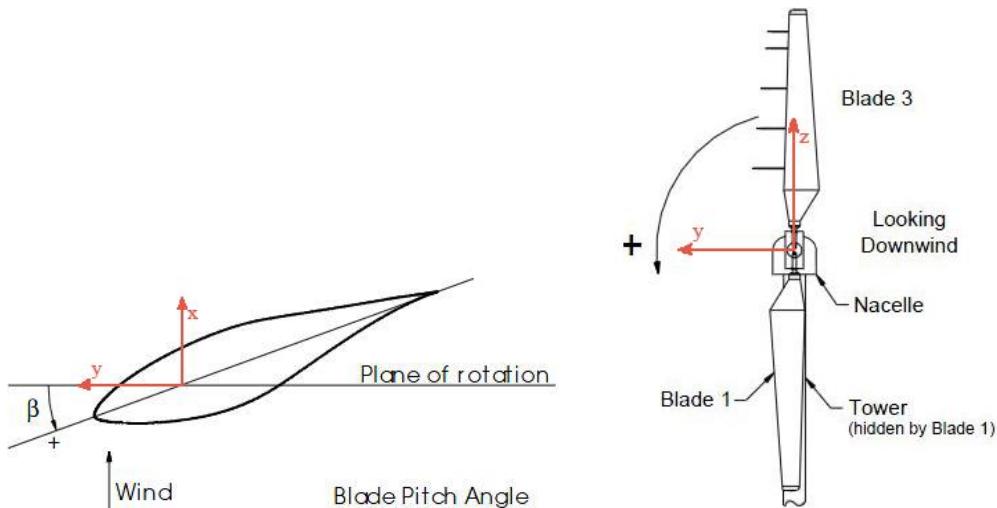


图 5-2 NREL phase VI 风机坐标正方向示意图

5.1.2 风轮模型建立过程

有了上面的数据，我们就可以着手建立风轮的模型了。按照先绘制 blade3 再绘制 blade1 的顺序绘制。详细过程如下所示：

(1) 数据处理

首先需要对各个 S809 剖面的数据进行处理。由上面绘制的 S809 翼型可见，默认状态下 S809 的弦长为 1，默认坐标原点为前端，x 轴为从首端点值尾端点方向，y 轴顺风方向。而所需要的 S809 翼型弦长依照不同位置不同，坐标原点在剖面旋转中心，x 轴方向为顺风方向，y 方向为顺风方向看向左。所以我们需要对原来的 S809 翼型坐标点进行坐标变换。

坐标变换的公式如下所示：

$$x = [y_0 \cos(\overline{Twist}) + (x_0 - 30\%) \sin(\overline{Twist})] \times \overline{Chord} \quad (4-1)$$

$$y = [y_0 \sin(\overline{Twist}) - (x_0 - 30\%) \cos(\overline{Twist})] \times \overline{Chord} \quad (4-2)$$

$$z = \overline{Radial} \quad (4-3)$$

式中， x_0 ， y_0 ——原 S809 翼型点坐标；

\overline{Twist} ——相应位置剖面的扭转角，参见表 5-2，度；

\overline{Chord} ——相应位置剖面的弦长，m；

\overline{Radial} ——相应位置剖面距离转动中心的径向距离，m。

(2) 绘制各剖面

本文绘制风机的三维模型采用 CATIA 建模软件。当数据整理完成后即可导入 CATIA 进行剖面绘制。Excel 数据导入 CATIA 软件可以采用 CATIA 自带的 VBA 简便程序实现，该程序目录为“Catia\intel_a\code\command\GSD_PointSplineLoftFromExcel.xls”。将数据按照要求整理好后，打开 CATIA 软件，然后在 Excel 中执行相应的宏命令即可以实现数据导入。本文导入时剖面的上下部分分开，这样方便之后对叶片过渡区域以及尾部区域几何结构的绘制。本文导入各个剖面效果图如下所示：

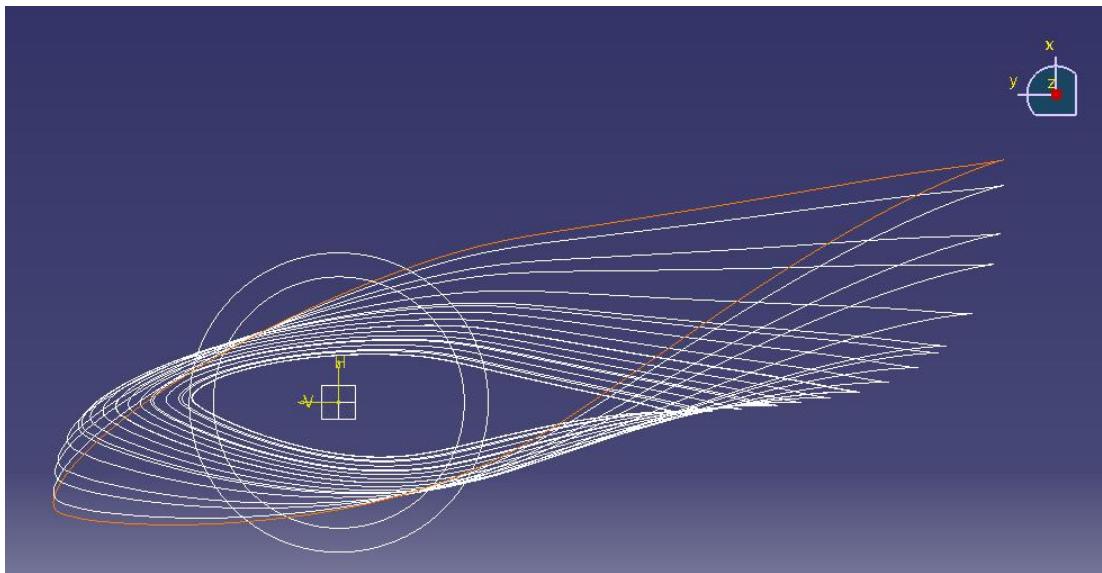


图 5-3 叶片个剖面翼型分布图

图中红色的剖面为 1.257m 处的 S809 剖面，而圆形是叶片根部形状。由图可以看到不同位置处的剖面的扭转和大小变化是十分明显的。

对相应剖面使用 CATIA 多截面曲面命令，即可得到相应的叶片表面，如下图所示：

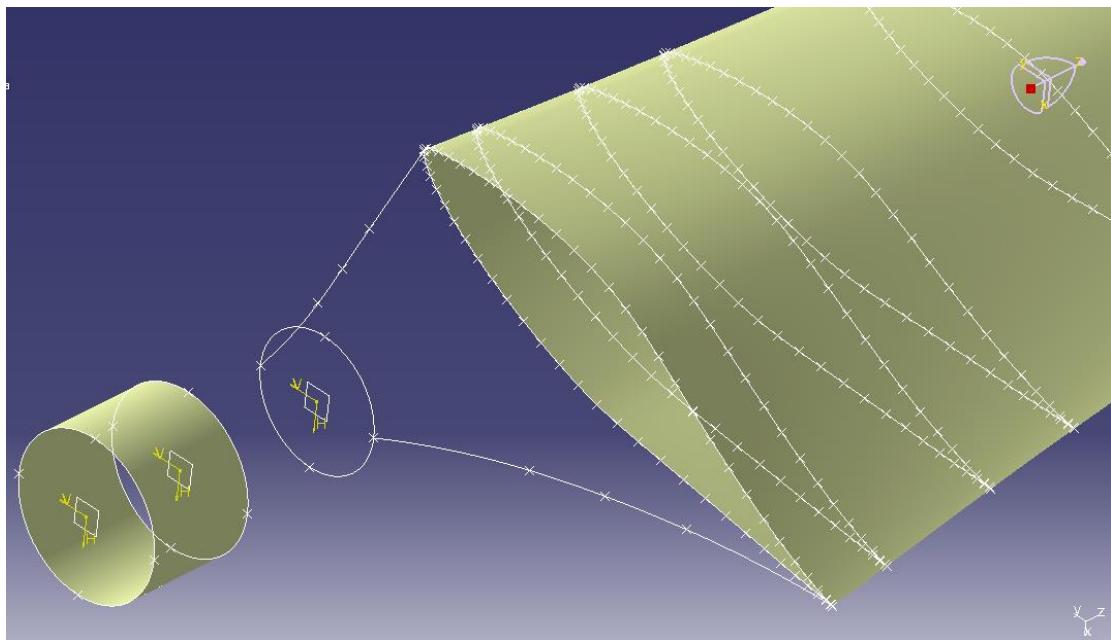


图 5-4 叶片表面图

(3) 绘制过渡区域

由实验给定的叶片几何数据，我们无法得知过渡区域确切的几何形状，从图 5-4 我们也可以看到，过渡区域剖面从圆最后变成 S809 剖面，整个区域唯一可以确定的便是中间三个剖面的首尾端点。这给整个叶片的绘制工作带来了不小的挑战。

据本文作者经验，如果试图直接用多截面曲面的方法绘制该区域，则会出现过渡区域在与 S809 剖面连接处附近怪异的膨大。作者推测这和 S809 翼型末端急速缩小有关。

故如果想要成功绘制该过渡区域，必须自行设计出三个过渡剖面的形状。查看之前的数据表格，发现还没有用到的只有最大厚度这个条件了。但是最大厚度点所在的位置无从得知。故本文先计算出 1.257m 处 S809 剖面最大厚度处点位置坐标，然后将其与叶片过渡段根部

的圆形剖面的最高点进行线性插值，这才计算得出过渡区域三个剖面位置的最大厚度点坐标。这时将得到的过渡区域剖面的点用样条线连接，样条线尾端参考 S809 剖面设定合适相切方向。以上经过过程不断尝试，最终得到合适的过渡段剖面如下图所示：

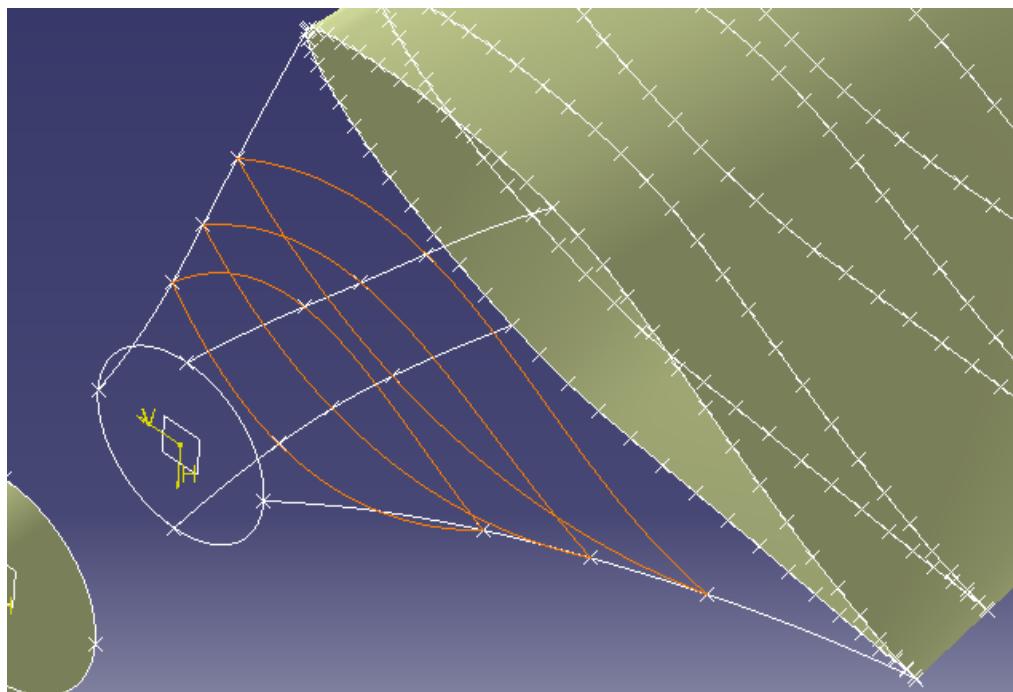


图 5-5 叶片过渡区域剖面绘制

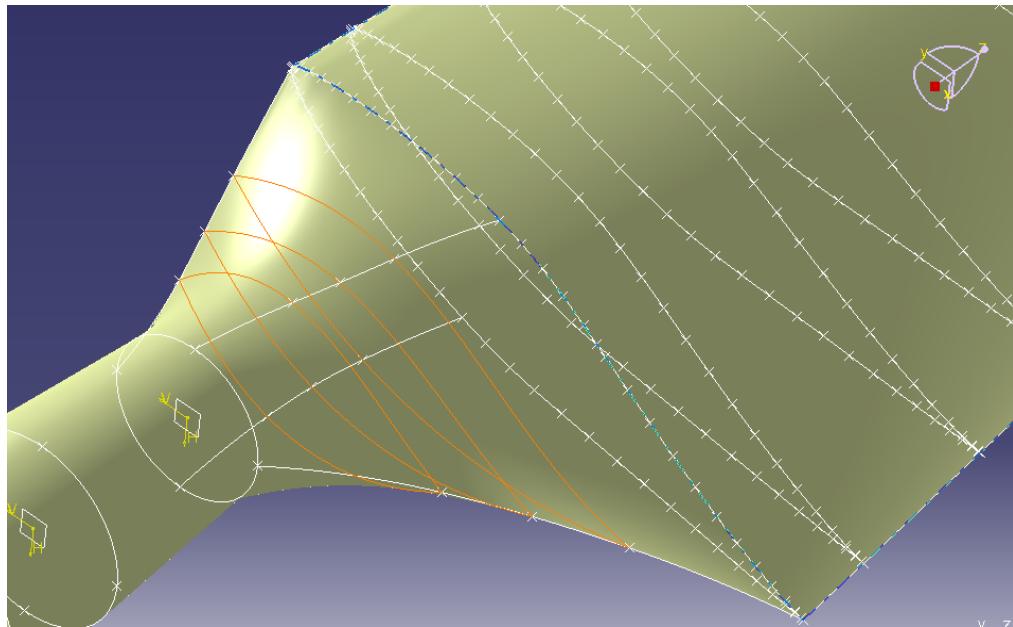


图 5-6 叶片过渡区域完成图

(4) 绘制叶片尾部

原数据表格亦没有给出叶片尾部的具体形状参数，故有两种处理方式，一是保留 S809 剖面，采用截断式；二是参考相关的图片资料等，自行绘制叶片尾部。本文为了和实验更加接近，选用第二种叶片尾部。

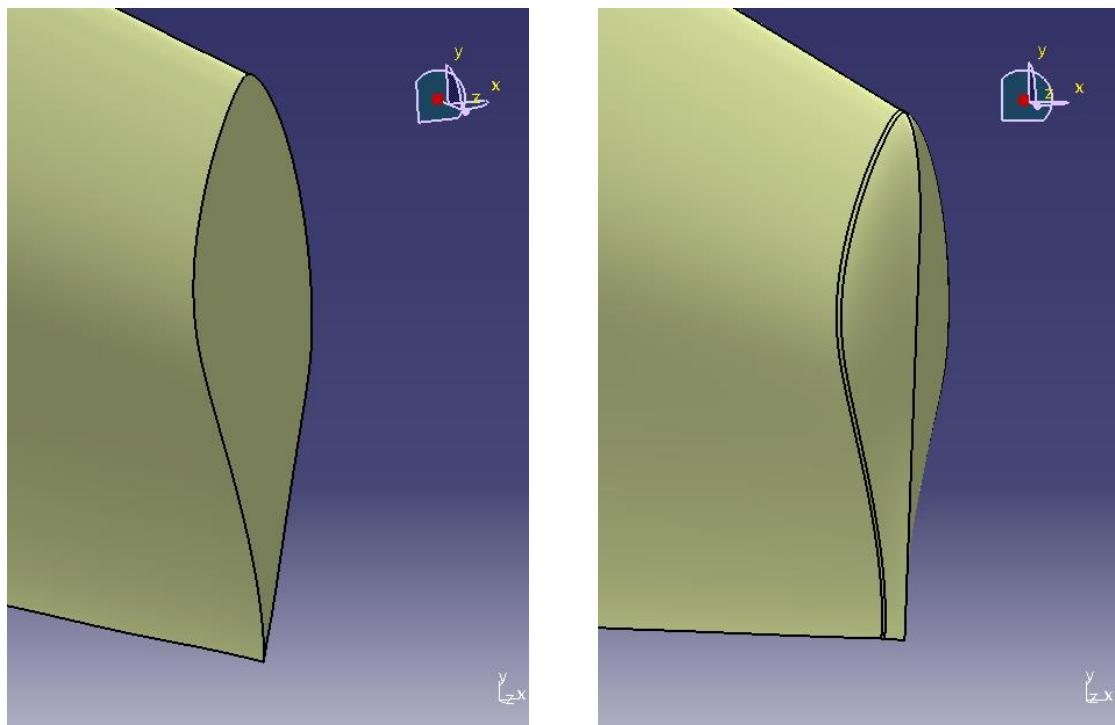


图 5-7 两种叶片尾部的处理方式

(5) 完成整个风轮绘制

在叶片的各个位置都绘制完成后，就可以将其绘制成风轮。在这之前首先要将叶片绕 z 轴旋转 4.689° 以使得叶片的桨距角达到 3° 的要求。然后将叶片绕 x 轴旋转 180° ，得到 blade1。方便起见，我们这里省去轮毂结构，将上下两个叶片直接用圆柱连接。最终效果如下图所示：

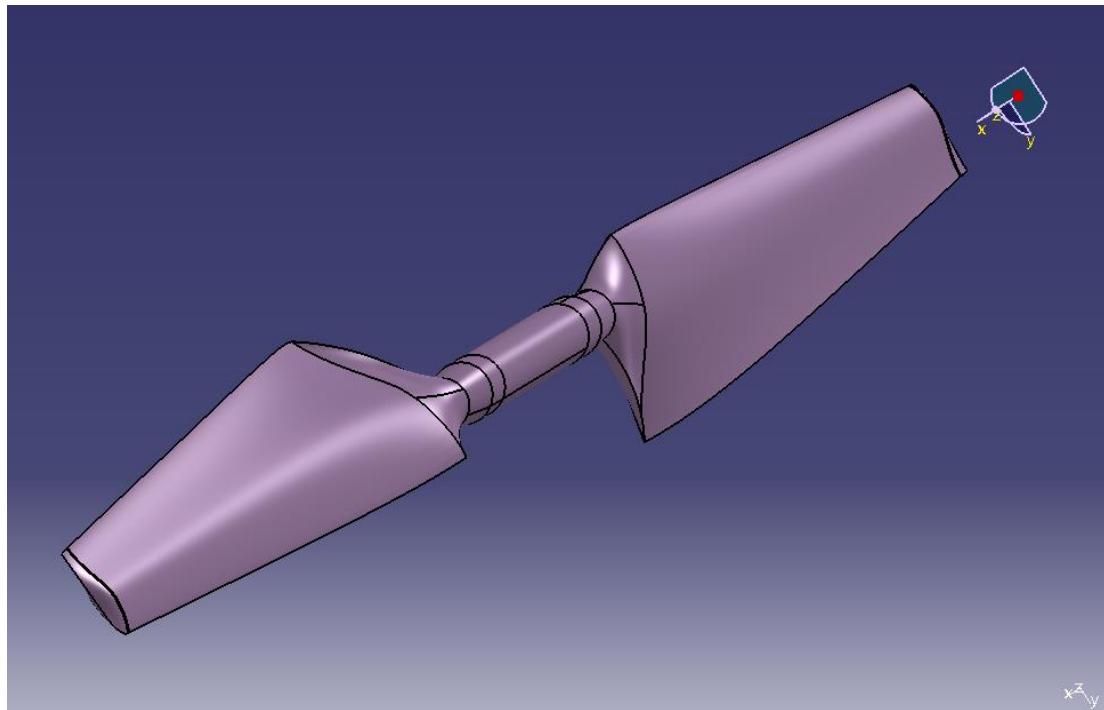


图 5-8 风轮最终效果图

5.2 网格绘制

如前所述，网格的绘制时，本文主要采用 OpenFOAM 自带的自动网格划分工具 snappyHexMesh。但是由于 NREL phase VI 风机叶片扭曲严重，而且导边处比较尖锐，故用 snappyHexMesh 划分网格时有不小挑战。此外，AMI 方法对网格交界面的质量有比较高的要求，这对网格划分也带来不小挑战。

5.2.1 计算域确定

在进行网格划分前，要先确定计算域。方便起见，本章中计算域为圆柱形。考虑到 NREL 的实验是在 NASA Ames $24.4 \times 36.6\text{m}$ 风洞中进行的，故保证截面面积相同则圆柱形计算域的半径可按下式选取：

$$R = \sqrt{\frac{24.4 \times 36.6}{\pi}} = 16.86\text{ m} \quad (4-4)$$

有作者^[18]通过计算发现，这个大小直径的边界对计算结果已经没有什么影响。本文认为该直径还是有一定的裕度的，为计算简便起见，本文确定计算域半径为 15m（约 3 倍叶片长度）。长度方向，本文综合参考了其他作者的设置，也结合自身计算机能力，确定为入口至风机距离为 5m（1 倍叶片长度），风机至出口距离为 20m（4 倍叶片长度）。

5.2.2 纯 snappyHexMesh 划分网格

snappyHexMesh 网格划分流程可参见前面一章。纯 snappyHexMesh 网格划分的设计思路参考 OpenFOAM-2.1.0 中的 propeller 算例，算例位于“\$FOAM_TUTORIALS/incompressible/pimpleDyMFoam” 目录下。

网格设计仿照 propeller 算例，为三层嵌套网格。最里层网格为转动区域，其他部分静止。网格默认采用正六面体单元，经过多次尝试后确定网格单元尺寸如下：

表 5-3 snappyHexMesh 网格划分参数设置

区域	尺寸(m)	网格 level	网格单元(m)
背景网格	$[-5, 20], r=15$	level 0	0.8
加密区	$[-2, 5], r=7$	level 2	0.2
小加密区	$[-0.8, 0.8], r=5.8$	level 2	0.2
风轮	$r=5.029$	level 5(6)	0.025(0.0125)

其中小加密区即旋转区域，其内部包含整个风轮。加密区长度主要考虑风轮转动和形成的涡扩散的距离，这里主要考虑一个叶片长度范围。加密区与小加密区网格级别设置为相同，主要是为了让 AMI 交界面网格更加光顺。而风轮网格 level 5(6)指的是风轮表面网格级别为 5 级，在风轮的边界上网格的级别为 6 级，这样设置能够更好地捕捉叶片的边界。

网格概况如下所示（因为计算域的对称，所以只绘制了一半）：

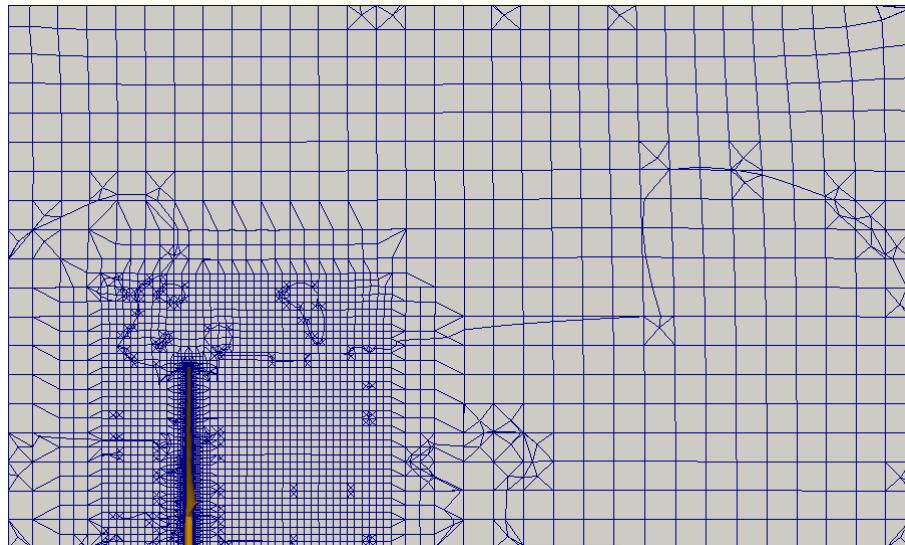


图 5-9 计算域网格侧视图

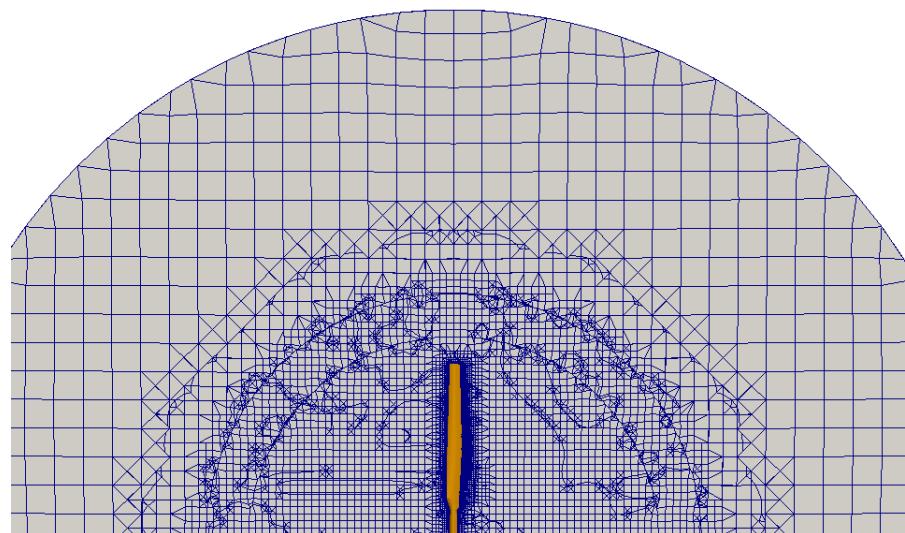


图 5-10 计算域网格正视图

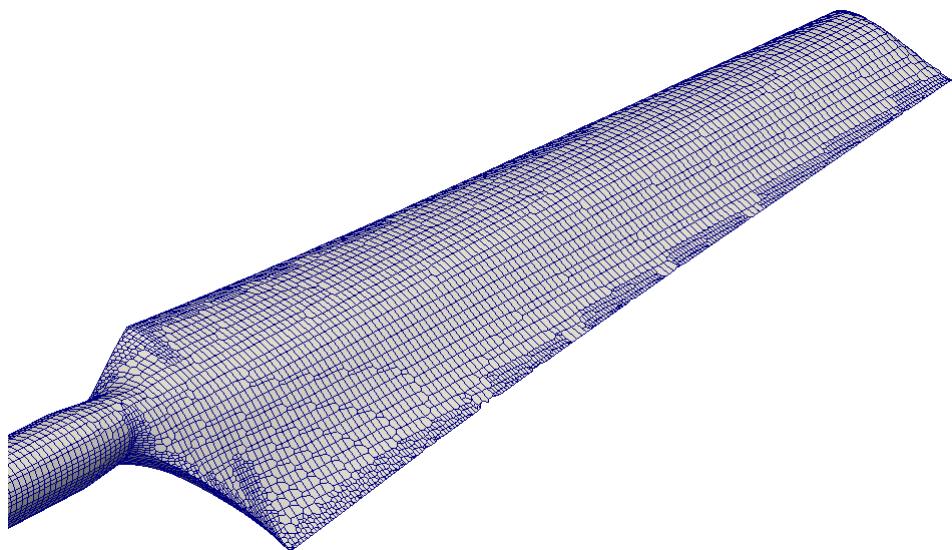


图 5-11 叶片表面网格情况

由上面的图示我们可以发现，snappyHexMesh 在绘制风机流场的网格上能力还是不错的。整个叶片也捕捉地不错。但是也出现了两个比较严重的问题。

(1) maxSkewness 过大

采用 snappyHexMesh 绘制完成网格后，使用 OpenFOAM 的 checkMesh 工具可以得到 maxSkewness 过大的警告，如下所示：

***Max skewness = 12.8685, 24 highly skew faces detected which may impair the quality of the results.

对此，本文作者在 snappyHexMesh 参数设置上进行了各种尝试，最终都无法解决该问题。使用 OpenFOAM 中的 foamToVTK 命令，然后在 paraView 中显示这些扭曲过大的曲面即可得下图：

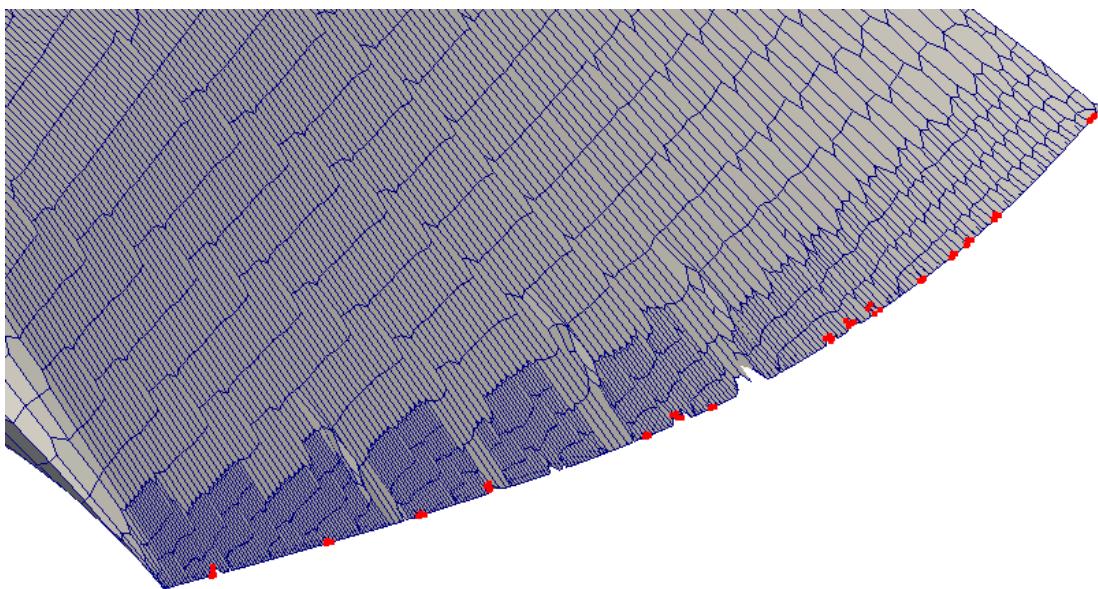


图 5-12 叶片随边处网格

通过这样的处理本文发现，几乎所有的扭曲过大的点都位于叶片的随边（上图红色标记）。而且 snappyHexMesh 绘制的叶片网格在随边处有轻微破碎。可见这是由于叶片随边过于尖锐导致的问题。

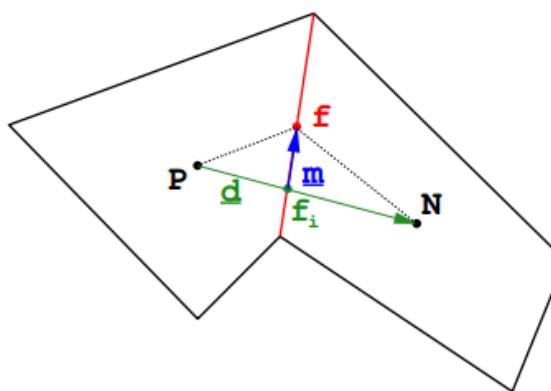


图 5-13 skewness 示意图

参考上图，d为两个相邻网格中心连线矢量与交界面相交于 f_i ，m为由 f_i 指向交界面中心 f 的矢量，OpenFOAM 中 Skewness 的定义^[37]为：

$$Skewness = \frac{|\underline{m}|}{|\underline{d}|} \quad (4-5)$$

故上面提到的 maxSkewness 过大显然是由于叶片随边过于尖锐引起。

在 snappyHexMeshDict 中 meshQualityControl 选项中默认可以接受的 maxSkewness 设定为：

```
maxBoundarySkewness 20;  
maxInternalSkewness 4;
```

而上面 checkMesh 得到的 maxSkewness 约为 13，通过上面分析也可以看到扭曲过大的网格主要也分布在叶片的随边，小于上面 maxBoundarySkewness 为 20 的容许值。故本文认为这个扭曲值还是比较合理的，本文不再花费时间去试图减小网格扭曲了。

(2) AMI 交界面网格质量差

虽然本文在设计网格时已经充分考虑保证 AMI 交界面的网格质量，但是可能是由于 snappyHexMesh 本身立足于六面体网格的缘故，最后得到的 AMI 圆柱面网格质量不是很好。进而导致在后续计算时随着网格的相互运动，在某一时刻 AMI 交界面出现过大空隙。这样就使得权重计算为 0，求解停止。相应的提示如下所示：

```
AMI: Creating addressing and weights between 7144 source faces and 7144 target faces  
AMI: Patch source weights min/max/average = 1.345e-08, 1.05287, 1.00044  
AMI: Patch target weights min/max/average = 0, 1.04703, 1.00034
```

为此，我们查看一下 AMI 圆柱面网格情况，如下所示：

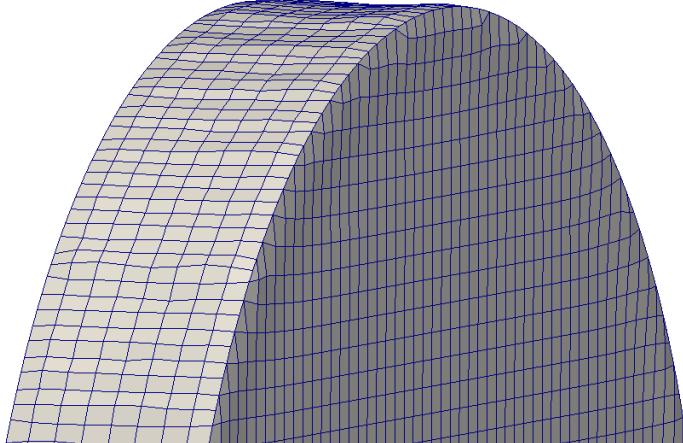


图 5-14 AMI 交界面网格情况

由上图可见，虽然 snappyHexMesh 对整个圆柱面捕捉地不错，但是圆柱侧面还是有很多坑坑洼洼的地方。这些地方在旋转过程中就容易出现过大空隙，导致计算停止。本文对此也尝试了各种办法，最后都没有得到有效改善，甚至无法判断在计算过程中何时会突然出现这种情况。这样显然不是解决办法。对此，本文参考了在 CFD-Online 论坛上热心网友的建议，尝试采用 ANSYS ICEM-CFD 软件来绘制背景网格，确保 AMI 交界面的光顺。

5.2.3 ICEM-CFD 和 snappyHexMesh 联合绘制网格

(1) ICEM-CFD 绘制背景网格

为了保证 AMI 交界面网格质量，对背景网格采用两个嵌套 O 型网格。外面的 O 型网格与计算域外边界相互关联，内部 O 型网格与内部预定的圆柱区域相互关联（该圆柱区域在风轮附近一段可作为 AMI 交界区域）。网格的相互关联和种子布置情况如下所示：

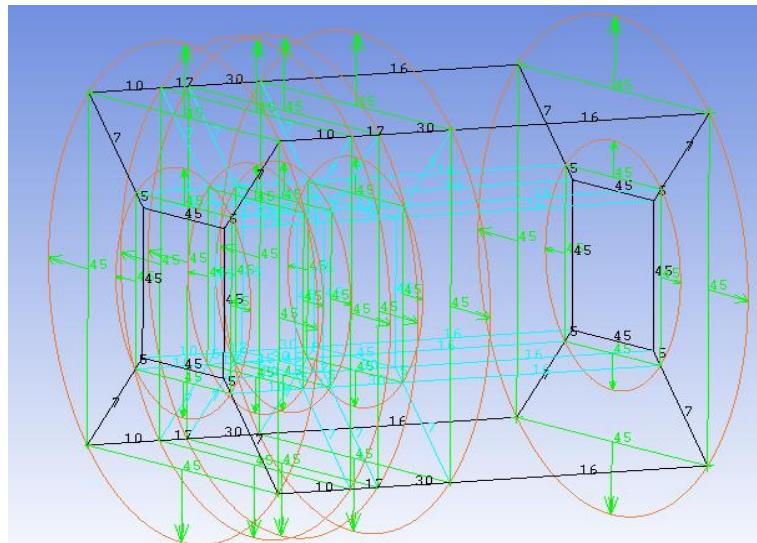


图 5-15 背景网格拓扑关系及种子布置情况

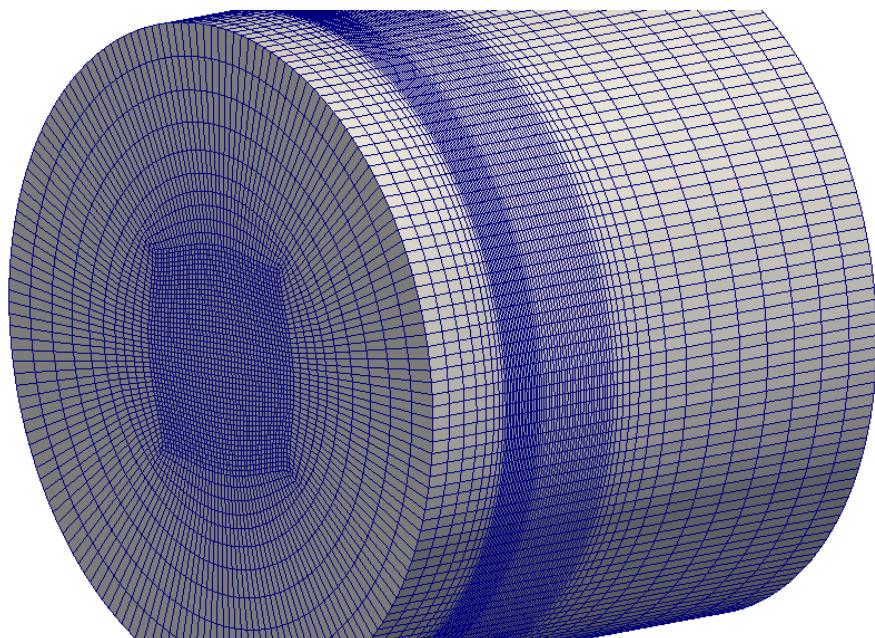


图 5-16 背景网格效果图

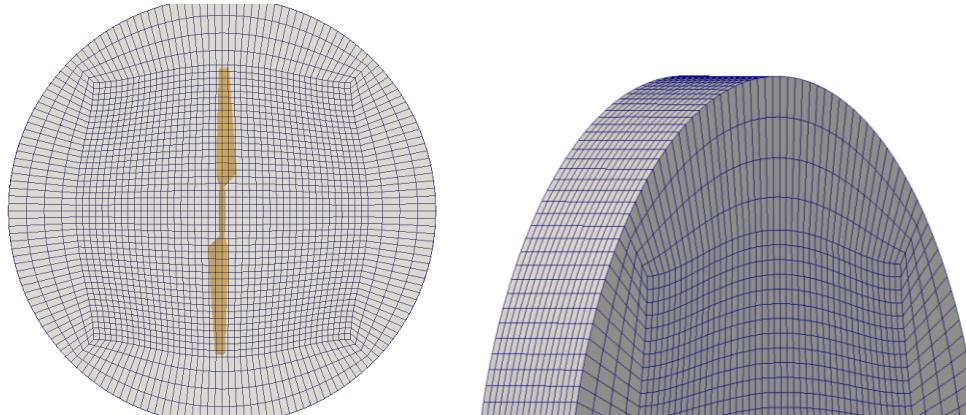


图 5-17 AMI 交界面网格图

网格设计时中心区域的圆柱半径为 7.5m，中心圆柱中心区域尽可能保证网格接近六面

体，整个区域要能够覆盖风轮为准，见上图 5-17。在侧向，靠近风轮区域的±0.8 米范围内，对网格进行了加密。其后 5 米范围内为了捕捉尾涡也进行了加密。网格种子的布置情况如上图 5-15 所示。最终效果如图 5-16 示，风轮中心区域网格大小为 $0.1m \times 0.2m \times 0.2m$ ，这里要保证边长比不大于 2，否则运用 snappyHexMesh 生成的网格质量较差。风轮后侧的加密区域保证网格 x 方向长度在 0.2m 左右。其他区域视情况进行网格稀疏处理，以减少总体网格量。通过图 5-17 我们可以看到 AMI 交界面区域的网格质量有明显的提高。这也达到了我们预期的目的。

网格绘制完成后，将其导出成 fluent_V6 格式，然后对其中的网格文件 fluent.msh 运用 OpenFOAM 的 fluent3DtoFoam 工具将该文件导入到 OpenFOAM 的算例中，保存到 constant 中 polymesh 文件夹下。

(2) snappyHexMesh 绘制整体网格

在绘制整体网格前，首先采用 surfaceExtract 对叶片的边进行提取，以便在后面生成网格时更好地捕捉边界。由于背景网格已经对相关区域进行了加密处理，故在 snappyHexMesh 时不需要再对计算域中网格进行加密处理。而叶片表面的背景网格单元边长为 0.2m 左右，较之前纯用 snappyHexMesh 绘制时的 0.8m 已经有两级的提升，故这里对叶片网格的设置只要在之前的基础上下降两级即可。即叶片表面设为 3 级，叶片边界处设为 4 级。

此外，如前所述，本文的计算使用 wall function 来模拟边界层的流动，故需要保证叶片表面的网格 y^+ 在 [30,300] 范围内。综合考虑不同风速对 y^+ 的影响，贴壁面网格层(surfaceLayer)的覆盖质量以及最后的计算量，本文确定在叶片表面添加三层网格层，膨胀率(expansionRation)1.5，最外层网格层相对厚度(finalLayerThickness)0.6。

最后得到的网格数量如下表：

表 5-4 网格总数变化情况

背景网格（万）	网格分裂后（万）	添加网格层后（万）	最终网格量（万）
25.5	35.6	40.6	40.6

网格情况如下列图所示：

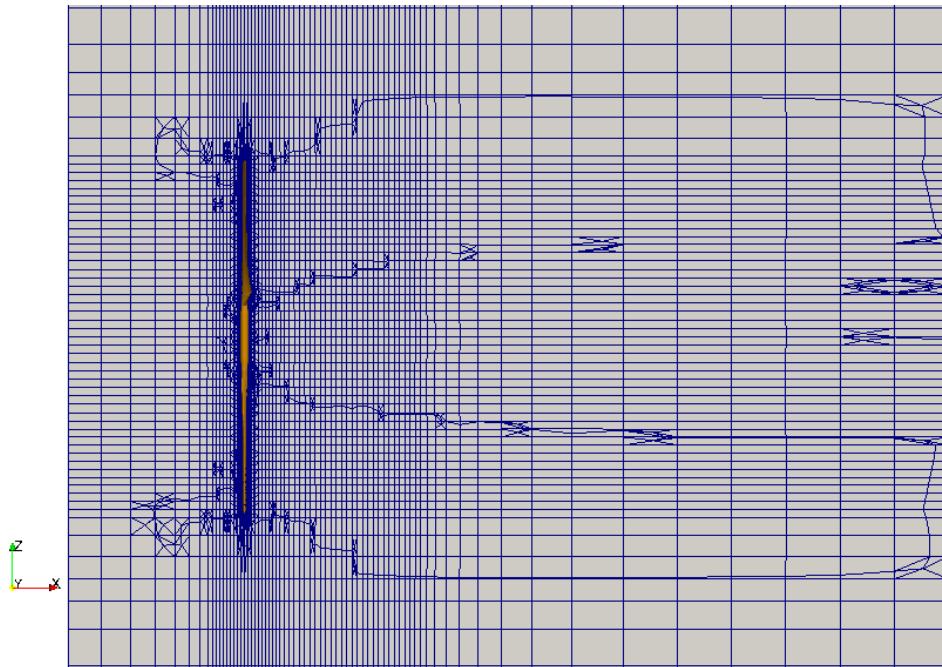


图 5-18 计算域网格侧视图

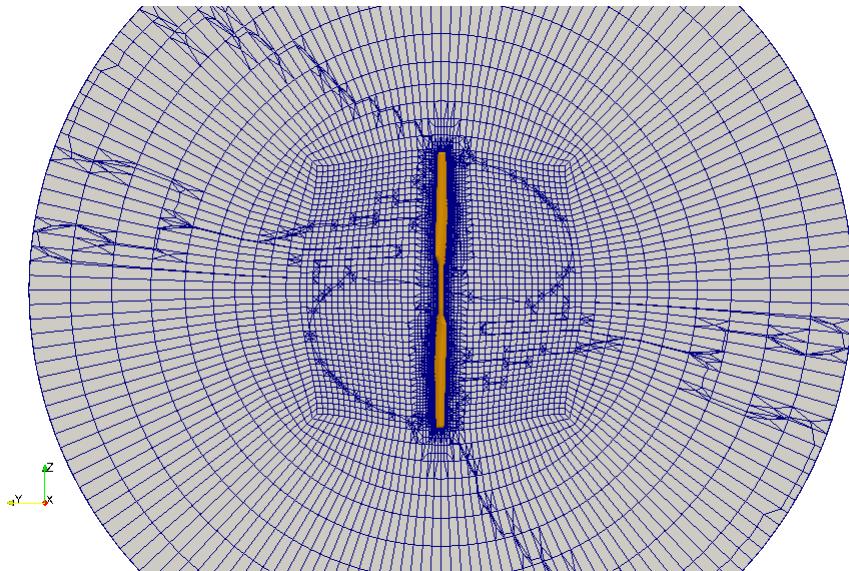


图 5-19 计算与网格正视图 ($x=0$ 处截面)

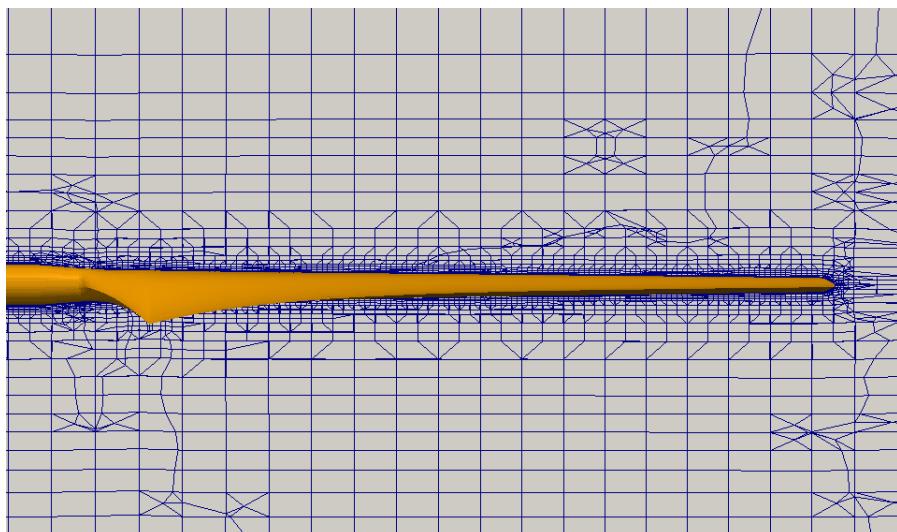


图 5-20 叶片附近网格分布情况

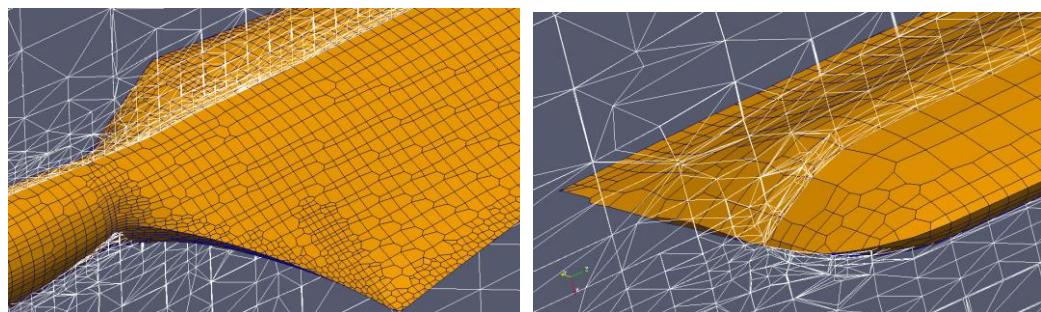


图 5-21 叶片首尾端网格分布情况

(3) AMI 交界面生成过程

网格生成完后，需要调用 OpenFOAM 的一些网格拓扑运算工具来生成 AMI 交界面。

第一步，用 topoSet 工具 remove 命令去除用 ICEM-CFD 绘制网格时建立的“场”（即材料点所在几何体），否则之后运算时会出现一个点属于两个场的现象。

第二步，再用 topoSet 工具相应命令创建所需的 AMI 交界面的区域。

(1) 用 new 命令新建一个 cellSet，名字为 A，源为中心线（沿 x 方向）为[-0.8m,0.8m]

半径为 7.5m 的圆柱区域，即之前设计网格时预先确定的区域：

- (2) 再新建一个 cellSet，名字为 B，源为 cellSet A；利用 invert 命令将 cellSet B 反选；
- (3) 用 new 命令新建 cellZoneSet 和 faceSet，源分别为 cellSet A 和 cellSet A 所有的面；
- (4) 用 subSet 向上面新建的 faceSet 添加子集，源为 cellSet B 所有的面（已反选）；
- (5) 新建一个 faceZoneSet，源为 cellSet A 和上面的 faceSet。

至此，我们得到了一个包含 AMI 内部单元和边界面（正反两个方向）的 faceZoneSet。

第三步，运行 changeDictionary 命令对前面选定的 AMIFaces 赋予一定的物理属性。该步骤将读入 AMI 交界面的信息(nFaces, startFace, neighbourPatch)，以及相应的边界类型(cyclicAMI)。这其中需要给出 AMIFaces 的起始面序号，初次运行可先不管。待其提示出错后做相应修改，然后再运行该命令即可。

第四步，在运行下一步之前需要将运行 snappyHexMesh 时生成的初始文件（一般是 0 文件）删除。

第五步，运行 createBaffles 命令，将内部的面(internal faces)转化成边界面(boundary faces)，为下一步做准备。

第六步，运行 mergeOrSplit-split 命令，将上一步生成的边界面分裂，得到 AMI 交界面：面法向向内的面及其里面的单元是旋转的；而面法向向外的面及其外面的单元是静止的。

生成的 AMI 交界面参见图 5-17。

5.3 初始条件、边界条件设置

边界条件和初始条件的设置位于 0 文件夹下。边界条件的设置参考下图。因为本文使用带 wall function 的 k- ω SST 湍流模型，所以也需要对相应的 k , ω , v_t 进行设置。除了来流速度依照实验而定， k 、 ω 的数值通过估算得到外，其他一律默认为 0。

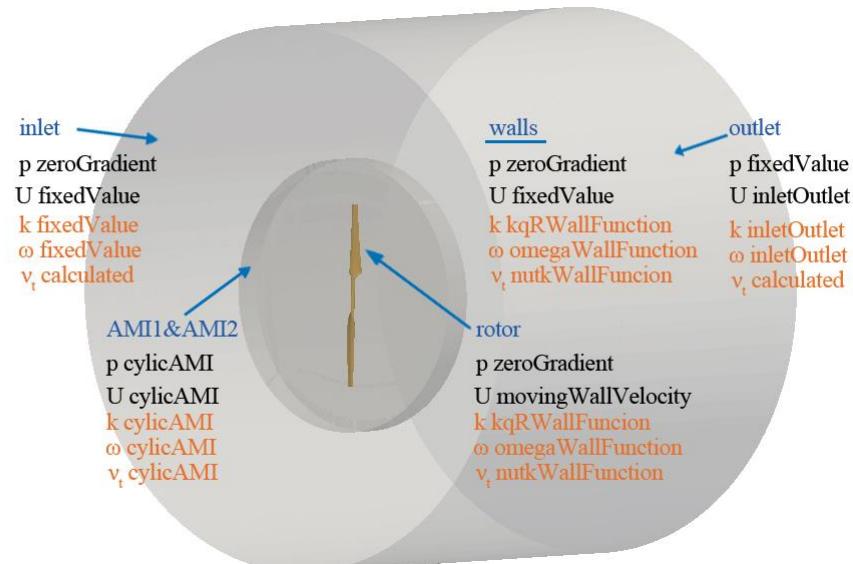


图 5-22 边界条件设置

各实验工况下的初始参数如下表所示：

表 5-5 实验各工况参数

实验编号	风速(m/s)	空气密度(kg/m ³)	运动粘度(m ² /s)	转速(rpm)
S0500000	5	1.243	1.424E-5	72
S1000000	10	1.246	1.423E-5	72
S1500000	15	1.224	1.450E-5	72
S2500001	25	1.220	1.454E-5	72

5.4 算例结构与求解参数设置

5.4.1 算例结构与求解参数设置

本文的标准算例结构如下图所示：

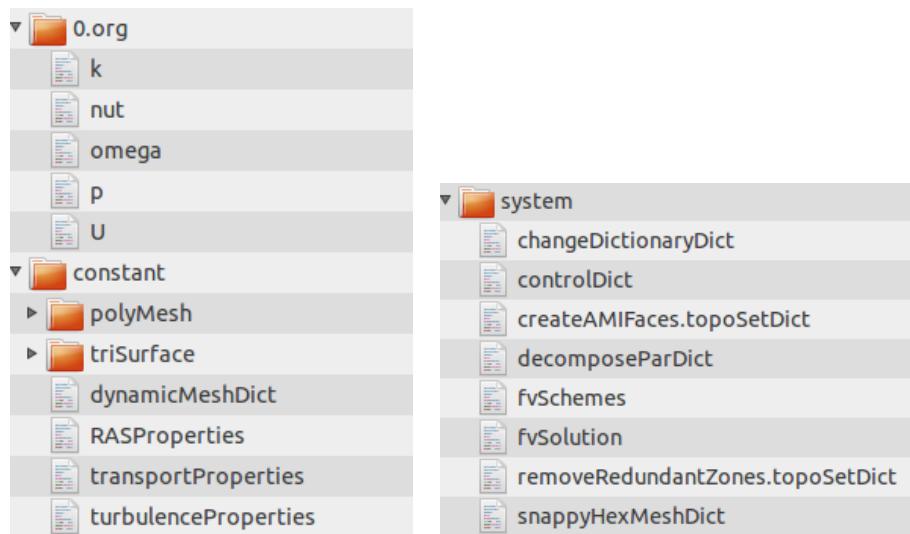


图 5-23 本文标准算例结构

本文标准算例主要包含三个文件夹：0.org, constant 和 system 文件夹。简要介绍如下：

(1) 0.org

该文件夹主要用于设置初始条件和边界条件，本文涉及的 U、p、k、 ω 、 v_t 均在此文件夹下相关文件内设置。这里用 0.org 主要为了防止在进行前处理操作时 0 文件夹被占用，如用 snappyHexMesh 生成网格时就会生成 0 文件夹。当其他前处理工作全部完成后，须将 0.org 重命名为 0 文件夹，作为算例的初始时刻文件夹。

此外在整个计算过程中，按照所需的时间间隔输出的场的信息也会保存在成相应的时间文件夹中。如 0.5 文件夹即保存了 0.5s 时刻的所有流场信息。

(2) constant

constant 文件夹顾名思义主要存放一些计算时的常量。其中的 polyMesh 文件夹中存放了整个计算域的网格信息。而 triSurface 文件夹用于存放 snappyHexMesh 划分网格时需要用到的几何 “.stl” 文件。

dynamicMeshDict 文件包含运动信息，对于本文即设置 AMI 内部区域运动形式为旋转，旋转轴为 x 轴，角速度为 72RPM。turbulenceProperties 即采用何种湍流模式，对于本文为 RANS 方程，故设置为 RASModel。RASProperties 用于设置 RANS 方程的涡粘模型，对于本文设置 kOmegaSST，并开启湍流即可。最后，transportProperties 主要设置流体的特性，对于本文为牛顿流体，而运动粘度依照实验设定即可。

(3) system

system 文件夹下主要存放控制文件，所有的控制参数都可以在这里设置和调节。

之前已经提到的 snappyHexMeshDict 用于设置 snappyHexMesh 网格划分相关参数。removeRedundantZones.topoSetDict、createAMIFaces.topoSetDict 和 changeDictionary 三个文件用于控制网格拓扑操作的参数，相关的操作前面网格部分已经论及。

还剩下四个文件，下面依次介绍：

- decomposeParDict

之前已经提到，OpenFOAM 中绝大部分功能和求解器都可以通过 MPI 实现并行计算。decomposeParDict 就是用于控制将计算域分割成多个处理单元的。本文采用 hierarchical 的分

割方式，将计算域在 x 方向分成四份，再在 z 方向分成两个，共计 8 个处理单元。

- controlDict

这个文件主要控制计算的时间步长，也包含一些后处理功能。

时间的控制上包含起始时刻，时间步长，终止时刻，文件输出的控制等。由于本文的算例计算量较大，为了加快速度并保证收敛。本文采用变步长的方式控制时间即采用 adjustableRunTime 来控制步长，保证最大克朗数不超过 2。通过计算本文的所有算例的稳定时计算步长一般在 $3e-5s \sim 1e-4s$ 左右。风轮转速为 72RPM 即 1.2 转每秒。综合考虑计算量与计算结果稳定性，本文计算终止时刻设定为 2.6s，即风机旋转 3 圈又 43.2° 。

在这个文件中还可以设置力和力矩的输出。只需设定需要计算力和力矩的物体名称，流体的密度，力矩中心即可。最后就可在算例目录下得到一个 forces 文件夹，里面包含了不同时刻三个方向的力和力矩值。

- fvScheme

这个文件主要设置 Navier-Stokes 方程的离散差分格式。对于本文，设置如下：

表 5-6 fvScheme 参数设置

ddt	Euler	div(phi,U)	Gauss linearUpwind grad(U)
grad(p)	Gauss linear	div(phi,k)	Gauss upwind
grad(U)	cellLimited Gauss linear 1	div(phi,omega)	Gauss upwind
laplacian	Gauss linear corrected	div((nuEff*dev(T(grad(U)))))	Gauss linea
interpolation	linear	snGrad	corrected

- fvSolution

该文件主要进行求解器设置，包括求解器类型以及容许残差等。对于本文设置如下：

表 5-7 fvSolution 参数设置

	pcorr	p	pFinal	“(U k omega)”	“(U k omega)Final”
solver	GAMG	GAMG	GAMG	smoothSolver	PbiCG
tolerance	1e-3	1e-7	1e-6	1e-6	1e-6
relTol	1e-7	0.01	0	0.1	0
smoother	DICGaussSeidel	DICGaussSeidel	DICGaussSeidel	GaussSeidel	--
preconditioner	--	--	--	--	DILU
maxIter	50	50	50	--	--

该文件也包含了对 PIMPLE 算法的设置，对于本文设置如下：

表 5-8 PIMPLE 算法参数设置

correctPhi	nOuterCorrectors	nCorrectors	nNonOrthogonalCorrectors
yes	3	2	3

5.4.2 算例运行流程

这里的流程主要是指，前处理完成之后算例运行的流程。关于网格的生成和拓扑运算流程请参见本章网格部分内容。本算例的运行流程如下所示：

- (1) 运行 `cp -r 0.org 0`；拷贝生成初始 0 时刻的流场文件。
- (2) 运行 `decomposePar`；将计算域分割成 8 个处理单元以供并行计算。
- (3) 将算例文件上传至集群。
- (4) 运行 `openfoam.job -8 xxx pimpleDyMFoam` 开始求解；其中 `xxx` 为用户自定任务名。
- (5) 求解完成后，运行 `reconstructPar` 命令将分割开的计算域重新合并。计算完成。

5.5 结果与讨论

NREL 非定常空气动力学实验的 S 系列包含叶片在 3° 桨距角 5m 到 25m/s 风速下（间隔 1m/s）的一系列实验。本文主要对其中的 5m/s, 10m/s, 15m/s 和 25m/s 风速下的风机及其流场进行数值计算。其中 15m/s 和 25m/s 的风速对于大部分风机叶片来讲已经达到失速条件，故而对这两个风速下的数值计算是十分具有挑战性的。

本文计算的收敛情况可参见图 5-24。图中给出了 15m/s 和 25m/s 风速下（考虑高风速下计算更困难）的推力时历曲线。由图可以看到在 0.5s 之后，两者推力值都基本保持不变。故本文选取计算时间 2.6s，折合为风轮旋转 3 圈多是比较合理的。

5.5.1 涡、速度场和各截面流线分布

这四个风速下的风机尾涡的情况可以参见图 5-25。该图由速度梯度张量第二不变量 Q (the second invariant of the velocity gradient tensor)^[41] 等值面表达，并由速度值染色。由于 Q 值和速度值相关，为了更加清晰显示涡的形态，对于不同速度采用不同的 Q 值。从图中可以看到，在 5m/s 和 10m/s 的风速下，叶片附近流体大多还是附着流动。从 5m/s 到 10m/s 叶片随边附近的涡有加强。这两个风速下，我们可以观察到稳定的叶尖涡，特别是在 5m/s 的时候更为明显。此外由于叶片的根部截面从 S809 翼型迅速转化到圆形，变化剧烈，故这部分区域也观察到了十分明显的涡。此外我们也可以注意到尾涡大约在风轮后约一个叶片长度距离后迅速地消散了，这和之前本文对风机后面约一个叶片长度区域内的网格加密有关。故需要观察到更长距离的尾涡变化，需要对风机后面的网格做更大范围的加密，这当然也会大大增加计算量。而在 15m/s 时几乎整个叶片范围内都出现了非定常的流动分离，这在 25m/s 的时候表现得更加厉害。此外，在这两个风速下，叶尖涡和叶片其他区域产生的流动分离之间有十分明显的相互作用，这在 15m/s 的图中非常明显。这种相互作用会导致叶尖涡破碎，从图中我们也可以明显看到在高风速下叶尖涡的长度有明显缩短。此外，在 25m/s 的图中，我们还可以观察到一些奇怪的呈对称分布的斑状的涡。进一步观察发现，这些斑状的涡位于 AMI 交界面上，故推测是由于 AMI 交界面旋转时出现的网格间细小的间隙导致，这一点可以从下面的速度场的图上观察到。

四个风速下的速度场情况如图 5-26 所示。该图为流场中心位置的截面，用速度的值染色。从图中可以看到，5m/s 和 10m/s 的情况下，速度场相对非常均匀稳定，可以观察到叶片每次转过竖直位置对中心截面的速度场带来的周期性波动。整个速度场在三个区域的速度有明显的降低，约为 60%~80%。中心区域速度降低是由于有圆柱形轮毂的阻挡，而上下两个区域内速度的降低则是由于叶片扫过，其能量被风机吸收而致。而对于 15m/s 和 25m/s 的情况，我们首先可以看到中心区域的低速带已经基本被压缩得看不到了。叶片范围内的低速带也被明显压缩，而且形状也发生了明显的改变。这说明在高风速下，风机后面的流场有非常强烈的交互作用，流动已经非常不稳定。从图中也可以明显看到前面提到的 AMI 交界面的空隙，但是可以发现交界面左右的流场基本达到自然衔接，说明该交界面的存在对于整个流场影响很小，这也从侧面验证了 AMI 方法在处理交界面数据交互上的突出能力。

为了更加清晰地观察叶片不同区域的流动详细情况，本文将给出 2.5s 时刻（即风轮正好转 3 圈时）四个风速下不同位置截面（对于 blade3）的流线分布并由压力系数染色。从图 5-27 和 5-28 中可以看到随着风速的增加，机翼的攻角增大，流动分离现象也逐渐加强。如图 5-27 所示，5m/s 的时候，整个叶片范围的流动几乎都是附着流，相对比较平稳。相应的压力分布也呈现典型的机翼压力分布特征。而到 10m/s 的时候，各个截面位置处流动都开始分离，有比较扁平的涡生成。到了 15m/s 和 25m/s 时，如图 5-28，机翼失速已经非常严重，有严重的分离和涡的生成。这时已经能够同时观察到多个涡的生成和脱落。这时的压力分布也已经被涡的生成而打乱。

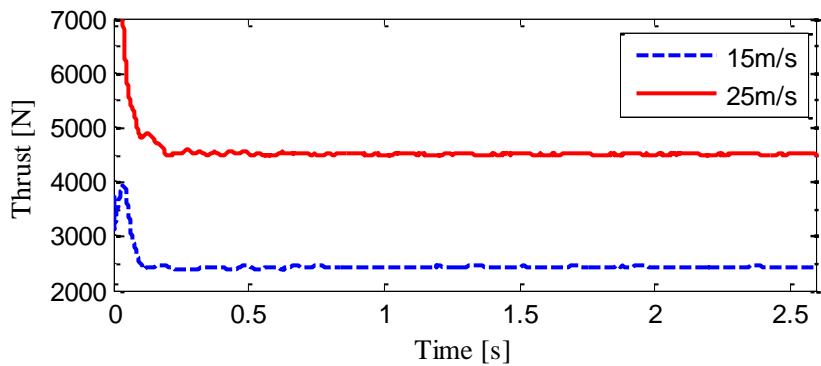


图 5-24 15m/s 和 25m/s 风速下风轮推力时历曲线

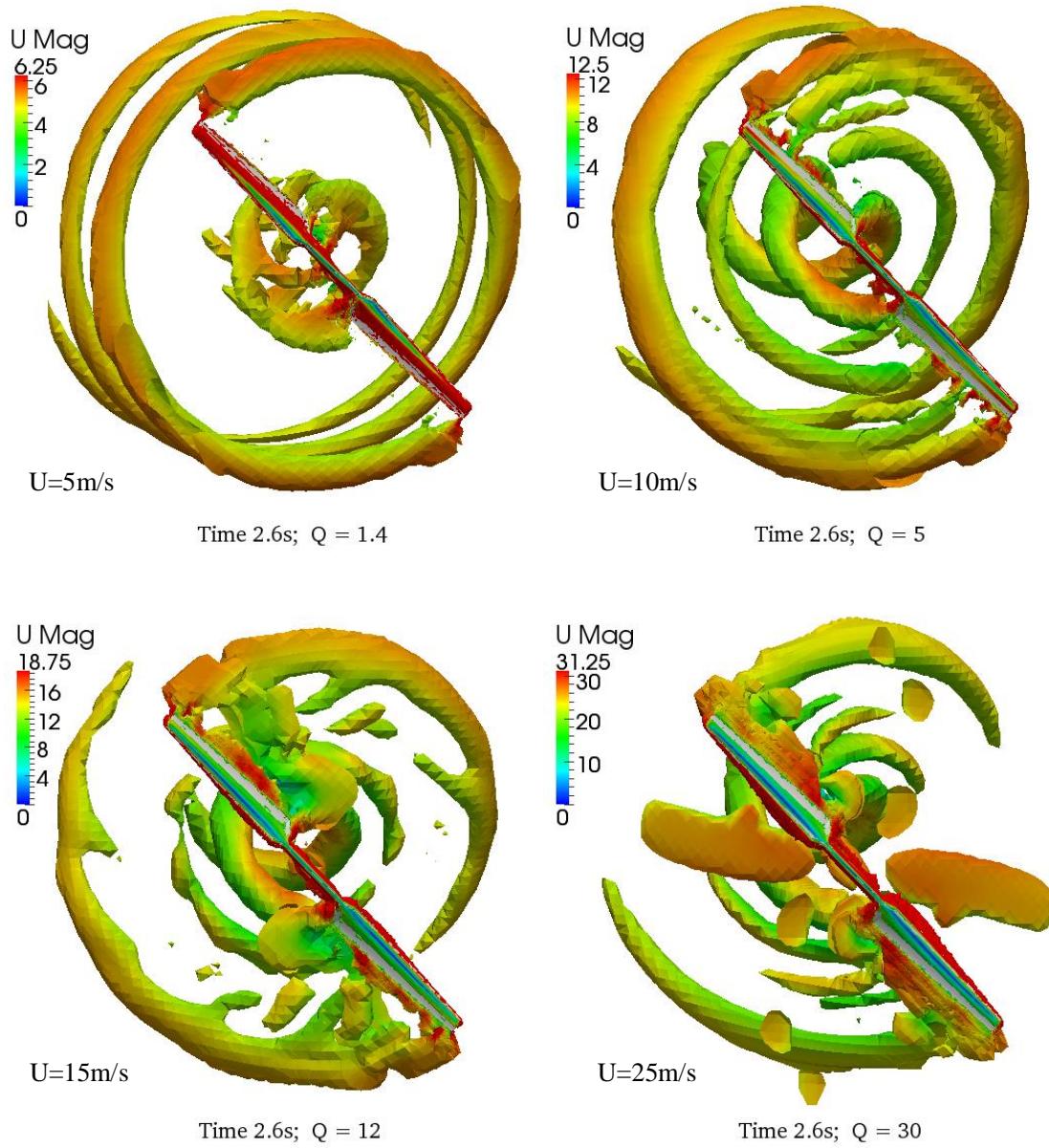


图 5-25 各风速下风轮尾涡的情况

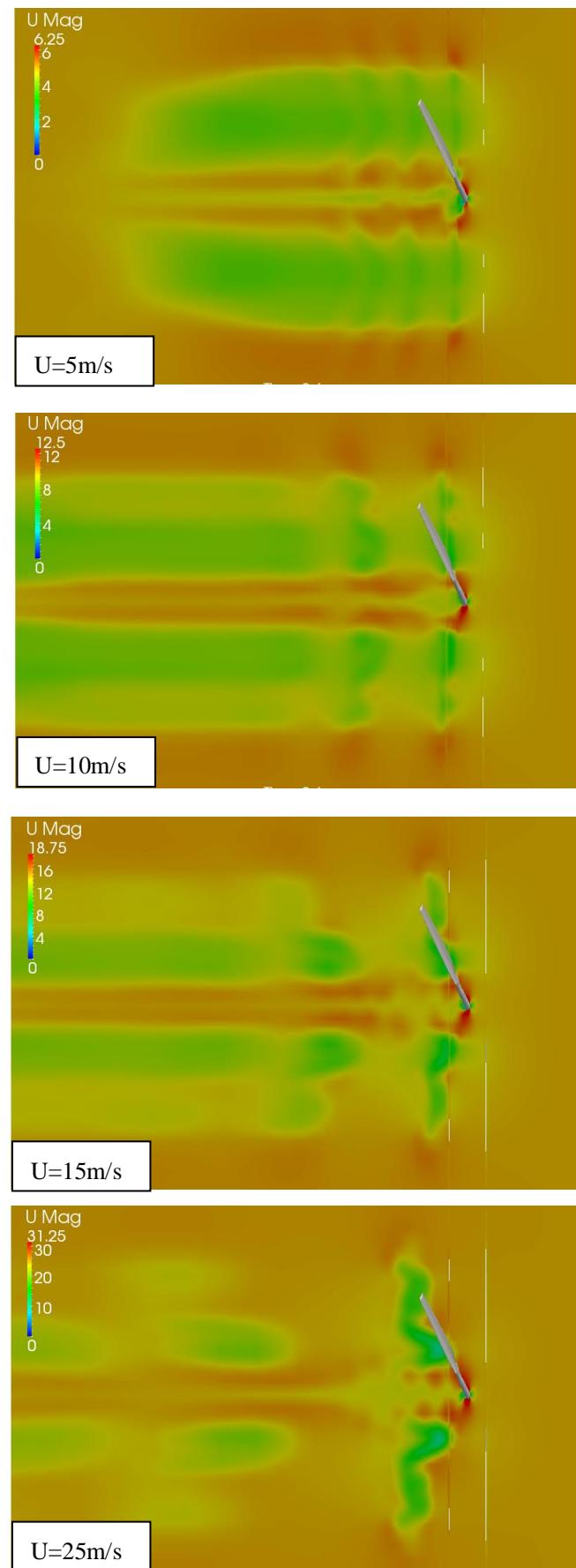


图 5-26 各风速下速度场情况 ($y=0$ 截面)

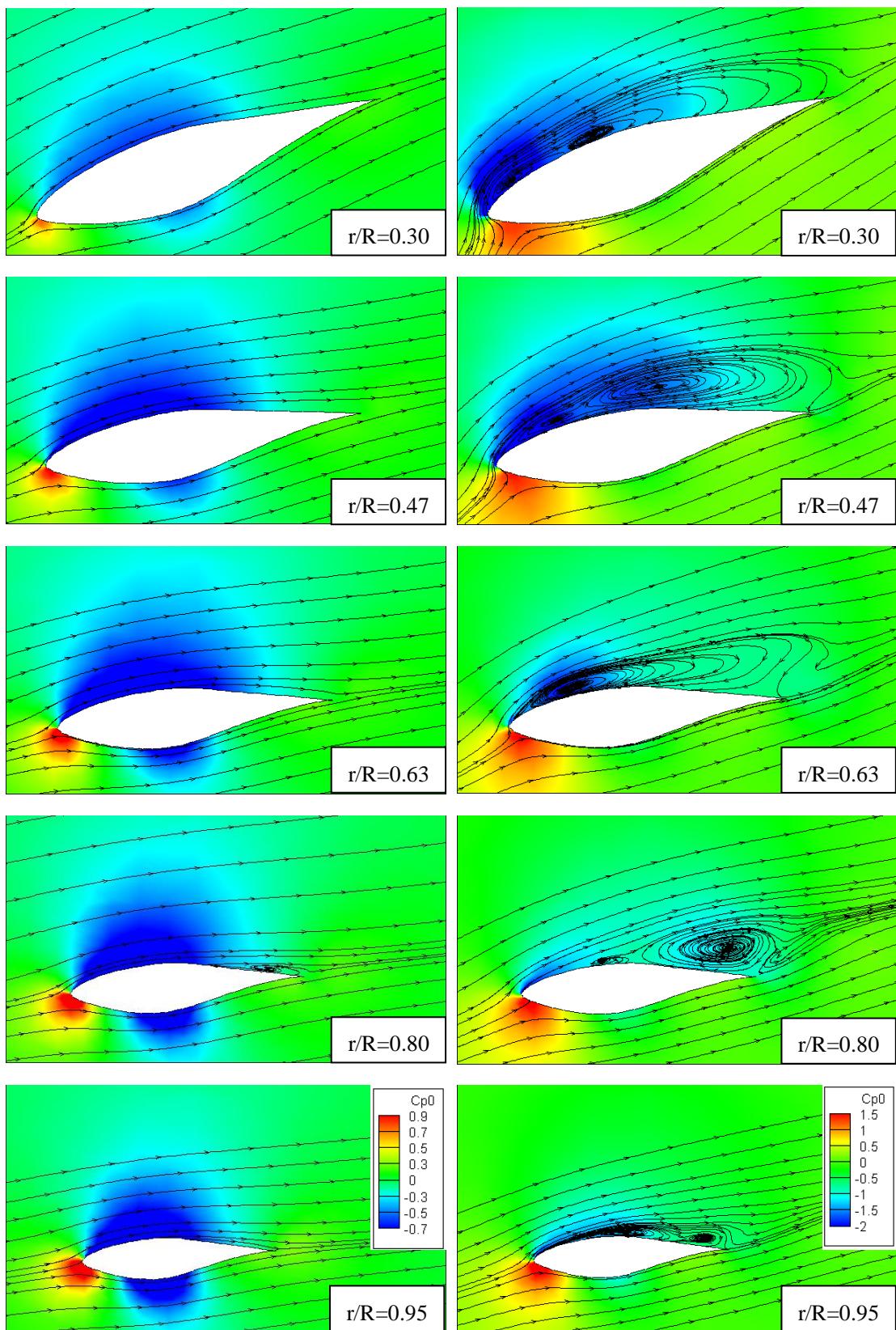


图 5-27 5m/s (左) 和 10m/s (右) 风速下叶片各个截面位置的流线分布

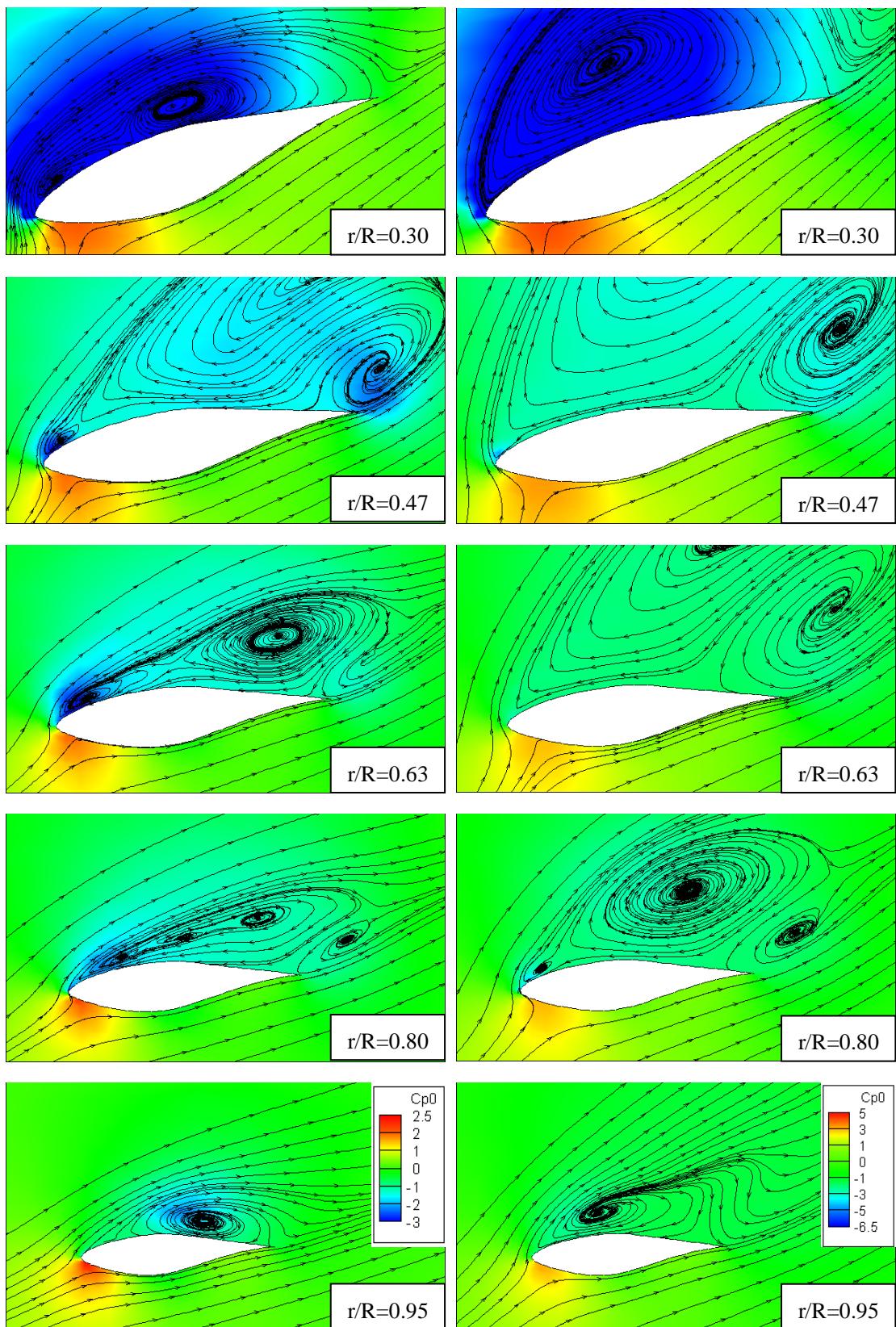


图 5-28 15m/s (左) 和 25m/s (右) 风速下叶片各个截面位置的流线分布

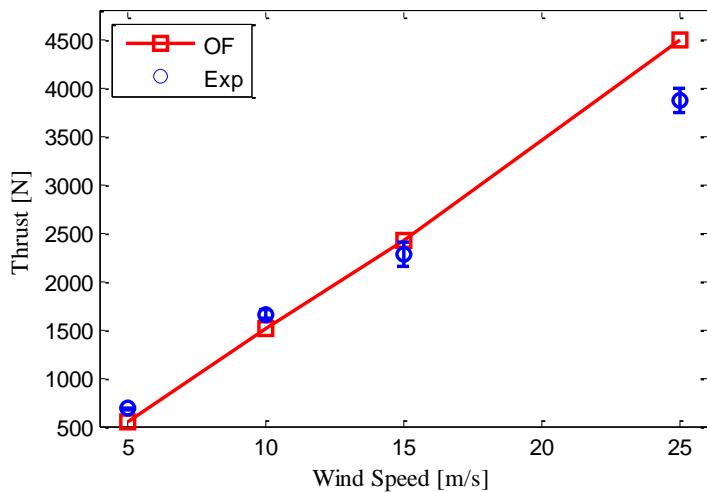


图 5-29 推力随风速变化曲线

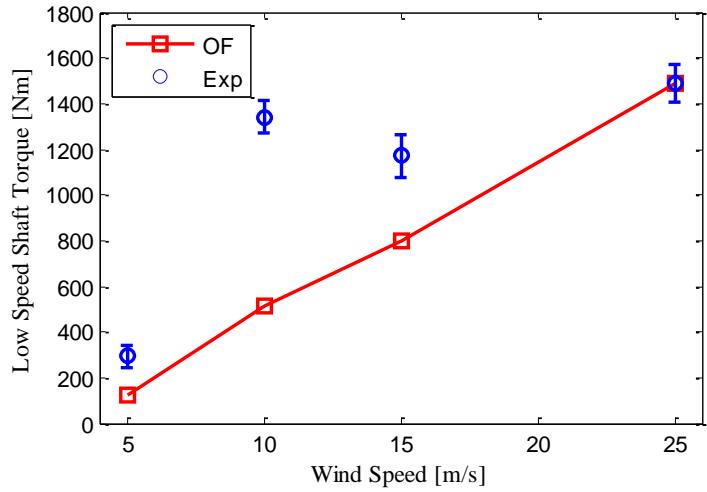


图 5-30 扭矩随风速变化曲线

5.5.2 推力和转矩

实验中的推力值是通过对压力在整个叶片表面积分得到的，故而只考虑了压力的成分而忽略了比较小的粘性成分。故本文在求解推力的时候也将只考虑压力成分。实验的转矩通过安装在实验风机低速刹车盘上的传感器测得值加上一些修正得到。

实验的推力和转矩由几个周期的值平均而得，图中的“I”形代表了实验值在一个周期内的变化，即代表土一个标准差。本文的推力和转矩值由风轮第二和第三个旋转周期的数据平均而得。

由上图 5-29 可以看到，CFD 对于风机推力的预测非常准确，基本都在实验值一个标准差范围。只有在 25m/s 的时候预测得比实验值有轻微的高出，如前所述，这个风速下风机周围流动有显著的失速和分离现象，这种状况在 Li^[21]的文章中也有提到。但是 CFD 预测的力矩的情况除了最大风速外，相比实验值有比较明显的偏低。本文在前面已经提到，不少研究者在计算扭矩时都有近 30% 的低估。但是本文 10m/s 风速下的扭矩值只有实验值的近 1/3，本文对此做了很多次的努力都没有明显的改善，具体的原因待后面更加深入地探讨。

5.5.3 C_n 和 C_t

由于推力和转矩都是通过将压力在整个叶片表面积分得到的，故只能表现出风机的整体性能。进一步研究叶片不同位置处截面的空气动力系数，可以让我们更加深入地探讨风机的空气动力性能，也可进一步验证求解器在计算风机空气动力性能细节上的能力。 C_n 和 C_t 其实依旧是通过压力积分得到的，但是和之前不同的是，这里仅对压力在特定的截面上进行积

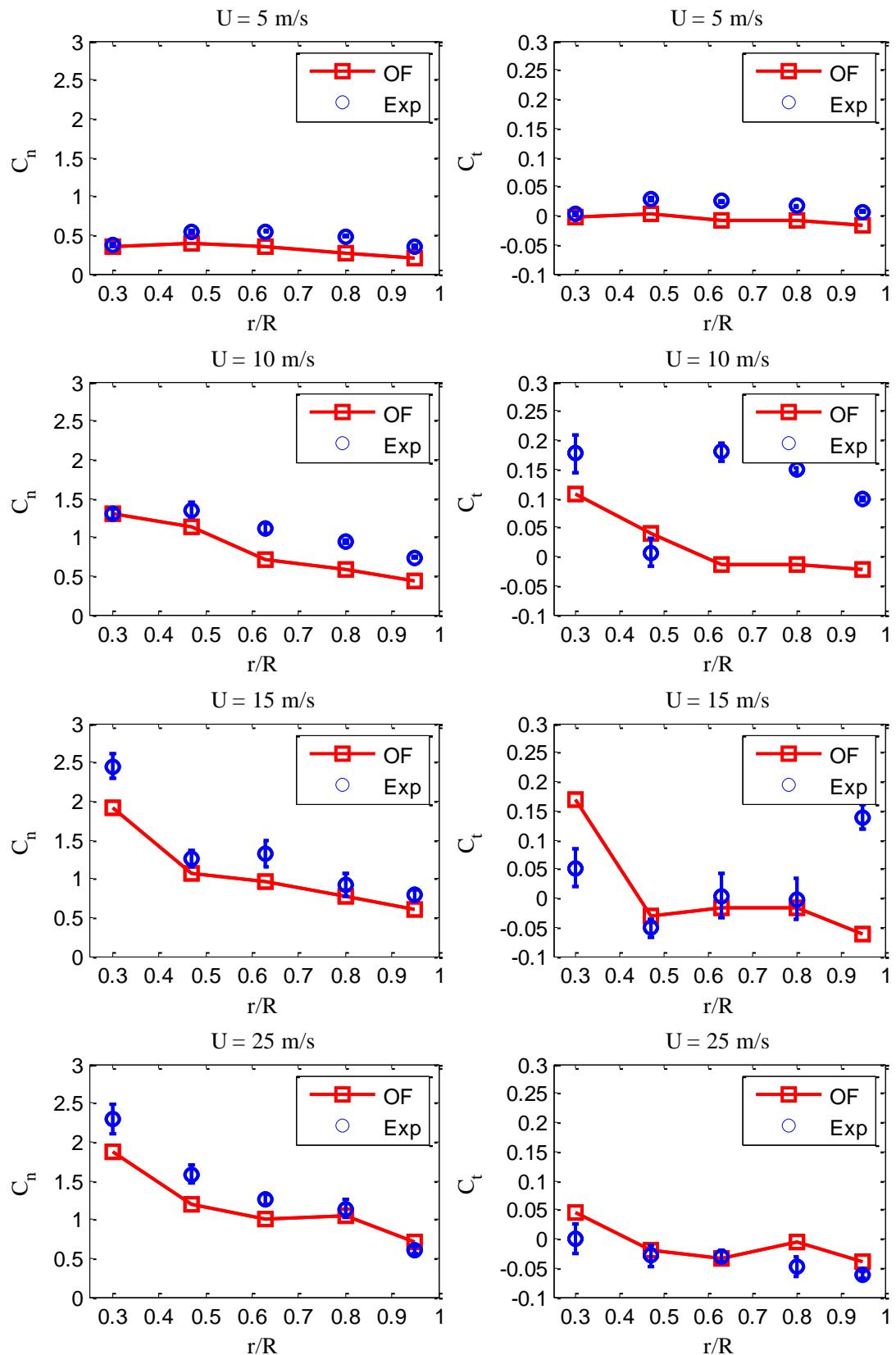


图 5-31 风机各个风速下 C_n (左) 和 C_t (右) 随截面位置的变化曲线

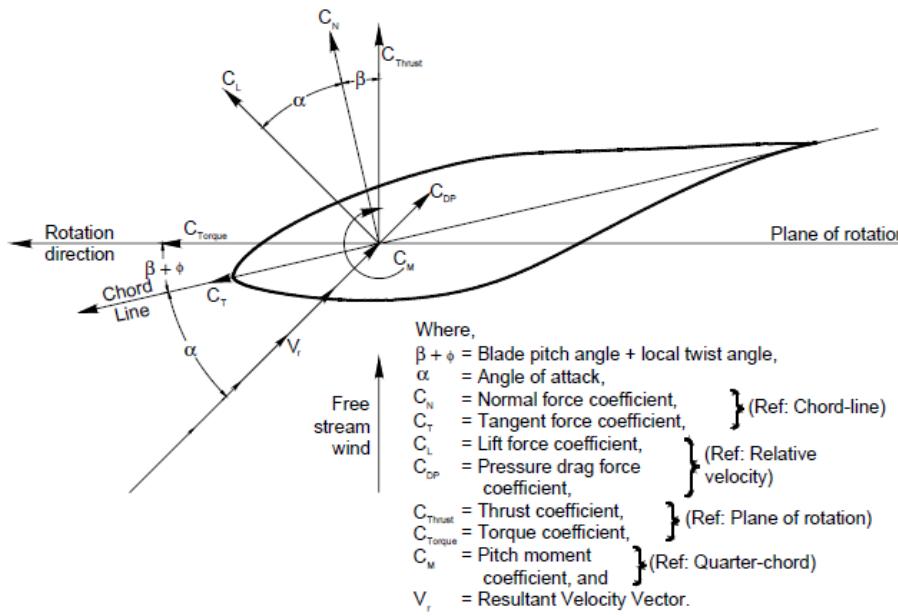


图 5-32 各空气动力性能系数示意图^[15]

分。关于机翼剖面各个空气动力性能系数参见图 5-32。

从图 5-31 可见，5m/s 和 25m/s 情况下的 C_N 和 C_T 预报比较准确，和实验值相差无几，前者是因为此风速下流动还几乎没有分离出现，而在高风速下 CFD 计算值能与实验吻合可能得益于 S809 翼型在失速状态下的特点^[18]。在 10m/s 和 15m/s 的风速下，叶片周围流体流动开始出现非定常的分离。这两个风速下的叶片不同截面位置处的 C_N 与实验值吻合较好，整体而言比实验值偏低一点，但整体趋势十分吻合。而此风速下的 C_T 计算值与实验有比较明显的偏差。10m/s 时 C_T 实验值在 $r/R=0.47$ 的位置处有比较明显的下降，这是由于叶片在该处附近流体分离点迅速移动到导边处所致^[18]。在 $r/R=0.47$ 之后，实验值又回到了原来的位置。但是本文计算得到的 C_T 在 $r/R=0.47$ 后保持在低位而没有重新升高，这可能是由于此时叶片表面的流动分离严重非定常，而使得 $r/R=0.47$ 之后的位置分离点也比较靠前所致。而对于 15m/s 中 C_T 实验值在 $r/R=0.95$ 处也有回升，而本文计算却没有回升，可能依旧是这个原因。从图 5-32 我们也可以发现， C_T 和力矩方向已经十分接近了，所以对于力矩的贡献比较大。而由上面的分析我们可知本文计算得到 C_T 在 10m/s 风速下 $r/R=0.47$ 之后，以及 15m/s 风速下 $r/R=0.95$ 位置截面处的没有像实验那样有回升，从而导致前面得到的力矩在 10m/s 和 15m/s 两处有比较明显的偏低。

5.5.4 C_p

前面讨论的 C_N 和 C_T 是 C_p 在特定截面上积分得到的。为了进一步探讨风机空气动力性能的细节，以及探讨之前所述问题的最终原因，我们需要将叶片各个截面压力分布情况与实验进行对比。相较之前，这将更加具有挑战性，因为压力值是针对某一个特定的点而言的，这也将更加明显地展示 pimpleDyMFoam 求解器计算风机空气动力性能的能力。

实验的压力值是由叶片在上半个半圆区域扫过时平均得到的，这样也除去了塔筒对于叶片压力分布的影响，比较符合本章只有风轮的情况。但是本文在实际操作时，由于后处理的限制，只选取了 2.5s 时（即风机正好旋转 3 圈时）叶片的压力情况做截面研究。本文中用到的压力系数定义如下：

$$C_p = \frac{P_0 - P_\infty}{0.5\rho[U^2 + (\omega r)^2]} \quad (4-6)$$

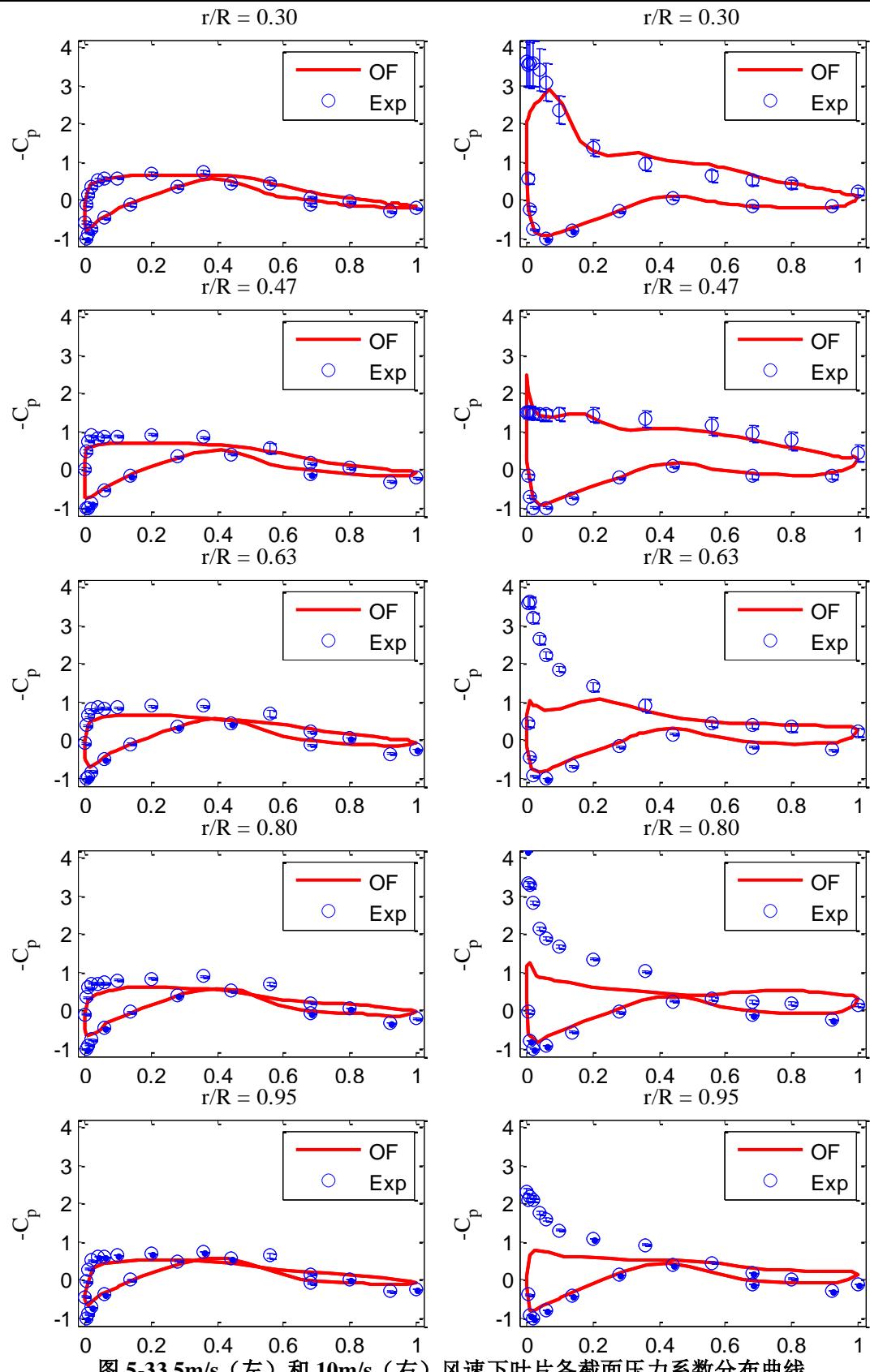


图 5-33 5m/s (左) 和 10m/s (右) 风速下叶片各截面压力系数分布曲线

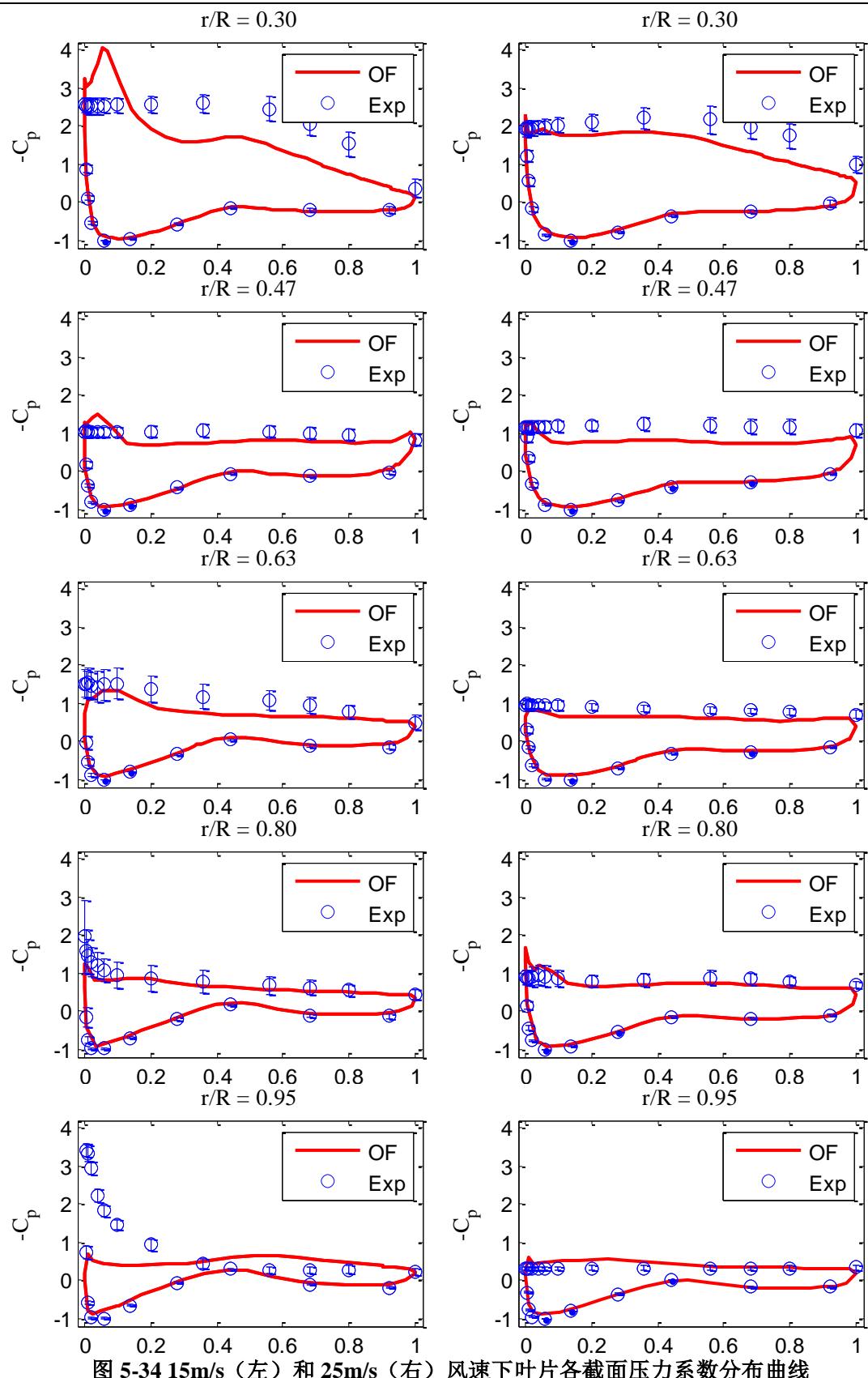


图 5-34 15m/s (左) 和 25m/s (右) 风速下叶片各截面压力系数分布曲线

式中, P_0 ——叶片表面附近的压力值, Pa; P_∞ ——无穷远处的压力值, 在本文中为 0Pa;

U ——风速, m/s; ω ——风轮角速度, rad/s; r ——截面距转动中心的距离, m。

通过图 5-33 和 5-34 可以看到, 5m/s 和 25m/s 风速下, 各个截面的压力系数分布与实验值都非常吻合, 这也进一步验证了之前的说法。而在 10m/s 的风速下, 前两个截面与实验值吻合得还好, 但是靠近叶片尾部的另外三个截面压力系数的曲线的峰值都远远没有达到实验值。仔细观察这些图可以发现, 偏差比较大的区域位于叶片吸力面 (suction side) 靠近导边的位置。而同样是叶片导边处的压力面 (pressure side) 的压力系数值就完全能够达到实验值。这也就解释了为什么之前计算 C_t 时在这几个位置得到的数值与实验值有明显的偏小。我们再看 15m/s 时, 也可以发现 $r/R=0.95$ 时压力系数曲线在叶片吸力面导边处也没有捕捉到峰值, 这也就导致了之前计算 C_t 时在 $r/R=0.95$ 时出现的低估。这些位置压力系数的峰值的低估也就直接导致了一开始力矩计算时出现的在 10m/s 和 15m/s 时偏低的情况。此外对于 15m/s 的 $r/R=0.3$ 截面下, 实验得到的数值要比 CFD 计算得到的更加平缓, 这种现象在 Li^[21] 的文章中也有提到。

5.5.5 关于计算误差的进一步讨论

通过之前的讨论我们已经发现, 本文计算得到的 10m/s 和 15m/s 处的扭矩值与实验相比有明显的偏小。进一步讨论发现, 在 10m/s 时 $r/R=0.63, 0.8$ 和 0.95 这三个截面处以及 15m/s 时 $r/R=0.95$ 截面处的 C_t 没有出现如实验值那样的升高, 而一直保持低位。再进一步分析这些位置的压力分布可以发现, 上面所述位置处的压力系数在叶片吸力面靠近导边位置处都比实验偏低很多。

再观察本文得到的所有情况下的压力系数分布曲线, 我们可以发现在叶片吸力面靠近导边位置, 本文的计算值都比较平缓, 而没有出现其他作者^[18, 21]计算结果中的峰值。仔细查看流线图 5-27 和 5-28 中的压力染色区域, 我们不难发现在叶片导边处的压力变化不太均匀。此外, 与本文采用同一网格, 但是选用基于多参考系(MRF)求解器 MRSimpleFoam 对本章算例进行计算, 计算结果也没有捕获到导边处的压力峰值^[42]。本文作者推测可能是叶片导边处的网格所致。

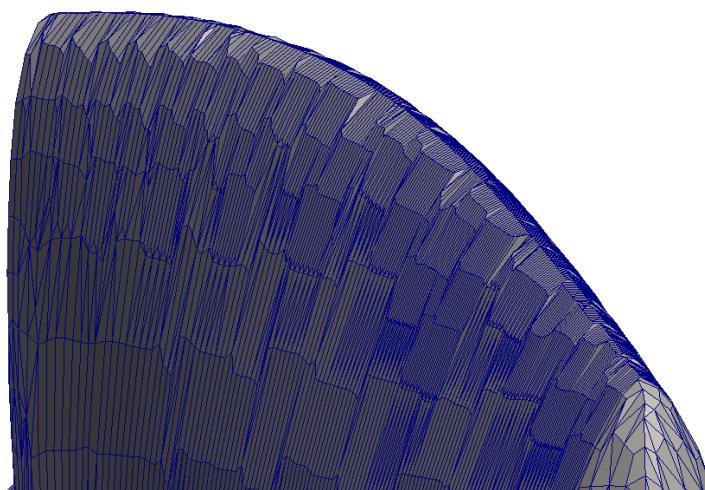


图 5-35 叶片导边处网格情况

叶片导边处的网格如上图 5-35 所示。从图中我们不难看到, 该处的网格面不是很平整, 有比较明显的起伏。这和该处的网格质量有极大的关系。虽然导边处的网格相比叶片中心区域有明显的加密, 但是单个网格单元尺寸还是过大, 导致叶片的导边不平整。我们知道对于 S809 翼型, 在导边处有一段曲率的非常大而尺寸有很短的曲线衔接上下边。这就使得在叶片的导边处很小的一段距离内有非常剧烈的曲率变化。故若想要精确捕捉该处的曲线及曲

面，需要放置非常小的网格单元。

而对于 `snappyHexMesh` 方法，由于网格的精度是通过背景网格一级级分裂加密得到的。故可想而知，对于这种小范围大曲率的变化，`snappyHexMesh` 也只能通过一段段的直线段去捕获，因此导致叶片导边处网格不平整，如上图所示。当然，我们可以通过进一步加密该处的网格来提升导边的捕捉效果。本文对此也做出过不少的尝试，包括对导边处网格加密级数提升以及背景网格在叶片导边附近加密（此处要注意网格的细长比小于 2，不然用 `snappyHexMesh` 划分所得网格质量严重下降）等等。以 10m/s 为例，以两者的计算结果来看，转矩有近 $200N \cdot m$ 左右的提高，可推知导边处的网格质量确实对结果有比较重要的影响。但是对于本文，采用这种方式加密后，计算量由比较大的增加，计算时间变得不能接受，故而放弃这种方式。

究其原因，其实还是 `snappyHexMesh` 算法本身对于物体边界的捕捉能力有限所致。本文之前也提到的单纯用 `snappyHexMesh` 生成网格时得到的 AMI 交界面质量也因为太差而无法进行计算，后来是由 ICEM-CFD 重新划分背景网格解决的。所以本文认为，`snappyHexMesh` 对于物体的边界以及大曲率区域的捕捉和网格划分能力还有待进一步提高。就目前状态，对于本研究内容若想要获得更好的结果，本文建议采用其他专业网格划分软件划分网格，如 ICEM-CFD 等，本文由于时间精力有限，不再对此作出探索。

此外，通过之前的论述我们也可以看到 10m/s 和 15m/s 风速是流动从稳定向不稳定的过渡时期，流动状态相对更为复杂，这种状态的数值模拟本身难度较大。在这种情况下对流项的作用会增大，考虑提升对流项的差分精度可能对计算结果会有所改善。但 OpenFOAM 暂未提供对流项的高阶差分格式，故本文对此也不再做进一步尝试。

5.6 本章小结

本章主要对 NREL phase VI 风轮进行了建模与数值计算。本章给出了对 NREL phase VI 原始数据的处理过程以及运用 CATIA 软件建立整个风轮的过程，详细描述了叶片过渡区域以及尖端的绘制方法。而后本章对风机运用 `snappyHexMesh` 进行网格划分，遇到了网格扭曲过大以及 AMI 交界面网格质量过差的问题，通过运用 ICEM-CFD 软件绘制背景网格解决。之后本文使用基于 OpenFOAM 的不可压 RANS 求解器 `pimpleDyMFoam` 结合 AMI 网格运动处理方法对风轮在四个风速下进行了数值计算，并将结果与实验展开深入对比。

本章得出了各个风速下风轮的尾涡情况，速度场的分布情况，推力曲线及扭矩曲线，不同截面位置的 C_n 、 C_t 曲线以及 C_p 的分布曲线和各截面的流线情况等丰富的结果。求解结果与实验值有很好的吻合，但是 10m/s 和 15m/s 时计算得到的扭矩值有明显的偏低。本章通过对计算结果逐步深入的分析，最后推测是由于叶片导边处网格质量比较差导致。这也说明了 `snappyHexMesh` 在处理大曲率边界上能力的限制。

第六章 NREL Phase VI 整机模型数值计算

前面一章已经介绍了对于 NREL phase VI 单个风轮的空气动力性能和流场的数值计算。从中本文发现可能由于 snappyHexMesh 在网格划分能力上的限制，导致叶片吸力面靠近导边处压力系数偏低，进而导致了最终的扭矩偏低。由于本章将采用与上一章相类似的网格划分和处理方式，故而也无法避免上一章的这些结果，这里不再赘述。本章主要关注轮毂、机舱和塔筒等几何物体对风机叶片的空气动力性能和风机周围流场产生的影响，并对此进行深入分析。

6.1 建立几何模型

风轮部分的几何模型参考前一章即可，本节主要关注轮毂、机舱和塔筒的模型建立。相关尺寸参数参见 NREL 实验报告^[15]。

6.1.1 建立轮毂和机舱模型

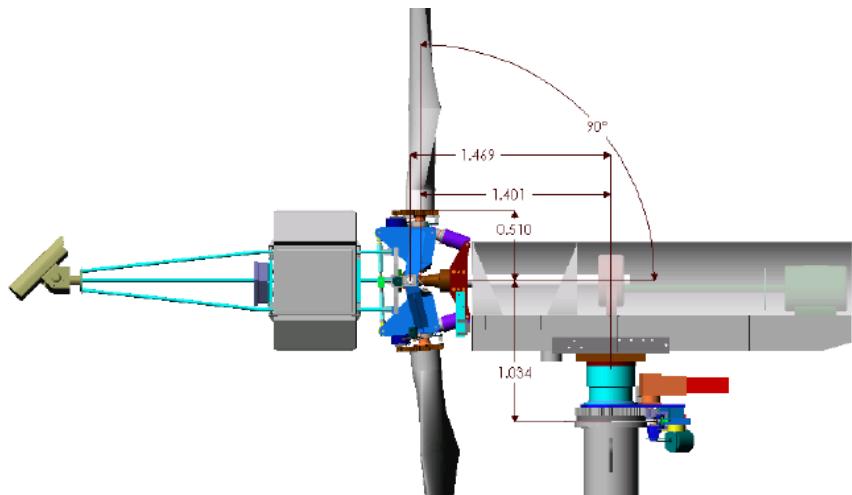


图 6-1 实验风机轮毂与机舱结构侧视图

实验风机轮毂及机舱的结构如上图所示，可见二者之间有很多的测量装置，而且实验也没有给出这部分具体的几何数据。故本文对轮毂和机舱做简化处理，如下图所示。

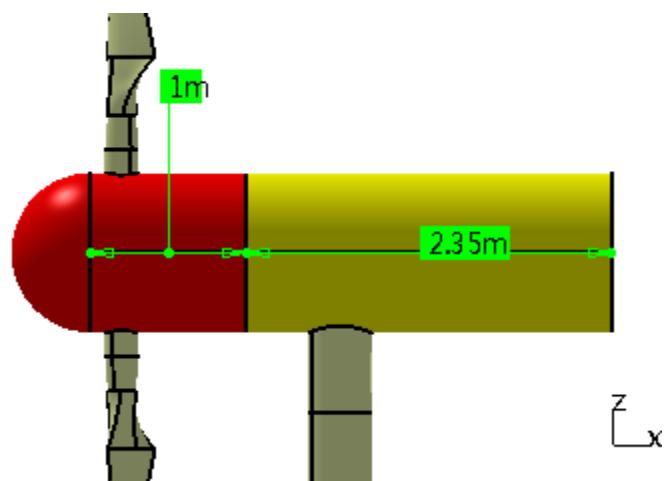


图 6-2 轮毂及机舱结构尺寸图

简便起见，本文将风机的轮毂和机舱绘制成一个半球面加上一个圆柱面的结构，二者半径相同，均为 0.51m，如上图 6-2 所示。图中红色部分为轮毂区域，可以带动叶片一起做旋转运动。其中圆柱部分长 1m，半球面的中心坐标为 (-0.2m, 0m, 0m)；后面黄色区域为机舱区域，和塔筒一起保持静止，全长 2.35m。

6.1.2 建立塔筒模型

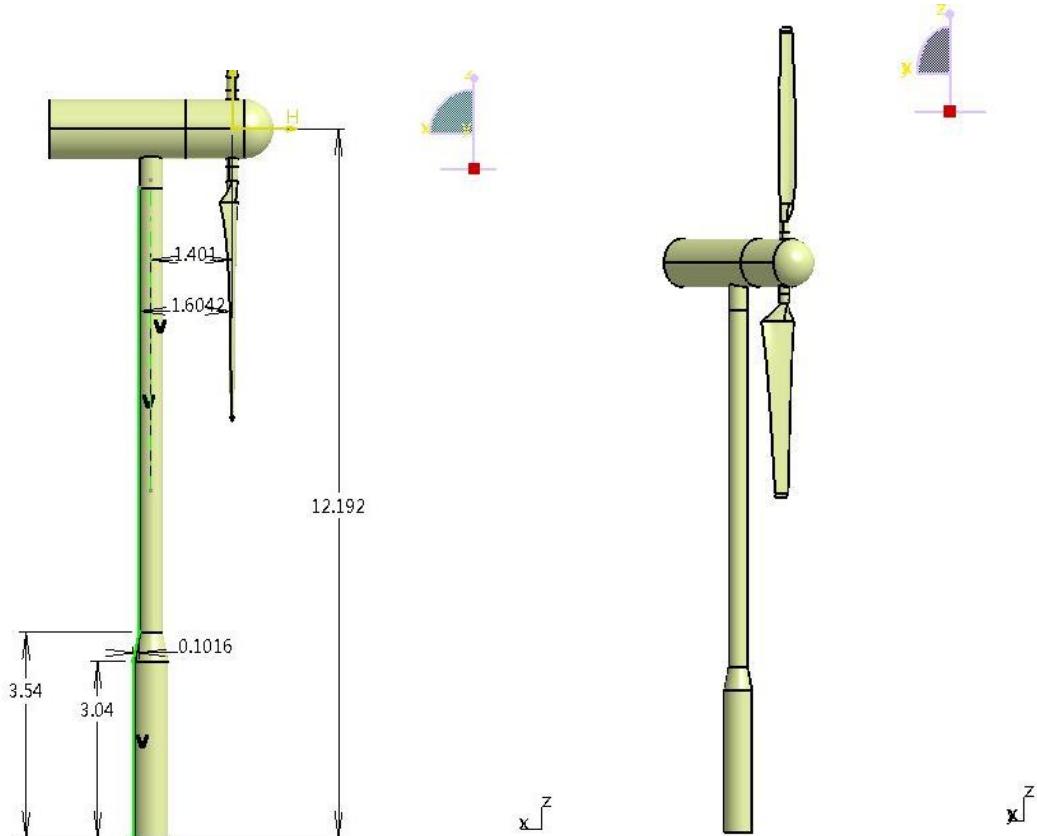


图 6-3 塔筒尺寸图

图 6-4 NREL phase VI 风机效果图

风机塔筒模型以及相关的尺寸参数如图 6-3 所示。由图可见，风机的轮毂中心距地面高度为 12.192m。塔筒由三段组成：下部为较粗的底部段，高度 3.04m，半径 0.3048m；上部位较细的顶部段，上端连接机舱，下端距离地面高度为 3.54m，半径为 0.2032m；中间区域为过渡圆台面。塔筒中心线与设定的坐标原点（即叶片转动中心点）x 方向距离为 1.401m。

至此 NREL phase VI 整机模型建立完成，如图 6-4 所示。

6.2 网格绘制

和前一章方法类似，本章也采用 ICEM-CFD 软件绘制背景网格，然后再利用 snappyHexMesh 生成全套网格，只需对之前的网格拓扑结构做出细微调整即可。具体的参数设置和上一章中的单个风轮情况相同。网格数量如下表所示：

表 6-1 网格数量变化情况

背景网格（万）	网格分裂后（万）	添加网格层后（万）	最终网格量（万）
25.5	50.6	65.6	65.6

整机模型的网格情况参见下列各图：

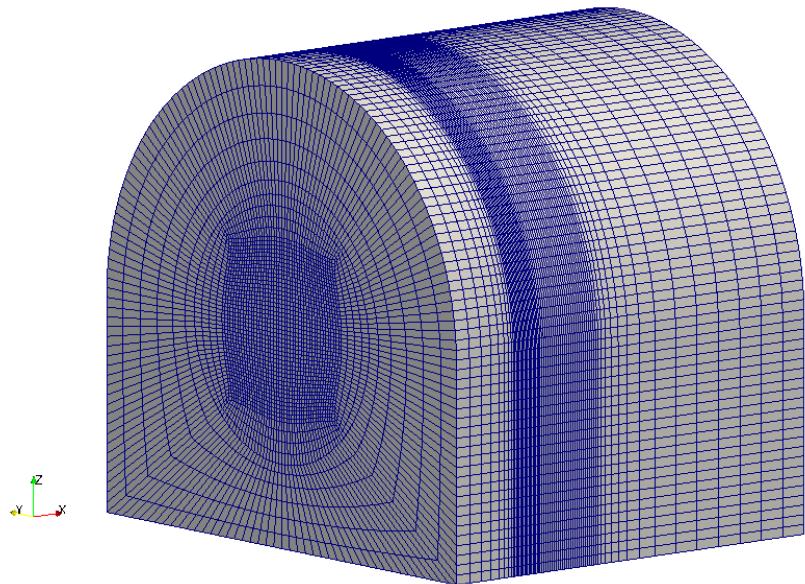


图 6-5 背景网格

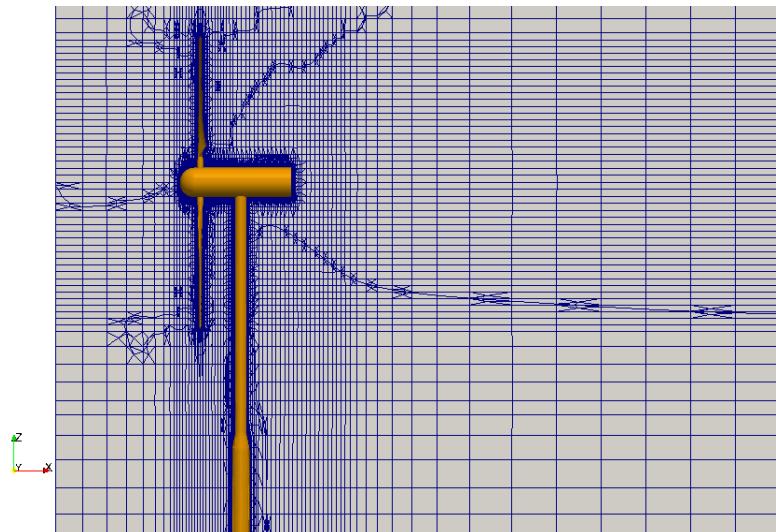


图 6-6 计算域网格侧视图 ($y=0$ 截面)

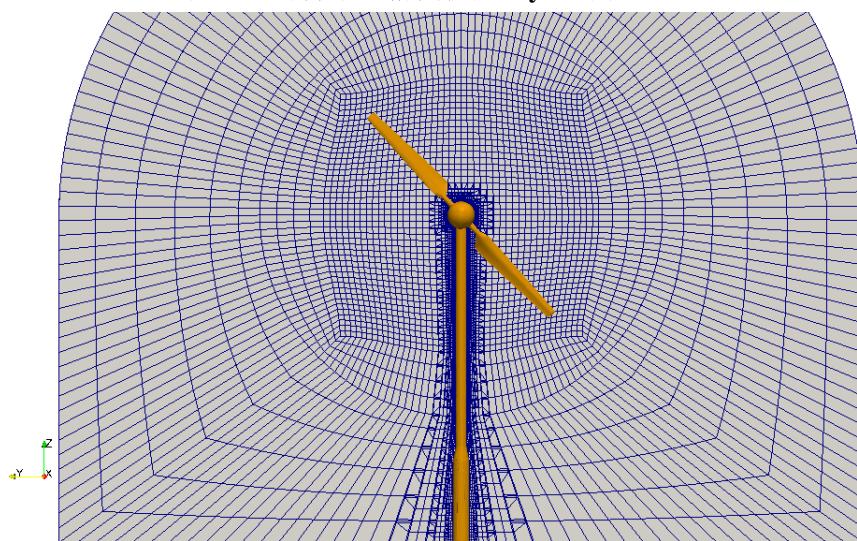


图 6-7 计算域网格正视图 ($x=1.401$ 截面)

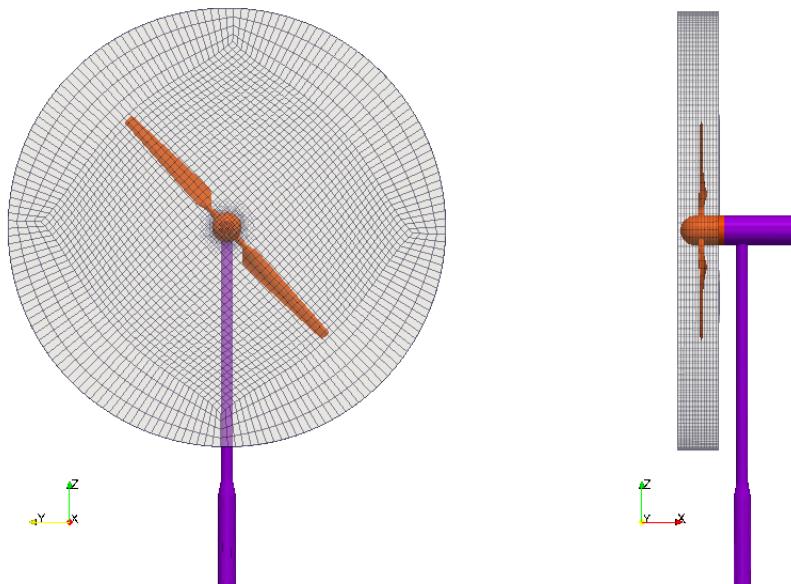


图 6-8 AMI 交界面网格情况正视图（左）、侧视图（右）（ $t=2.6s$ ）

6.3 其他参数设置

本文中其他参数的设置与前一章中单个风轮算例的设置基本相同。其中整机结构多出的轮毂、机舱及塔筒结构的初始边界条件均与风轮相同，即当成壁面(wall)设置即可。唯一有差别的是 AMI 交界面的区域前后范围有所改变。之前的风轮 AMI 交界面前后范围为[-0.8m, 0.8m]。整机模型设计时也是按照该尺寸进行的，参见图 6-2 即可得知。但是在网格划分结束后发现，原来设定的交界面右侧边界即 0.8m 处网格受到了塔筒网格影响。为了保证计算稳定，本文将 AMI 交界面的前后范围调整至[-0.8m, 0.6m]，如上图 6-8 所示。

6.4 结果与讨论

6.4.1 涡

图 6-9 给出了四个风速下风机的尾涡情况。和之前一样，该图由速度梯度张量第二不变量 Q 等值面表达，并由速度值染色。从图中可以看到，和之前只有风轮时一样，5m/s 时叶片附近基本是附着流动；而 10m/s 时，叶片随边附近兴涡范围有扩大，两者都可以观察到稳定的叶尖涡和叶根涡。到 15m/s 时整个叶片随边处都开始泄涡，叶尖涡与叶片的流动分离相互作用加强，导致叶尖涡破碎。和之前不同的是，塔筒的存在使得叶尖涡和叶根涡在经过塔筒时被切断。此外塔筒上半段有明显的涡生成，而且塔筒产生的涡和叶尖涡发生了明显的相互干扰，这在 10m/s 和 15m/s 时更加明显。15m/s 时塔筒后面的涡和在叶片产生的涡干扰下有明显变形和断裂。25m/s 时塔筒后整个叶片范围内的涡都与叶片的流动分离产生相互作用。此外，塔筒的下半段没有明显的涡生成，参见图 6-7 可知，这是由于该处的网格已经变稀疏的缘故。另外，由于轮毂和机舱位于风轮旋转中心，而且二者结构简化处理后近似为子弹型，长度方向又指向来流方向，故而该也处没有明显的涡的生成。对比图 6-9 和之前只有风轮结构的涡结构图，我们也可以发现轮毂和机舱的存在限制了叶根涡向旋转中心的扩散。

为了进一步探讨塔筒产生的涡与叶尖涡的相互作用，本文以 10m/s 风速为例，取风机下方叶片 $r/R=0.95$ 位置的水平截面三个时刻（2.275s、2.35s 和 2.425s）进行研究，如图 6-10 所示。图中背景以速度值染色，涡由 Q 等值面表达并由 Q 染色（取 $Q=0\sim20$ 分 50 级），白色线条为该截面的网格情况。首先观察速度场，可以看到塔筒前后的速度值有明显下降，而塔筒后的速度场有明显的波浪形变化，这是典型的圆柱绕流的速度场。我们再看 2.275s 时

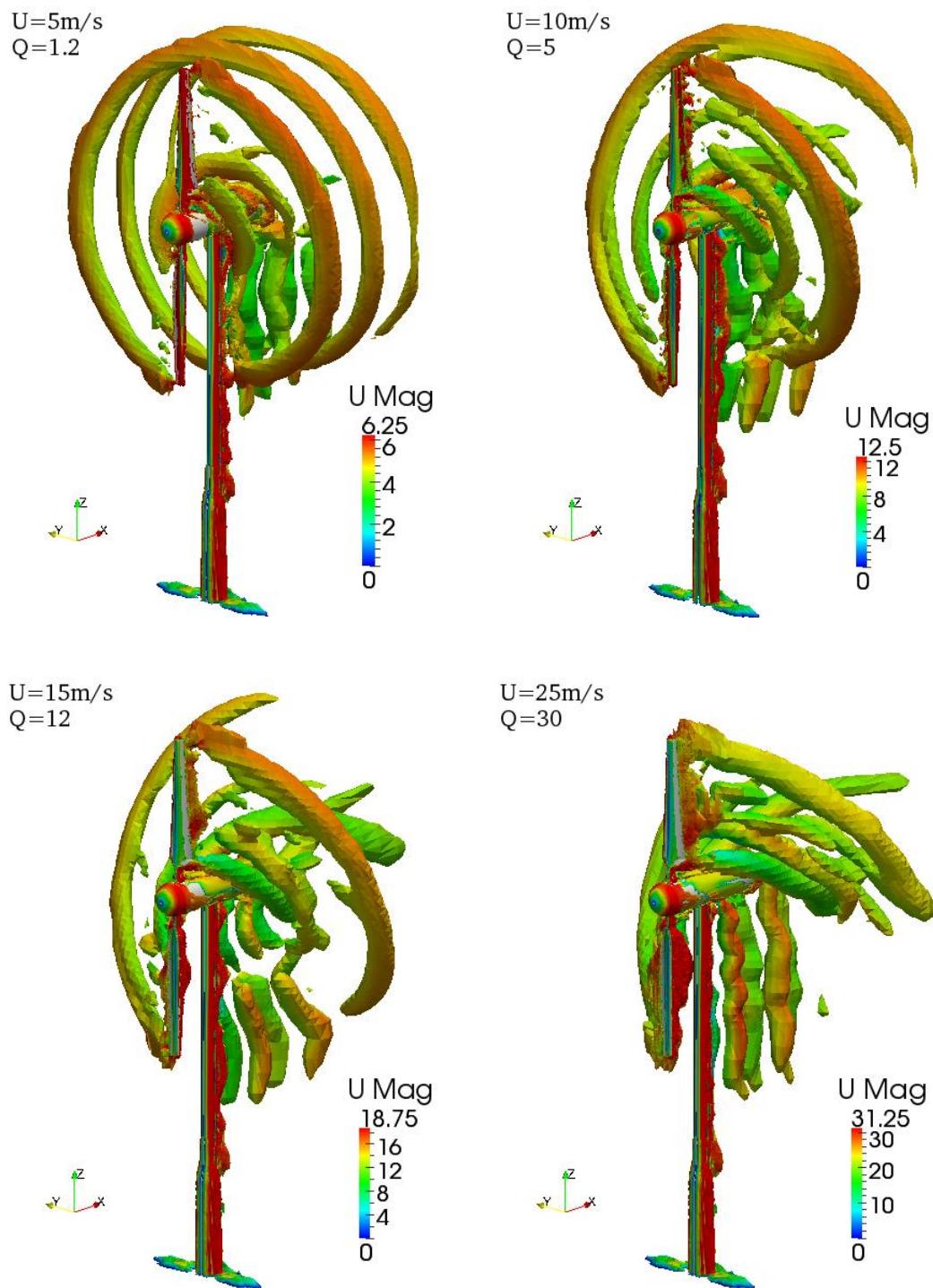


图 6-9 各风速下风机尾涡的情况

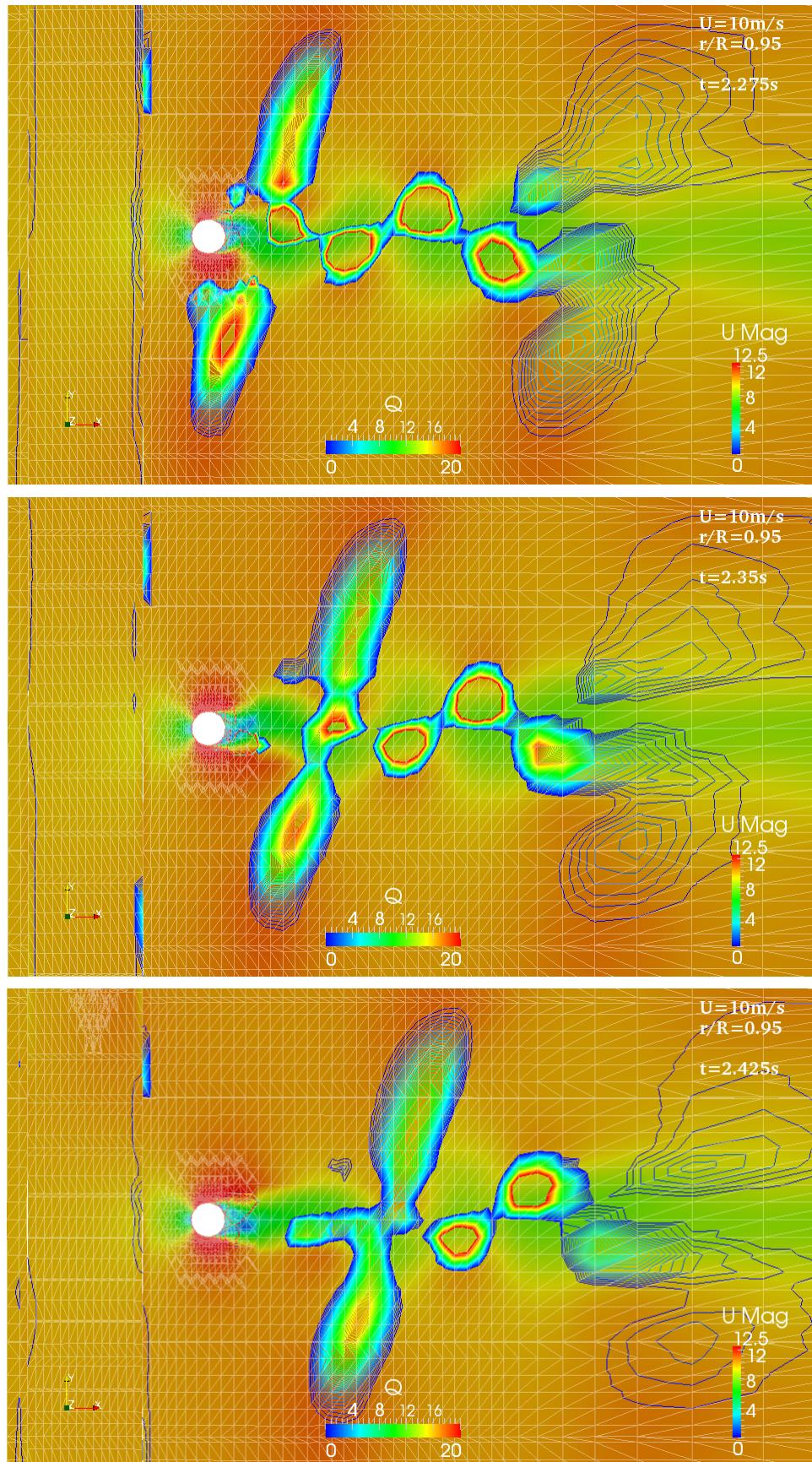


图 6-10 10m/s 风速下风机下叶片 $r/R=0.95$ 截面不同时刻的尾涡情况

的涡结构，可以看到塔筒后方流场呈现一上一下周期性的泄涡，形成典型的卡门涡街(Kármán vortex street)。该时刻可以看到叶片划过该截面时留下的上下两个叶尖涡截面，上方的叶尖涡已经和塔筒的一个尾涡接触。到了 2.35s 时，上下两个叶尖涡与塔筒的尾涡都相互接触，三者相互干扰，使得塔筒尾涡产生变形。到了 2.425s 时，上下两个叶尖涡已经连接在一起，而中间的塔筒尾涡已经与叶尖涡融合不见了。由此我们验证了之前提到的叶尖涡与塔筒尾涡之间相互作用的论断，故塔筒的存在对风机的尾涡会造成明显的影响。而从这三张图也可以清晰地看到尾涡在发展到塔筒后一个叶片距离时由于该处网格变稀疏而耗散。

6.4.2 推力

从上小节我们已经看到，塔筒的存在对于风机的尾涡有明显的干扰现象，本节我们将进一步探究塔筒存在对于风机推力的影响。图 6-11 展示了一个旋转周期内叶片所受推力的变化，和之前单个风轮不同，这里的推力由单个叶片上的压力积分后乘以 2 得到，这样就避免了按照两个叶片处理时平均并弱化所要观察的现象。对于整个风机的算例，从图中我们可以明显观察到叶片塔筒的相互作用(blade-tower interaction)，推力在叶片旋转到 180 度(即从最上方旋转到最下方)时，推力值有明显的减小，在图上形成一个谷；而叶片继续旋转时，推力值又恢复到原位。而单个风轮的算例则在整个旋转周期内都是处在一个比较平稳的位置。此外，随着风速的增加，该推力的减少极值占平均推力的比例有明显的下降，5m/s 时约有 9%，到 10m/s 下降为 6% 左右，而到 15m/s 和 25m/s 时则进一步降到了 4% 和 2% 左右。这说明随着风速的增加，推力减额极值的增加幅度小于推力的增加幅度。从图中我们也可以看到，推力的减少仅位于叶片在塔筒附近相对较窄的区域内。由此我们可以推知，对于上风向风机，叶片塔筒相互作用导致的推力的减少对于整台风机的推力影响不大。但是，我们必须意识到风机叶片推力的减少具有非常显著的周期性，故会对叶片带来一个固定频率(即风机旋转频率)的振动，设计时应该让风机相关部件固有频率避开这一值。

6.4.3 C_p

塔筒存在对于风机空气动力性能的影响最根本的还是由于其影响了叶片的压力分布，故本节选取 5m/s 和 10m/s 的单个风轮和整个风机算例的五个剖面位置的压力分布曲线做比较，其中整个风机算例的压力分布选取叶片位于最下方时刻。从图 6-12 可以看到，整个风机算例两个风速下各个截面的压力系数和之前单个风轮算例的曲线非常接近，尤其是 5m/s 时几乎没有差别。这首先是由于压力系数正交化处理而弱化了这种差别，其次在上小节也提到这种推力的减少占平均推力比例其实不大的缘故。而在 10m/s 时我们可以相对稍微明显地观察到整机算例叶片吸力面的压力值要比单个风轮算例的偏小，这也就导致了整机算例该时刻叶片推力的减少。

6.4.4 塔筒压力分布

前面主要探讨了塔筒存在对于风机叶片空气动力性能的影响，这种影响肯定是相互的，这里我们就探究下叶片对于塔筒压力分布的影响。本文选取 5m/s 和 10m/s 风速下，三个典型时刻(2.35s、2.425s 和 2.5s) 塔筒前缘的压力分布曲线，如图 6-13 所示。先观察 5m/s 的情况，我们发现对于不同的时刻，叶片对于塔筒压力分布的影响仅集中于叶片长度范围内，而该范围外的塔筒压力分布始终没有明显的变化。从图中我们可以看到，即便在 2.35s，叶片离塔筒角度超过 60° 时，位于叶片扫掠范围内的塔筒压力已经有明显的减小，这个现象也有其他研究者观察到^[43]。当叶片转到与塔筒重合位置时，塔筒该区域的压力显著减小，在压力曲线上呈现明显的低谷，甚至到达了负值。而对于 10m/s 的算例，在其他两个时刻叶片扫掠范围内塔筒压力的减小不是很明显，压力呈直线状分布；而当叶片与塔筒重合时，压力曲线也出现了非常明显的低谷。这充分说明了叶片对于塔筒压力分布有明显的周期性影响，当叶片与塔筒重合时会造成该处塔筒位置压力急剧下降。考虑到实际情况时，这种压力的下降可能导致叶片向塔筒产生变形并发生碰撞，故在设计时需要额外注意。

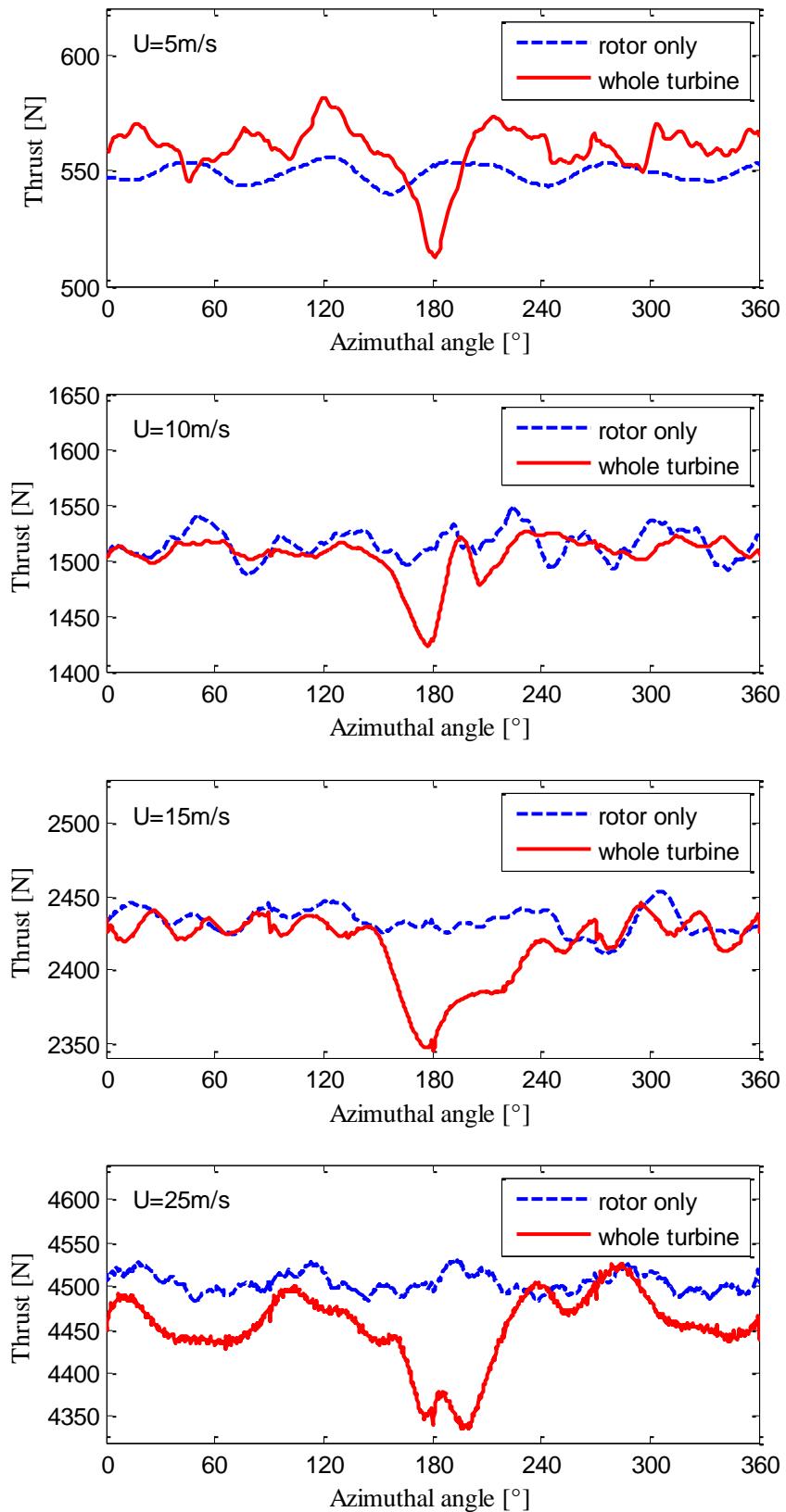


图 6-11 各风速下风机旋转一周推力变化曲线

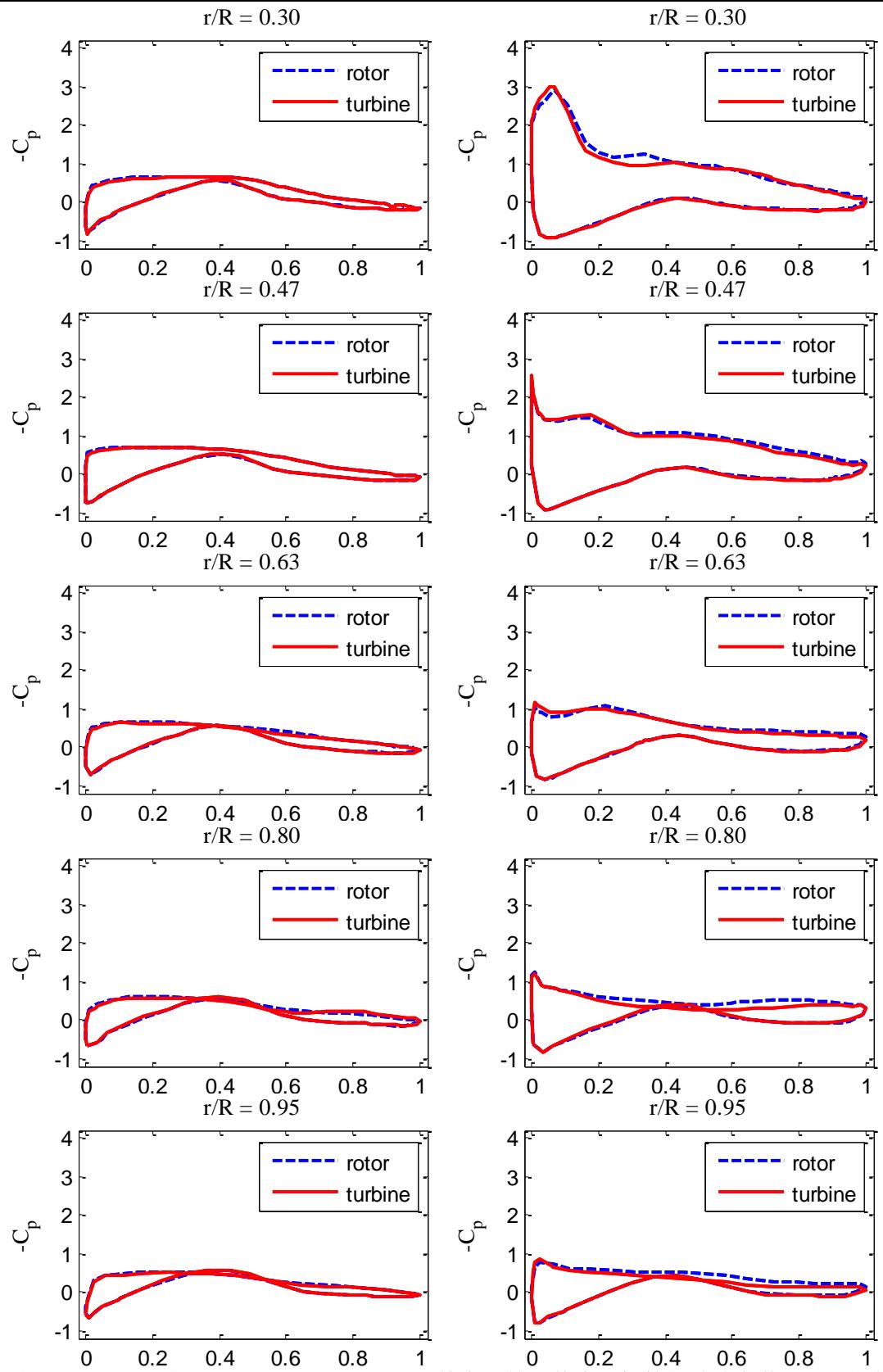


图 6-12 5m/s (左) 和 10m/s (右) 风速下单个风轮和整个风机算例叶片各截面 C_p 分布

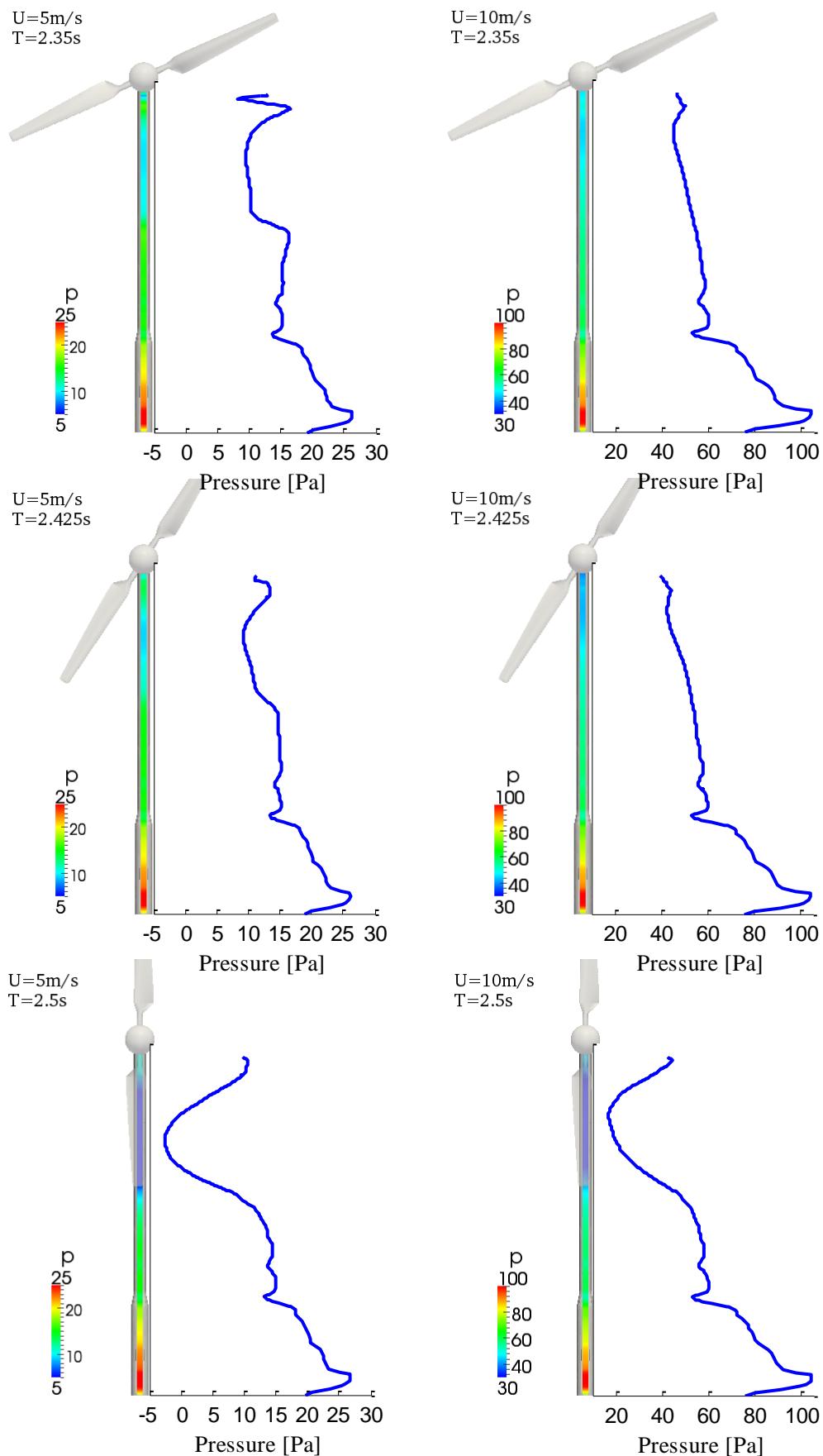


图 6-13 5m/s (左) 和 10m/s (右) 风速下塔筒在不同时刻压力分布情况

6.5 本章小结

本章主要对 NREL phase VI 整个风机进行了建模与数值计算并研究了轮毂、机舱和塔筒结构对于风机流场与空气动力性能的影响。本章首先通过 NREL 给出的相关数据建立了整个风机的模型。然后在上一章网格的基础上做出调整绘制得到了整个风机计算域的网格。之后本文使用基于 OpenFOAM 的不可压 RANS 求解器 pimpleDyMFoam 结合 AMI 网格运动处理方法对整个风机在四个风速下进行了数值计算，并将结果与之前单个风轮的进行比较。

本章得出了各个风速下风轮的尾涡情况，一个旋转周期内推力变化曲线，不同截面位置的 C_p 分布情况以及 5m/s 和 10m/s 风速下塔筒在不同时刻的压力分布曲线。通过比较可以得到轮毂和机舱结构对于风机流场和空气动力性能没有太大影响。而塔筒存在对于风机的尾涡，推力有比较明显影响。本章重点对 10m/s 风速下相应截面的涡情况进行了分析，观察到了叶尖涡与塔筒产生的涡的相互作用。本章也研究了叶片与塔筒重合时的 C_p 分布，发现与之前的结果相差不大，由此本章推测对于上风向风机塔筒存在对于风机的推力影响可以忽略，但是仍需考虑它可能会引起的周期性振动。本章之后对塔筒在不同时刻的压力分布进行分析得出，在风机叶片与塔筒重合时，塔筒前缘叶片长度范围内的压力值会急剧下降。考虑到实际情况，这很可能导致叶片向塔筒变形，造成二者碰撞，故设计时需要着力避免。

第七章 结语

7.1 总结

本文使用基于 OpenFOAM 平台的不可压 RANS 求解器 pimpleDyMFoam 结合 AMI 运动网格处理方法首先对 NREL phase VI 风机在 3° 桨距角，四个风速下（5m/s、10m/s、15m/s 和 25m/s）的三维流场和空气动力性能进行了数值计算。该计算结果与实验值有良好的吻合，但是对于 10m/s 和 15m/s 的情况出现了比较大的偏差，本文对此进行了深入分析。之后本文对整台风机进行建模，并进行数值计算，深入探讨分析了轮毂、机舱和塔筒结构对于风机流场和空气动力性能的影响。

通过整个研究，本文得出如下结论：

- (1) snappyHexMesh 自动网格划分工具能够有效处理风机叶片网格的划分，该工具具有非常强的灵活性，可以有效提升网格划分效率，但它对于尖锐边界以及曲率过大的边界的捕捉能力有待提高。
- (2) pimpleDyMFoam 求解器结合 AMI 运动网格处理方式能够有效求解三维风机的流场和空气动力性能。但该方法对 AMI 交界面网格质量有较高要求，使用时需要特别注意。
- (3) k- ω SST 湍流模型结合壁面函数的方式可以有效模拟壁面处的流动模式，具有较强的分离捕捉能力。而且结合壁面函数后，该方法可以大大减少计算量。
- (4) 本文计算结果与实验有良好的吻合，但是转矩值在 10m/s 和 15m/s 时出现了严重的偏低，分析认为是由于叶片导边处网格精度不够导致求解器对该处吸力面的压力峰值预报严重偏低所致。此外也需要考虑到这两个风速是流动从稳定到不稳定的过渡区域，对求解的要求比较高。
- (5) 对于上风向风机，轮毂与机舱结构对于风机的流场与空气动力性能计算影响很小。塔筒结构能够影响风机的尾涡，并会导致叶片推力的出现周期性低谷，但总体影响不大；但是叶片掠过塔筒时会使得塔筒处压力急剧下降。故在实际设计时除了考虑这种周期性受力变化带来的振动外，也要防止叶片经过塔筒时由于受吸变形而和塔筒发生碰撞。

7.2 展望

本文所采用的方法能够有效处理风机的三维流场和空气动力性能计算，以此为基础，未来可以将研究内容深入扩展，比如：

- (1) 利用专业网格划分软件处理叶片的网格，并着重计算 10m/s 和 15m/s 的风速情况，研究叶片导边处网格质量对风机空气动力性能的影响。
- (2) 进一步研究风机在不同偏航角，不同桨距角，不同转速，不同叶片长度，剪切风，湍流风，下风向等状态下的空气动力性能。
- (3) 研究两台风机间相对位置不同对于其流场及空气动力性能的影响。
- (4) 结合两相流求解器和造波模块，对各种形式的海洋固定式风机的受力进行耦合计算。
- (5) 结合两相流求解器，添加锚链模块，对各种海洋浮式风机的受力和运动进行耦合计算。

需要注意的是，上面的第四和第五项目前实现还是有很大难度的，特别是第五项，目前 OpenFOAM 的模块无法直接解决，可能需要引入 Overset 的网格处理方式。不过，本文相信随着研究的深入和计算机及软件的发展，实现上述目标也是指日可待了。

参考文献

- [1]. Archer, C. L.,Jacobson, M. Z. Evaluation of global wind power[J]. Journal of Geophysical Research, 2005, 110 (D12110): 1-20.
- [2]. Wikipedia. Wind power, http://en.wikipedia.org/wiki/Wind_power. accessed 2012.04.
- [3]. Meyer, N. I. Danish wind power development[J]. Energy for sustainable development, 1995, 2 (1): 18-25.
- [4]. Gipe, P. Wind energy comes of age California and Denmark[J]. Energy Policy, 1991, 19 (8): 756-767.
- [5]. GWEC. Global Wind Report Annual market update 2011[R]. 2011.
- [6]. Consult, B. International Wind Energy Development World Market Update 2009[R]. 2009.
- [7]. Hansen, M. O. L.,Madsen, H. A. Review Paper on Wind Turbine Aerodynamics[J]. Journal of Fluids Engineering-Transactions of the Asme, 2011, 133 (11).
- [8]. Glauert, H. Airplane propellers[M]. New York: Dover Publications, 1963.
- [9]. Shen, W. Z.,Mikkelsen, R.,Sørensen, J. N., etc. Tip loss corrections for wind turbine computations[J]. Wind Energy, 2005, 8 (4): 457-475.
- [10]. Leishman, J.,Beddoes, T. A semi-empirical model for dynamic stall[J]. Journal of the American Helicopter Society, 1989, 34 (3): 3-17.
- [11]. Whale, J.,Fisichella, C.,Selig, M. Correcting inflow measurements from HAWTS using a lifting surface code[A]. In 1999 ASME wind energy symposium[C], 1999; 175-185.
- [12]. Hess, J. L. Review of integral-equation techniques for solving potential-flow problems with emphasis on the surface-source method[J]. Computer Methods in Applied Mechanics and Engineering, 1975, 5 (2): 145-196.
- [13]. Landahl, M. T.,Stark, V. J. E. Numerical lifting-surface theory-Problems and progress[J]. AIAA Journal, 1968, 6 (11): 2049-2060.
- [14]. Preuss, R. D.,Morino, L.,Suciu, E. O. Unsteady potential aerodynamics of rotors with applications to horizontal-axis windmills[J]. AIAA Journal, 1980, 18 (4): 385-393.
- [15]. Hand, M. M.,Simms, D. A.,Fingersh, L. J., etc. Unsteady Aerodynamics Experiment Phase VI : Wind Tunnel Test Configurations and Available Data Campaigns Unsteady Aerodynamics Experiment Phase VI : Wind Tunnel Test Configurations and Available Data Campaigns[R]. National Renewable Energy Laboratory, 2001.
- [16]. NREL 10-m Wind Turbine Testing in NASA Ames 80'x120' Wind Tunnel.
<http://wind.nrel.gov/amestest/>. accessed 2012.04.
- [17]. Simms, D.,Schreck, S.,Hand, M., etc. NREL Unsteady Aerodynamics Experiment in the NASA-Ames Wind Tunnel : A Comparison of Predictions to Measurements[R]. National Renewable Energy Laboratory, 2001.
- [18]. Sørensen, N. N.,Michelsen, J.,Schreck, S. Navier-Stokes Predictions of the NREL Phase VI Rotor in the NASA Ames 80 ft*120 ft Wind Tunnel[J]. Wind Energy, 2002, 5: 151-169.
- [19]. Duque, E. P. N.,Burklund, M. D.,Johansen, W. Navier-Stokes and Comprehensive Analysis Performance Predictions of the NREL Phase VI Experiment[J]. Journal of Solar Energy Engineering, 2003, 125: 457-467.

-
- [20]. Potsdam, M. A.,Mavriplis, D. J. Unstructured Mesh CFD Aerodynamic Analysis of the NREL Phase VI Rotor[A]. In 47th AIAA aerospace sciences meeting including the New Horizons Forum and aerospace exposition[C], Orlando, Florida, 2009.
 - [21]. Li, Y. W.,Paik, K. J.,Xing, T, etc. Dynamic overset CFD simulations of wind turbine aerodynamics[J]. Renewable Energy, 2012, 37 (1): 285-298.
 - [22]. Digraskar, D. A. Simulations of Flow Over Wind Turbines[D]. University of Massachusetts Amherst, 2010.
 - [23]. 张义华. 水平轴风力机空气动力学数值模拟[D]. 重庆大学, 2007.
 - [24]. 俞国华,杜朝辉. 水平轴风力机气动性能的三维数值计算[J]. 中国科技论文在线, 2009, 4 (10): 6.
 - [25]. 王强. 水平轴风力机气动性能数值模拟研究[D]. 南京航空航天大学, 2010.
 - [26]. 查晶晶. 基于 OpenFOAM 的数值造波与消波模型及其应用[D]. 上海交通大学, 2011.
 - [27]. Jasak, H.,Jemcov, A.,Tukovic, Z. Openfoam: A c++ library for complex physics simulations[A]. In International Workshop on Coupled Motions in Numerical Dynamics[C], IUC, Dubrovnik, Croatia, 2007.
 - [28]. Weller, H. G,Tabor, G,Jasak, H, etc. A tensorial approach to computational continuum mechanics using object-oriented techniques[J]. Computers in physics, 1998, 12 (6): 620-631.
 - [29]. Features of OpenFOAM. <http://www.openfoam.com/features/>. accessed 2010.04.
 - [30]. 缪莎. 典型水面船模和穿梭船粘性兴波问题的数值模拟[D]. 上海交通大学, 2011.
 - [31]. Sasongko, N. Open Source Computational Fluid Dynamic: Challenges and its Future. 2009.
 - [32]. Schmitt, F. G About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity[J]. Comptes Rendus Mécanique, 2007, 335 (9-10): 617-627.
 - [33]. Menter, F. R. Two-equation eddy-viscosity turbulence models for engineering applications[J]. AIAA Journal, 1994, 32 (8): 1598-1605.
 - [34]. Menter, F. R. Review of the shear-stress transport turbulence model experience from an industrial perspective[J]. International Journal of Computational Fluid Dynamics, 2009, 23 (4): 305-316.
 - [35]. Wikipedia. law of the wall. http://en.wikipedia.org/wiki/Law_of_the_wall. accessed 2012.05.
 - [36]. CFD-Wiki. Near-wall treatments. http://www.cfd-online.com/Wiki/Two_equation_models. accessed 2012.05.
 - [37]. Jasak, H. Error analysis and estimation for the finite volume method with applications to fluid flows[D]. University of London, 1996.
 - [38]. OpenFOAM: The OpenSource CFD Toolbox, User Guide, Version 2.1.0. 2011.
 - [39]. Beaudoin, M.,Jasak, H. Development of a Generalized Grid Interface for Turbomachinery simulations with OpenFOAM[A]. In Open Source CFD International Conference 2008[C], Berlin, 2008.
 - [40]. Beaudoin, M.,Jasak, H. OpenFOAM Turbo Tools: From General Purpose CFD to Turbomachinery Simulations[A]. In ASME-JSME-KSME Joint Fluids Engineering Conference[C], 2011.
 - [41]. Hunt, J.,Wray, A.,Moin, P. Eddies, streams, and convergence zones in turbulent flows[R]. Center for Turbulence Research Report CTR-S88, 1988.
 - [42]. 周胡,王强,万德成. 风机叶片三维扰流场的数值模拟[A]. 见第十一届全国水动力学学术会议暨第二十四届全国水动力学研讨会并周培源教授诞辰 110 周年纪念大会[C], 无锡, 2012.
 - [43]. Gomez-Iradi, S.,Steijl, R. Development and Validation of a CFD Technique for the Aerodynamic Analysis of HAWT[J]. Journal of Solar Energy Engineering, 2009, 131.

致辞

这里我首先要对我的导师万德成教授表示由衷的感谢。在大四上半学期，我有幸加入万老师带领的 IWSH 会议志愿者队伍。会后万老师将我带入了他的 CFD 小组，也将我领入了 CFD 这个未曾接触的世界。万老师通过与我的一次次交谈了解我的研究兴趣，并以此安排我的研究任务。在做毕设的漫长过程中，万老师总是给予及时的指点与教导，帮我端正研究方向，也为我解开心中的疑惑。但是在具体研究方法和研究内容上，万老师给予我极大的自由，并鼓励我自行探索解决问题。我非常感激万老师的这一做法，这个过程让我真正地领略到了做科研的乐趣。此外，万老师也一次次教导我们为人处世的道理，给我留下了深刻的印象。再次感谢万德成老师对我的教诲与指导！

此外，我也要非常感谢美国国家可再生能源实验室的 Scott Schreck 先生无偿提供给我相关的实验数据，并耐心地予以指导。我也要感谢爱荷华大学 Yuwei Li 博士对于计算方法的指导。我也非常感谢 CFD-Online 论坛 OpenFOAM 板块上的各位网友的热心交流与讨论，你们让我在几乎绝望的时候看到了解决难题的希望。

当然，我不会忘记万老师 CFD 小组成员给予的极大帮助。最大的感谢给予沈志荣师兄，沈师兄耐心细致地指导总是能为我化解迷津，将我领到正确的道路，没有你我几乎不可能完成这次毕设任务，这里再次表示感谢。此外我还要感谢曹洪建师兄、刘远传师兄帮助我熟悉了解 OpenFOAM 的运算流程，感谢周胡师兄在整个过程中与我的沟通交流。

最后，我要真切地感谢我的女友刘雪莹。感谢你陪伴我走过了大学的精彩四年，给予我太多的美妙回忆，你的微笑是我最大的支持与动力。我也要感谢我的室友在做毕设期间的默默关照与支持。最后的最后，非常感谢我的家人，你们永远是我最坚强的后盾。

我爱你们！

NUMERICAL STUDY OF 3D FLOWS AND AERODYNAMIC PERFORMANCE OF WIND TURBINE

Stepped into the 21st century, the world is facing two serious crisis: the depletion of fossil fuels and the global warming (and other environmental problems). In order to tackle these issues, more and more countries pay increasing attention and expectations to renewable energy. Wind energy is a kind of renewable energy that is clean, relatively cheap and enjoys great generation potential, thus being developed all over the world now.

Horizontal axis wind turbine is the most popular method to capture wind energy. However, due to the complex interaction between wind with blades, wind turbine always has very complex and unstable aerodynamic performance. What's more, the interaction will directly influence the efficiency of wind turbine, and it also has a great effect on the loads on blade, gearbox and bearings, thus affecting the reliability and life span of the turbine. Therefore, accurate prediction of flows and aerodynamic performance of wind turbine is of great significance for the design of a wind turbine with higher efficiency and longer life span.

Since the max blade tip velocity for current wind turbine is usually less than 100m/s, so for flows around wind turbine, the Mach number is less than 0.3. Therefore, we can use incompressible fluid solvers for most turbines. Different kinds of methods for predicting the aerodynamic performance of wind turbine have been developed.

The most popular method is Blade Element Moment (BEM) method. It first divides the blade into many numbers of small parts, called the blade element. Then it calculates loads on every blade elements and integrates them together to obtain the loads on the whole blade. This method is very efficient, which is the main advantage. However, since the actual blade is three-dimensional, BEM models need to use empirical corrections to two-dimensional results to account for three-dimensional effects, such as dynamic stall, tip loss and so on. Besides, in order to calculate loads on blade elements, BEM method is highly dependent on the given lifting and drag coefficients of the blade elements.

In order to get more physics of wind turbine aerodynamics while keeping efficient, three-dimensional inviscid models were introduced, including lifting line, panel, vortex and Boundary Integral Equation and so on. Although these methods could capture more three-dimensional effects than the BEM method while retaining high efficiency, they are not able to handle viscous effects since they are based on potential flow theory. At the same time, flows around wind turbine usually have very strong viscous effects and separation. So application of these methods is limited.

The next method is Computational Fluid Dynamics (CFD) which solves the Navier-Stokes

equation. According to different ways of treatment of turbulence, computational fluid dynamics is divided into Direct Numerical Simulation (DNS), Large Eddy Simulation (LES) and Reynolds-Averaged Navier-Stokes (RANS). DNS method solves the governing equation all over the computation domain directly based on very fine mesh and small time step. Due to the huge computation time, DNS is usually taken as impractical for industrial problems. LES divides turbulence into large scale part and other relative smaller part. Only the large scale part of turbulence is solved directly while the rest part of turbulence is treated as corrections to large scale part. However, this method is still too time consuming to be used in most research libraries. RANS is the most widely used CFD method in industry. It treat the turbulence into the averaged part and the fluctuation part around the averaged part. It's relatively more computationally efficient than the other two methods, which is adopted in this paper.

Most numerical computations of wind turbines are compared to the National Renewable Energy Laboratory (NREL) Unsteady Aerodynamics Experiment (UAE) carried out in the NASA Ames 24.4m×36.6 wind tunnel in 1999. The testing wind turbine is the NREL phase VI, which has a power of 20kW. It has two blades with NREL S809 profile. The rotor diameter is 10.058m and the hub height is 12.192m. The experiment provides quantity of high-quality data, thus is regarded as a benchmark for evaluation of wind turbine aerodynamics computer codes.

This paper presents the simulations of the NREL phase VI wind turbine based on the open source finite volume CFD platform OpenFOAM. The simulations are performed using unsteady RANS solver pimpleDyMFoam with k-omega SST turbulence model and wall function approach for the near-wall simulations. PIMPLE algorithms (coupled SIMPLE and PISO algorithms) are used to solve the pressure-velocity coupling in the Navier-Stokes equation. Arbitrary Mesh Interface (AMI) is used to handle sliding mesh interface. Computations for the turbine includes four wind speed (5, 10, 15, and 25m/s) at a fixed blade pitch angle of 3°with constant rotational speed 72 rpm, including a rotor only model and a whole wind turbine. Extensive comparison against experimental results is made for the rotor only case to demonstrate the accuracy of the solver and methods used in the simulation. Then, comparison of the rotor only case and whole turbine case is made to invest the effect of hub, nacelle and tower on flows and aerodynamic performance of wind turbine.

For the rotor only case, the paper first describes in detail about how to built the three-dimensional rotor model with CATIA. This part focuses on the arrangement of experimental data for the blade and extensive attention is paid on dealing with transition part and tip of blade. The paper provides a successful way of building the model. Then the paper makes attempts to test the mesh capability of snappyHexMesh utility in OpenFOAM. With a great number of trials and errors, snappyHexMesh utility is demonstrated to be very flexible and efficient in meshing. However, it is also believed that the utility has drawbacks when meshing with sharp features or features with very large curvature. In this paper, it results in some highly skewed faces in the trailing edge of blade and bad quality AMI faces. Instead, the paper uses ICEM-CFD to draw the background mesh and then imported into snappyHexMesh to finish the mesh. Although a bit complicated, this method is demonstrated to be very successful. Then numerical simulations of NREL phase VI rotor is performed using the pimpleDyMFoam solver, and the results are

compared to experiments. The comparison shows good agreement which demonstrated the ability of the code pimpleDyMFoam in predicting flows and aerodynamic performance of wind turbine and the Arbitrary Mesh Interface in dealing with mesh movements. However, big difference is observed for aerodynamic torque at 10m/s and 15m/s. With deep investigation and discussion of the distribution of normal force coefficient, tangential force coefficient, pressure coefficient and finally mesh at the leading edge of blade, the paper concludes that the poor quality at the leading edge of the blade results in a big under-prediction of pressure coefficient at the suction side of the leading edge which finally result in the under-prediction of aerodynamic torque. This indicates limited ability of snappyHexMesh in capture features with large curvature. The paper also gives extensive discussion of the vorticity field and velocity field. It is obvious that at 5m/s and 10m/s most of the flow is attached, these cases also show strong blade tip and root vortices. When wind speed increase, flows around the whole trailing edge of blade suffer intensive separations. In these cases, interaction of the tip vortices with unsteady separation from the blade causes breakdown of the tip vortices.

For the whole turbine case, hub, nacelle and tower are added into the geometry. The mesh topology is almost the same with the rotor only case with minor changes of the sharp of the background mesh domain. The rest case setup is identical to the rotor only case. Then numerical simulations of NREL phase VI turbine is performed using the pimpleDyMFoam solver, and the results are compared to rotor only case to investigate the influence of hub, nacelle and tower on flows and aerodynamic performance of the wind turbine. The comparison shows that hub and nacelle have minor effect on the turbine's performance, while the tower can induce more obvious influence. The vorticity field shows extensive interactions between tower vortices and blade tip vortices. A deep investigation of the interaction process is also showed at a cut plane of the 10m/s wind speed case. The comparison also shows that the thrust curve gets a drop when the azimuth angle reaches 180 degree. However, the difference is relatively small of the pressure coefficient distribution between the two cases, which indicates that for upwind turbines, the blade tower interaction has little effect on the total thrust output. But it is very possible that the cyclical drop of loads on blade will introduce vibration which should be considered. Surprisingly, the paper shows that the tower suffers a very intensive drop of surface pressure when the blade approaches the tower. To consider it in reality, this phenomenon can cause deformation of blade toward the tower, which may result in dangerous collision.

From this study, following conclusions can be made:

- (1) snappyHexMesh is a very flexible and efficient meshing tool, but its capability to capture sharp features and features with large curvature need to be improved.
- (2) pimpleDyMFoam solver combined with AMI can solve flows and aerodynamic performance of wind turbine successfully. But special attention should be paid to the mesh quality of AMI faces.
- (3) k- ω SST turbulence model with wall function can capture the unsteady separation of the flows. Besides, this method can greatly reduce the amount of mesh.
- (4) hub, nacelle and tower have little effect on the global performance of wind turbine. But the pressure of tower can suffer a big drop when the blade approaches which may cause collision of blade and tower in reality.